

Universidad de las Ciencias Informáticas

FACULTAD 5



**Título: Multimedia para la enseñanza del proceso de
Gestión de Riesgos en la Calidad de Software en la UCI.**

**Trabajo de Diploma para optar por el Título de
Ingeniero en Ciencias Informáticas**

Autor: Marbelis Barrera Barroso

Tutor: MSc. Yamilis Fernández Pérez

Ciudad de la Habana

Junio 2009

''Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.''

Albert Einstein.

Datos de Contacto

Tutor: MSc. Yamilis Fernández Pérez

Correo electrónico: yamilisf@uci.cu

Graduada de Ingeniera en Sistema Automatizado de Dirección, en 1992 en el ISPJAE, Profesora asistente desde 1995. MSc. en Informática Aplicada en 1995. Imparte docencia en universidades desde 1992. Ha desarrollado trabajos con Universidades extranjeras en Brasil, Bolivia, Canadá, etc. Es la jefa de departamento docente central de Ingeniería y Gestión de Software de la UCI desde su fundación.

AGRADECIMIENTOS

*A mis **Padres**: Por todo el amor, la bondad, gratitud y entrega total en el apoyo a mis estudios, con paciencia han sabido conducirme por el camino correcto para la obtención del éxito final de mi carrera, por la sabiduría con la que han mantenido la unión de la familia, gracias por darme tanto amor para hoy ver mis sus sueños hechos realidad y a la vez tener un futuro garantizado, gracias por legarme tanta fortuna y convertirme en un ser tan dichoso, que sin ustedes no hubiese sido posible.*

*A mi **Tía María**: Mi segunda madre la cual me aconseja, me regaña dándome siempre las mejores enseñanzas, convirtiéndose en mi mejor profesora de clases en perfección que nunca tuve en ninguna enseñanza, por lo que te digo que continuaras lavándome y planchándome las ropas para no salir a la calle estrujada porque ya hoy soy ingeniera y el tiempo para trabajar como toda una profesional no me alcanzara, hoy se hacen realidad tus sueños y los míos.*

*A mi **Hermana Nena**: Por ser el regalo mas grade que me ha proporcionado la vida ya que tuvimos el privilegio de estar y andar juntas desde el vientre de nuestra madre hasta los días anteriores a hoy, el destino nos jugo una mala pasada, no permitiéndonos graduarnos junta ya que estudiamos la misma carrera y en este mismo centro es por eso que hoy con todo el amor del mundo te agradezco la valentía con que asumes los sinsabores da la vida.*

*A **Oswaldo mi novio**: Por todo el amor, apoyo, comprensión y dedicación con la que me has tratado, aunque me regañas mas que mi mamá, se que lo haces por mi bien, y no porque seas un viejo. De no haber estado a mi lado todo este tiempo, estoy segura que este día jamás habría llegado. Hoy creo que más que mis sueños y los de mi familia se cumplen los tuyos.*

A mis compañeras, amigas, amigos y hermanas: No mencionare a ninguno en específico, para no accionar celos, pues cada uno sabe en particular el lugar que ocupo en mí corazón, pues todos fueron y serán muy importante para mí, son el recuerdo más lindo más querido y máspreciado que me llevo de esta Universidad.

A mi tutora, claustro de profesores, tías de la residencia estudiantil: en fin a todas a aquellas personas que de una forma u otra fueron protagonista para que este día llegara.

DEDICATORIA

A mis padres por ser la luz que me ilumina, a mi tía por ser mi segunda madre, a mis hermanas por ser el regalo más grande que me ha dado la vida, a mi novio por haber luchado a mi lado sin cansancio.

Resumen

Con el desarrollo actual de las Tecnologías de la Informática y las Comunicaciones (TIC) ha sido inevitable el uso de ellas en la sociedad debido a la cantidad de posibilidades que nos brindan. En el sector educacional se han conseguido grandes logros con la producción de software educativo. La incorporación de estímulos visuales y sonoros del software educativo basado en técnicas multimedia, agiliza y dinamiza la enseñanza. Debido a esto, la Universidad de las Ciencias Informáticas (UCI) se ha especializado en la creación de sistemas multimedia destinados al apoyo del proceso de enseñanza.

Por tal motivo, el presente trabajo tiene como propósito el análisis, diseño e implementación de una multimedia interactiva para el apoyo del proceso de enseñanza-aprendizaje del curso de “Gestión de Riesgos”.

Se realiza un estudio de las herramientas existentes y las metodologías a utilizar para dar cumplimiento al desarrollo del trabajo, considerándose como la herramienta más apropiada Macromedia Flash 8 y su lenguaje de programación ActionScript. Como metodología de desarrollo el Proceso Unificado del Software (RUP), como lenguaje representativo UML y la extensión del mismo para aplicaciones multimedia OMMMA-L, y como lenguaje para gestionar y agrupar los datos en volúmenes compactos de información se utiliza XML.

Palabras Clave:

Software Educativo, Multimedia, Multimedia Interactiva, Gestión de Riesgos.

Índice

INTRODUCCIÓN	11
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.	16
Introducción:	16
1.1 ¿Qué es Calidad?	16
1.1.2 <i>¿Qué es la Calidad del software?</i>	17
1.2 ¿Qué es Riesgo?	17
1.2.1 <i>Historia de la Gestión de riesgo.</i>	18
1.2.2 <i>Objetivo de la gestión de riesgo</i>	19
1.3 Software educativo.	20
1.3.1 <i>¿Qué es Software Educativo?</i>	20
1.3.2 <i>Características esenciales de los programas educativos.</i>	20
1.4 Multimedia.	21
1.4.1 <i>Tipos de Multimedia.</i>	21
1.4.2 <i>Definición de Hipertexto.</i>	22
1.4.3 <i>Definición de Hipermedia.</i>	23
1.4.4 <i>Multimedia en la educación.</i>	23
1.5 Tendencias tecnológicas actuales.	24
1.5.1 <i>¿Qué es un Entorno Virtual de Aprendizaje (EVA)?</i>	24
1.5.2 <i>Moodle:</i>	25
1.6 Herramientas para la creación de Software Multimedia.	25
1.6.1 <i>Macromedia Director.</i>	26
1.6.2 <i>ToolBook.</i>	26
1.6.3 <i>Macromedia Flash 8</i>	26
1.6.4 <i>Herramienta a utilizar.</i>	27
1.7 Metodologías a considerar para el desarrollo del software.	28
1.7.1 <i>Proceso Unificado de Rational (RUP).</i>	28
1.7.2 <i>Programación Extrema (XP).</i>	28
1.7.3 <i>Desarrollo Guiado por la Funcionalidad (FDD).</i>	29
1.7.4 <i>Selección de la metodología a utilizar.</i>	30
1.8 Lenguajes de modelado.	30
1.8.1 <i>UML (Lenguaje Unificado de Modelado) y el Rational Rose.</i>	30
1.8.2 <i>Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA -L).</i>	31
1.9 Lenguajes utilizados en la multimedia.	32
1.9.1 <i>El lenguaje XML.</i>	32
1.9.2 <i>ActionScript.</i>	33

Conclusiones del Capítulo 1:-----	34
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN. -----	35
Introducción.-----	35
2.1 Identificación de la audiencia.-----	35
2.2 Especificación del contenido.-----	35
2.2.1 Mapa Conceptual.	37
2.2.2 Mapa conceptual General.	37
2.2.3 Sub-mapa de Riesgo del software.	39
2.2.4 Sub-mapa de Identificación del Riesgo	40
2.2.5 Sub-mapa de Gestión de Riesgo en la fase de Ingeniería de Requisito.	41
2.3 Descripción del Modelo de Dominio.-----	41
2.3.1 Diagrama de clases del Modelo de Dominio.	42
2.4 Diagramas de Navegación.-----	43
2.5 Requerimientos Funcionales.-----	45
2.6 Requisitos no funcionales.-----	46
2.6.1 Hardware.	46
2.6.2 Usabilidad.	46
2.6.3 Software.	46
2.6.4 Apariencia.	47
2.6.5 Requisitos de diseño e implementación.	47
2.6.6 Requisitos de soporte.	48
2.7 Modelo de casos de uso.-----	48
2.7.1 Actor presente en el sistema:	49
2.7.2 Breve descripción de los casos de uso del sistema:	49
Conclusiones del Capítulo 2:-----	52
CAPÍTULO 3: SOLUCIÓN DE LA DESCRIPCIÓN PROPUESTA. -----	53
Introducción.-----	53
3.1 Principios de Diseño.-----	53
3.2 Diagramas de Presentación del Modelo de Diseño.-----	54
3.3 Descripción y Estructura de Principales Archivos XML.-----	56
3.4 Diagrama de Componentes.-----	58
Conclusiones del Capítulo 3:-----	59

CONCLUSIONES GENERALES	60
RECOMENDACIONES	61
REFERENCIAS	62

Introducción

En la actualidad, la producción de software es un pilar muy importante que ha llegado a involucrar a una cantidad considerable de naciones, organizaciones y profesionales de Informática y la computación. Todos se empeñan en mantener o lograr posiciones prestigiosas en el mercado, o simplemente dependen en gran medida de la producción o elaboración de éste, ya que cada vez son más los sistemas controlados por software.

Estudios realizados por (The Standish Group International) en 1995, revelan resultados sorprendentes en los costes del software. Se estima que las compañías y el gobierno de los USA gastaron 81.000 millones de dólares en proyectos cancelados. Los proyectos terminados, cuyo plazo de ejecución ha sido superior al previsto, supondrán un coste de 59.000 millones de dólares. Se estima que solo el 16.7% de los proyectos software fueron terminados en el plazo y con el presupuesto previstos. (RUBINSTEIN, 2007).

Estudios más recientes realizados por el mismo grupo (The Standish Group International) en el 2006, revelan que el 35 por ciento de los proyectos de software iniciados pueden clasificarse como exitosos; sólo el 19 % de los proyectos iniciados fueron simples fallos. Estos datos demuestran que aunque el problema no se ha solucionado totalmente, existe una mejora en el desarrollo del software. El presidente de este grupo Jim Johnson citó tres razones para la mejora de la calidad del software: una mejor gestión de proyectos, desarrollo iterativo y la nueva infraestructura Web.

Las producciones de software en el mundo se ven afectadas por diferentes inconvenientes como son:

1. Grandes retrasos en la programación.
2. Inconsistencia en su funcionamiento.
3. Insatisfacción de los clientes.

Estas tres causas fundamentales están dadas en parte por la falta de calidad y la mala práctica de la gestión de riesgo, ya que comprender los riesgos que se corren con la realización de estos procesos y tomar medidas para evitarlos o gestionarlos,

es un elemento clave en la realización de los mismos. Pues como bien citó Pressman, uno de los aspectos más importantes en el desarrollo del software es la calidad y, dentro de ella, el tema de gestión de los riesgos cobra vital importancia. (Pressman 2005)

La gestión de riesgos en el ámbito del software, procura formalizar conocimiento orientado a la minimización o evitación de riesgos en proyectos de desarrollo de software, mediante la generación de principios y buenas prácticas de aplicación realista (Ropponen, 2000). Hasta el momento se han propuesto y utilizado diferentes enfoques de gestión del riesgo desde que Boehm (Boehm, 1988) atrajo a la comunidad de ingeniería del software hacia la gestión del riesgo. Sin embargo, es evidente que pocas organizaciones utilizan todavía de una forma explícita y sistemática métodos específicos para gestionar los riesgos en sus proyectos software.

Los objetivos de la gestión de riesgos son identificar, controlar y eliminar las fuentes de riesgos antes de que empiecen a afectar el cumplimiento de los objetivos del proyecto (Boehm, 1991).

Los riesgos pueden afectar los futuros acontecimientos, por lo que podemos por tanto, cambiando nuestras acciones actuales, crear una oportunidad para una situación diferente y, con suerte, mejor para nosotros en el futuro.

Los riesgos del proyecto amenazan al plan del proyecto. Es decir, si los riesgos del proyecto se hacen realidad, es probable que la planificación temporal del proyecto se retrase y que los costos aumenten. Los riesgos del proyecto identifican los problemas potenciales de presupuesto, planificación temporal, personal, asignación y organización.

La Gestión de Riesgos en proyectos de desarrollo de software en Cuba se percibe a niveles organizacionales, y no como parte de un proceso de gestión específico al desenvolvimiento de los proyectos. El tratamiento de los riesgos se manifiesta fundamentalmente en los planes de seguridad y de contingencia. Se reconoce que estas planificaciones son vitales, pero con frecuencia solo tienen en cuenta los riesgos de tipo tecnológico. (VELIZ, 2007)

La Universidad de las Ciencias Informáticas (UCI) como una de las primeras y más importantes obras de la Batalla de Ideas en nuestro país, proyecto creado e ideado por nuestro Comandante en Jefe, es una entidad que puede favorecer al crecimiento de la economía, pues la mayoría de sus proyectos presentan convenios con otros países. La misma no se encuentra ajena a los acontecimientos científico-técnicos a escala mundial, por lo cual a sido un pilar importante y fundamental. La misma se ve entonces en la necesidad de capacitar al personal vinculado a los proyectos, logrando una formación con sólidos conocimientos teóricos y amplias habilidades prácticas.

En la UCI la Gestión de Riesgos se da como un tema dentro de la asignatura curricular Ingeniería de Software 1, y se imparte como un curso del perfil de calidad. Ésto no quiere decir que no exista desconocimiento del tema, a pesar de haber sido tratado en una serie de trabajos de diploma en cursos anteriores. En una encuesta realizada a la profesora Yeleny Zulueta, especialista en el tema de Gestión de Riesgos de la facultad 9, ella plantea que aunque todos los proyectos productivos deberían llenar una plantilla con la lista de sus riesgos, dicha actividad se realiza en muy pocos casos. De cierta forma, se conocen informalmente los riesgos que podrían afectar el trabajo, pero éstos no son registrados y no se procede a su gestión, lo cual afecta considerablemente la calidad de los productos finales. Esta situación se debe en la mayoría de los casos a la falta de conocimiento sobre posibles modelos y herramientas para llevar a cabo una planificación adecuada de la gestión de riesgos.

Dada la situación anterior, el **problema científico** que se pretende resolver con la realización de este trabajo es:

¿Cómo sustentar la inclusión de buenas prácticas del proceso de enseñanza de la Gestión de Riesgos en las TIC?

Como **objeto de estudio** se plantea:

Medios de enseñanza para la Gestión de Riesgos en la UCI.

Objetivo general:

Implementar una multimedia interactiva que apoye el proceso de enseñanza-aprendizaje del tema gestión de riesgos en la Calidad de Software en la UCI.

Se tiene como **campo de acción** de la investigación:

El proceso de desarrollo de una multimedia para la enseñanza de la Gestión de Riesgos.

Tareas:

- Analizar la utilización en el mundo de herramientas educativas como medios de enseñanza de la gestión de riesgos, e identificar estándares y métodos utilizados por éstas.
- Analizar las ventajas del lenguaje de programación para multimedia “*Action Script*”, así como del lenguaje de marcas XML para almacenar el contenido.
- Seleccionar la metodología de desarrollo de software a utilizar.
- Realizar un estudio de los cursos de Gestión de Riesgos existentes en la universidad, y seleccionar elementos que contribuyan al diseño e implementación del software.
- Confeccionar mapas conceptuales que contengan los conceptos necesarios para la comprensión de los procesos de gestión de riesgos.
- Recopilar los contenidos que serán utilizados por la multimedia.
- Diseñar e implementar una multimedia que contenga los contenidos recopilados y los mapas conceptuales confeccionados, y asociar a dichos contenidos distintos recursos - videos, juegos, auto-evaluaciones, etc. - que faciliten la comprensión de los mismos.

Idea a defender:

Con la elaboración de la multimedia “Gestión de Riesgos”, se conseguirá garantizar el apoyo en el proceso de enseñanza-aprendizaje de la Gestión de Riesgos en la Calidad de Software para los estudiantes de la UCI.

Durante el desarrollo de la investigación se emplean los siguientes métodos de investigación:

Histórico-Lógico: Para realizar un estudio profundo de toda la evolución y desarrollo del proceso de gestión de riesgos.

Analítico-sintético: Para el procesamiento de la información recopilada durante la investigación.

Empírico: En la realización de las encuestas hechas a profesores y especialistas en el tema Gestión de Riesgos.

Estructura de la tesis:

Capítulo 1: Fundamentación Teórica

En este capítulo se realiza la fundamentación del tema donde se explican algunos conceptos generales acerca del mismo, las tecnologías y tendencias actuales en la informática tomadas a consideración, además de explicar las herramientas, las metodologías y los lenguajes utilizados en el desarrollo de la aplicación.

Capítulo 2: Descripción de la Solución.

Este capítulo se comenzará con la realización de la especificación del contenido, bien detallado, que contendrá el sistema. Se realiza un levantamiento de los requisitos del sistema, tanto funcional como no funcional. Se opta por la realización de un Modelo de Dominio debido a lo difuso que se encuentran los procesos de negocio. Se realiza el Modelo de Navegación. Por último, se realiza el Modelo de Caso de Uso del Sistema, donde se justificarán los principales actores, así como la descripción de los casos de uso correspondientes.

Capítulo 3: Solución de la Descripción Propuesta.

En este capítulo se realiza la construcción de la solución propuesta en los flujos de trabajo de Diseño e Implementación. Se genera una serie de artefactos pertenecientes a la metodología OMMMA-L, como son los Diagramas de Presentación para las principales pantallas de la Multimedia. También se aborda el modelo de implementación, el cual incluye el diagrama de componentes y el diagrama de despliegue.

Capítulo 1: Fundamentación teórica.

Introducción:

Desde tiempos remotos la educación ha constituido la base de toda sociedad. La preparación individual de cada persona es la principal fuente de desarrollo de una nación. Por ésto y por las necesidades acumuladas de cada individuo, la educación ha encontrado nuevos retos y desafíos que la mantienen cada vez más enfocada en encontrar nuevos métodos y nuevas formas de impartir los conocimientos, de manera que la información llegue a cada estudiante de la misma forma.

Unido a ésto, se evidencia el desarrollo de las tecnologías de la información y las comunicaciones (TIC), tecnologías que la didáctica ha sabido aprovechar para su beneficio y el de los estudiantes, y que cada vez más centra su estudio en encontrar nuevas formas que permitan explotar al máximo sus ventajas para facilitar un mayor conocimiento y aprendizaje en los estudiantes en cualquier materia.

En el presente capítulo, se exponen los principales conceptos asociados al desarrollo del trabajo, que posibilitarán una mejor comprensión de las terminologías utilizadas, por ejemplo la introducción al concepto de calidad del software, tratando dentro de éste el tema de la gestión de riesgos. Se exponen además la importancia del software educativo y la importancia de la multimedia, con otros temas asociados que se verán en el desarrollo del capítulo.

1.1 ¿Qué es Calidad?

La calidad no es más que la cualidad de un objeto de encontrarse en su estado óptimo y del agrado del cliente. En la vida diaria nos enfrentamos a los efectos de la calidad. Desde los orígenes de nuestra existencia, la calidad ha tenido una gran influencia en el desarrollo alcanzado hasta la actualidad, debido a que constantemente se han buscado nuevas alternativas para satisfacer las necesidades y complacer los gustos de todas aquellas personas interesadas en un

determinado producto. La eficiencia, garantía y la competencia, exigen constantemente un incremento en la calidad.

1.1.2 ¿Qué es la Calidad del software?

A través de los años se han creado diferentes opiniones acerca de lo que es la calidad del software, y varios especialistas en el tema la han definido de diversas maneras, por ejemplo:

Según ISO no es más que: “El conjunto de características de una entidad que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas” ISO 8402 (UNE 66-001-92).

El profesor Pressman ha creado varias definiciones de lo que es la Calidad de Software. En 1992 la define como: “La concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados, y con las características implícitas que se espera de todo software desarrollado profesionalmente.” (PRESSMAN, 2005).

Oscar M. Fernández la define como “un conjunto de cualidades que lo caracterizan y que determinan su utilidad y existencia”. La calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

(Oscar 1995)

Desde la posición del autor, y teniendo en cuenta los conceptos anteriores, se define la Calidad de Software como Pressman.

1.2 ¿Qué es Riesgo?

El riesgo es algo con lo que convivimos diariamente, en cada paso que damos o decisión que tomemos, ya que éste se encuentra implícito en toda actividad. Es la probabilidad de una amenaza, o de que ocurra algún fenómeno peligroso. Todo cambio trae consigo riesgos que implican elecciones e incertidumbre. Los riesgos se pueden identificar, medir, e incluso evitarlos.

En su libro sobre análisis y gestión del riesgo, Robert Charette presenta la siguiente definición de riesgo:

En primer lugar, el riesgo afecta a los futuros acontecimientos. El hoy y el ayer están más allá de lo que nos pueda preocupar, pues ya estamos cosechando lo que sembramos previamente con nuestras acciones del pasado.

La pregunta es: ¿Podemos por tanto, cambiando nuestras acciones actuales, crear una oportunidad para una situación diferente y, con suerte, mejor para nosotros en el futuro? Ésto significa, en segundo lugar, que el riesgo implica cambio, que puede venir dado por cambios de opinión, de acciones, de lugares... En tercer lugar, el riesgo implica elección y la incertidumbre que entraña la elección. Por tanto, el riesgo, como la muerte, es una de las pocas cosas inevitables de la vida.

1.2.1 Historia de la Gestión de riesgo.

Históricamente la gestión de riesgos ha pasado por tres generaciones:

G1 Casuística: Es la primera generación. Data de principios de los años 80. En esta generación se definen los riesgos tecnológicos y las listas de comprobación de riesgos, y se limitan las tareas a la identificación de riesgos en los proyectos.

G2 Taxonómica: Es la segunda generación. Data de principios de los años 90. En esta generación se analizan los riesgos al inicio del proyecto. Cocho, Adam y Torralba definen esta visión como “preventiva”, “teorizante” y de medidas “curativas. También califican los modelos “meramente reactivos, con unas relaciones de causa-efecto basadas sólo en una confianza que parte de experiencias poco validadas”.

Dentro de esta generación se pueden incluir los siguientes modelos:

- **Modelo de Boehm:** De éste surgió una de las ideas fundamentales que las metodologías posteriores adoptarían: el análisis de riesgos. Este modelo tiene un carácter iterativo. En sus primeras fases plantea la necesidad de realizar diversas iteraciones dirigidas a mitigar los riesgos más críticos encontrados en el proyecto.

- **Modelo de Riesgos del SEI:** Este modelo consiste en la propuesta de un cuestionario desarrollado por el Software Engineering Institute (SEI), que tiene como objetivo colaborar en la identificación de riesgos de los proyectos.

G3 Causal: Es la tercera generación. Data de mediados de los 90 y se refiere en particular a proyectos informáticos. Surge de forma simultánea en Europa y en EEUU partiendo de la preocupación por proyectos de tanto riesgo como la adquisición o el desarrollo de software. Ésta es la generación actualmente emergente.

Los principales modelos de gestión de riesgos propuestos por G3 son:

- **Modelo MAGERIT de Gestión de Riesgos en Sistemas adaptado a Proyectos (Transición):** Este es un modelo formal para investigar los riesgos que soportan los sistemas de información, y para recomendar las medidas apropiadas que deberían adoptarse para controlar estos riesgos.

- **Euro-método:** Parte de un modelo inicial de buenas prácticas de mejora de la relación cliente-proveedor y de un modelo central de “estrategias orientadas a contingencia”, modelo que se ha ido convirtiendo en modelo de análisis de riesgos de los proyectos en la fase temprana de su contratación.

(Julián 2001)

1.2.2 Objetivo de la gestión de riesgo

La gestión de riesgos permite identificar, dirigir, evaluar, evitar y evadir los riesgos del proyecto. También nos debe permitir tomar las medidas necesarias para controlarlos o eliminarlos, y que aún cuando se produzcan, sea posible minimizar sus efectos negativos y nos permita finalizar satisfactoriamente el proyecto software.

Los riesgos siempre implican dos características, la incertidumbre y la pérdida. La incertidumbre no es más que la posibilidad de que el riesgo pueda o no ocurrir, y la pérdida es cuando el riesgo es real y nos trae consecuencias negativas, pérdidas o retrasos en la elaboración del software.

La gestión de riesgos en el ámbito del software procura formalizar conocimiento orientado a la minimización o evitación de riesgos en proyectos de desarrollo de software, mediante la generación de principios y buenas prácticas de aplicación realista [Ropponen, 2000]

1.3 Software educativo.

Desde el triunfo de la Revolución, nuestro comandante en jefe Fidel Castro Ruz ha considerado que la educación es y será siempre una tarea fundamental de la Revolución. Para tener una sociedad verdaderamente justa, libre y humana, constituye una prioridad nacional dar solución a uno de los problemas científicos más apremiantes como lo es la calidad de los aprendizajes.

La informática en el sector educacional ha obtenido grandes logros con la producción de software educativo, la cual desempeña uno de los papeles más importantes en el aprendizaje de los docentes. La incorporación de estímulos visuales y sonoros del software educativo basado en técnicas multimedia, agiliza y dinamiza la enseñanza, logrando que el estudiante se sienta más atraído y que mantenga la atención y el interés, mejorando la retención de la información presentada de forma divertida y diversa.

1.3.1 ¿Qué es Software Educativo?

El software es esencial para trabajar con una computadora, ya que él forma parte de ésta. Es tan importante como el teclado o el monitor, sin él la máquina sería inservible. Existen software de muchos tipos, los cuales se emplean en diversas ocasiones y con fines diferentes. Uno de los tipos de software existentes es el software educativo, que no es más que un “Software diseñado y producido para la docencia o cualquier otro software que, aunque no haya sido diseñado con fines puramente educativos, se esté empleando con el propósito de enseñar y aprender” (Lagos, 2002).

1.3.2 Características esenciales de los programas educativos.

La característica principal de los programas educativos es su gran flexibilidad, ya que éstos pueden tratar los más diversos contenidos, desde los más básicos (Matemática, Historia, Ciencias, etc.), así como la alta interactividad que poseen,

pues permiten un aprendizaje autoguiado y auto-iniciado que contesta inmediatamente a las acciones de los estudiantes. Cada persona va construyendo su conocimiento, bien sea de manera individual o colectiva. Permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.

1.4 Multimedia.

Hoy en día se hace cada vez más común el uso del término multimedia, aunque aún existen personas que creen que la multimedia no es más que un juego de computación, o solo la asocian con la televisión. Este punto de vista no es para nada erróneo, ya que tanto la televisión como muchos juegos, emplean la multimedia, pero esto no quiere decir que no existan otras aplicaciones.

Multimedia es todo aquello que utiliza, conjunta y simultáneamente, diversos medios de comunicación en la presentación de la información, tales como imágenes, animación, vídeos, sonido y texto. Básicamente, multimedia es la cualidad de un sistema o documento que utiliza más de un medio de comunicación al mismo tiempo [VAUGHAN, 1997].

Aunque este concepto es tan antiguo como la comunicación humana, ya que al expresarnos en una charla normal hablamos (sonido), escribimos (texto), observamos a nuestro interlocutor (video) y accionamos con gestos y movimientos de las manos (animación), apenas ahora, con el auge de las aplicaciones multimedia para computadora, este vocablo entró a formar parte del lenguaje cotidiano [VAUGHAN, 1997].

El concepto multimedia incluye dos características esenciales: la interactividad y la integración de diferentes medios en un mismo documento.

1.4.1 Tipos de Multimedia.

Existen diferentes tipos de multimedia como por ejemplo:

Presentación.

La multimedia se puede presentar como la vemos en la televisión o el cine, de forma lineal. No necesita que el usuario o exportador tenga que interactuar con

ella, pues ésta corre de principio a fin toda la presentación. Por lo general, se le llama multimedia de Presentación, pues mayormente se utiliza como material de apoyo para conferencias, exposiciones, en el cine, para promoción de productos etc.

Interactiva.

Cuando la multimedia nos permite interactuar con ella, es decir, que el usuario tenga la posibilidad de explorar a voluntad su contenido, se denomina multimedia interactiva. Ésta permite, mediante hipervínculos, que el usuario interactúe con el sistema. Se puede acceder a la parte que más interese al navegante, sin necesidad de repasar todo el contenido. Puede retroceder o avanzar según los intereses personales, brindando así una libertad plena de navegación.

Estas cualidades son las que hacen que la multimedia interactiva se utilice por lo general en materiales para la diversión y entretenimiento, manuales de entrenamiento, en libros electrónicos, materiales de referencias, etc.

Aplicaciones educativas.

Estas multimedias son tanto Interactivas como Presentaciones. Son programas diseñados con fines educativos para ayudar en el proceso de enseñanza-aprendizaje, permitiéndole al usuario estudiar a su propio ritmo, interactuar con el sistema de modo intuitivo, ya sean tutoriales, materiales de ejercitación o juegos educativos.

Asociados al término multimedia aparecen varios conceptos, como es el caso de Hipermedia e Hipertexto. Estos son conceptos muy relacionados que se confunden con gran facilidad, por lo que a continuación se les ofrecerá una definición de estos términos.

1.4.2 Definición de Hipertexto.

No es más que un documento que se puede leer de forma no secuencial. Se presenta la información de forma dinámica, puede cambiar de lugar, cambiar su presentación, estructurarla en varios niveles, y sobre todo ofrecer diversos caminos de lectura. Son bloques de texto relacionados por secciones, enlaces o

hipervínculos y anclajes, brindando la posibilidad de que el lector pueda elegir el camino a transitar.

El hipertexto es una nueva forma de mostrar la información, uniendo temas con palabras en los textos, permitiendo que el usuario acceda a otros textos relacionados con dicha palabra, sin tener que abandonar el documento para buscar en otros. Esto permite que el lector no se vea en la necesidad de seguir un hilo único y lineal.

1.4.3 Definición de Hipermedia.

La hipermedia combina las técnicas de multimedia e hipertexto, por lo que su definición es una liga entre ambas (multimedia e hipertexto).

Para obtener una hipermedia, solo es necesario unir el hipertexto con gráficos, sonidos, dibujos y videos, ya que solo se diferencian en que el hipertexto muestra la información en forma de texto, y la hipermedia le adiciona imágenes y sonido, pero ambas presentan la información de forma no lineal, uniéndola mediante vínculos.

1.4.4 Multimedia en la educación.

El mundo ha experimentado una gran transformación. Los cambios realizados responden a las demandas de la vida, y el desarrollo tecnológico alcanzado en la actualidad ha creado una nueva era.

Es importante que cada nueva generación esté más preparada para enfrentar los grandes desafíos del futuro, por lo que ha sido necesario imprimirle a la enseñanza un carácter verdaderamente activo.

Las tecnologías de la informática y las comunicaciones son utilizadas de manera creadora como parte indisoluble de la clase. El vídeo, la televisión y la computación, constituyen poderosos medios para lograr una mayor actividad del estudiante, y que ésta sea realmente desarrolladora.

En épocas anteriores, algunas horas bastaban para que los alumnos se apropiaran de los conocimientos necesarios, pero el volumen de información acumulado por la humanidad exige que se busquen nuevas vías que permitan al maestro enseñar y aprender.

Los programas educativos posibilitan una nueva forma de efectuar los procesos de enseñanza-aprendizaje. Constituyen instrumentos muy efectivos para una enseñanza activa. Estos programas brindan una adecuada formación de capacidades y habilidades en los estudiantes, tales como enseñarles a pensar por sí solos, que sean capaces de manejar fuentes de información y crearles las condiciones para que puedan apropiarse de los conocimientos por sí mismos.

El impacto de las tecnologías multimedia toma auge en la educación. Al integrar adecuadamente varios medios (audio, video, gráficas, animación y texto) al mismo tiempo, se mejora notablemente la atención. La multimedia tendrá una función semejante a la de los libros en el aprendizaje, ya que ésta contiene la misma información, solo que de forma más atractiva y dinámica, permitiendo que el usuario adquiera sus conocimientos de manera más efectiva, ofreciéndoles la posibilidad de controlar el flujo de información según su ritmo de aprendizaje.

La tecnología educativa ha facilitado e incrementado en gran medida la educación a distancia. A través de una presentación atractiva, ésta le proporciona al estudiante la información necesaria para el desarrollo de su aprendizaje.

1.5 Tendencias tecnológicas actuales.

1.5.1 ¿Qué es un Entorno Virtual de Aprendizaje (EVA)?

Un Entorno Virtual de Aprendizaje (EVA) consiste en el conjunto de las distintas funcionalidades asociadas a la actividad formativa. EL EVA permite la creación y mantenimiento de comunidades virtuales, proporcionando los servicios con los que cada comunidad se identifica, y que garantizan la integración, enriquecimiento y la fiabilidad de sus usuarios. (www.conocimientosweb.net).

Un Entorno Virtual de Aprendizaje no es más que un sitio donde se puede encontrar documentación, cursos online, tutoriales y otros documentos,

organizados, clasificados y orientados a la enseñanza. Cuenta con un servicio de autenticación y permite el intercambio de información e ideas, facilitando la interacción entre distintos usuarios.

Ejemplos de EVAs utilizados en las Universidades del mundo son: Dokeos, ILIAS, Sakai, Claroline, ATutor, LRN, BlackBoard y Moodle, entre otros. De estos EVAs, el más utilizado es Moodle, porque se adapta mejor a los cursos basados en la interacción entre los participantes.

1.5.2 Moodle:

Moodle es un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea.

Moodle, (Entorno Modular de Aprendizaje Dinámico Orientado a Objetos).

Moodle es utilizado en todo el mundo. En los centros de enseñanza es una herramienta muy útil para estos propósitos, ya que permite colocar los más disímiles contenidos como pudieran ser (multimedia, apuntes, videos, imágenes). Permite fomentar el auto-aprendizaje y el aprendizaje colaborativo. En él se pueden realizar exámenes online y evaluar las tareas de los estudiantes.

En nuestra universidad, desde el 2005, se utiliza el (EVA moodle), el cual ha estado prestando servicio tanto a estudiantes y profesores como a trabajadores del centro. Éste cuenta con cursos para todas las asignaturas que se imparten en la universidad, documentación para los cursos optativos, guías de estudio de diferentes materias y servicios para la auto-evaluación, entre otros, siempre orientados a las necesidades de los usuarios.

1.6 Herramientas para la creación de Software Multimedia.

Existen muchas herramientas que pueden ser utilizadas para la creación de multimedia. A continuación ofrecemos una representación de algunas de éstas y sus principales características.

1.6.1 Macromedia Director.

Es una extraordinaria herramienta para la creación de multimedias, para crear programas que aprovechen todo tipo de información, los que pueden desplegarse en CD/DVD-ROM, quioscos multimedia e Internet.

Es un programa para producir películas ejecutables en Macromedia usando mapas de bits y en programación Lingo. Este lenguaje de programación acelera los tiempos de desarrollo y permite integrar a sus producciones una interactividad de alto nivel y única. Permite generar presentaciones con el uso de la tecnología multimedia ((Adobe Systems Incorporated, 2009)) que, a través de CDs., pueden ser distribuidas. Además, incorpora a las películas disímiles formatos como imágenes JPEG, BMP, PNG, GIF... videos (MOV, AVI...), sonidos (WAV, AIFF...) o animaciones Flash. Este programa incluye además mapa de bits, editores básicos para texto, sonido y vectores.

1.6.2 ToolBook.

Desde su aparición a principios de 1990, ToolBook se ha convertido en una de las aplicaciones preferidas por los expertos constructores de aplicaciones multimedia, pero la misma tiene una gran desventaja y es que esta herramienta tiene gran dependencia de la plataforma Windows.

Esta herramienta permite diseñar una amplia variedad de actividades de distinta índole combinando textos, imágenes, sonido, animación, vínculo a sitios Web y auto-evaluaciones en distintos formatos.

1.6.3 Macromedia Flash 8

Macromedia Flash es una herramienta de autor que utiliza gráficos vectoriales e imágenes, sonido, lenguaje ActionScript, flujo de vídeo y audio bi-direccional. En sentido estricto, Flash es el entorno y el programa de máquina virtual utilizado para ejecutar los archivos generados con Flash.

Macromedia ha trabajado en el desarrollo de flash, convirtiéndola en una herramienta de desarrollo principalmente para la creación de multimedia.

Posee varias características que hacen que su uso sea agradable y que facilitan el trabajo con la misma, tales como:

- Interfaz gráfica amigable, potente y sencilla de usar.
- Soporta vídeo con nuevas funcionalidades.
- Carga dinámica de imágenes, video y sonido.
- Pre-visualización de animaciones.
- Ayuda tanto para la programación como para el diseño de animaciones.
- Incluye componentes ya creados que te pueden ayudar a la hora de hacer tus animaciones.
- Puede interactuar con una base de datos.
- Librería de símbolos.
- Soporte de audio MP3.
- Interacción con otros lenguajes como XML.

Esta herramienta cuenta con un lenguaje propio que es Action Script.

1.6.4 Herramienta a utilizar.

La herramienta seleccionada por el autor es Macromedia flash 8, herramienta que en la actualidad es la más utilizada en el mundo para la creación de multimedia. Sus diferentes versiones, las ventajas en su manejo y la inclusión de diferentes herramientas de producción, la convierten en la mejor opción. Entre las principales características que motivan su uso se encuentran:

La programación con un lenguaje de script, la cual presenta una programación poderosa, lo que permite ampliar el campo de acción del paquete.

Macromedia Flash 8 presenta niveles de integración nunca antes vistos con la familia de aplicaciones creadas por Macromedia (Director MX 2004), que pueden combinar las fuerzas de este poderoso producto, para así construir presentaciones interactivas.

Sus herramientas están basadas en el tiempo. Cada una utiliza su propia interfaz de usuario para manejar los eventos en su tiempo. Muchas utilizan una línea de tiempo para darle secuencias a los eventos.

1.7 Metodologías a considerar para el desarrollo del software.

Las metodologías de desarrollo de software son un conjunto de procedimientos, métodos y técnicas que ayudan a organizar el trabajo y la documentación con el objetivo de facilitar el desarrollo del producto. Es decir, son una guía de pasos a seguir, donde además indican qué persona y qué papel debe de tener para realizar una actividad en el desarrollo del software. Además detallan la información que se debe producir después de terminada una actividad.

Esta tecnología se divide en dos grandes grupos:

- Metodologías Ligeras. (XP (EXtreme Programming) y Crystal Light Methods)
- Metodologías Pesadas. RUP (Rational Unified Process).

Las pesadas se basan en conseguir su objetivo por medio del orden y la documentación, mientras que las ligeras se basan en la comunicación directa e inmediata entre aquellos que intervienen en el proceso.

A continuación se analizan algunas de estas metodologías mostrando sus características.

1.7.1 Proceso Unificado de Rational (RUP).

Es un proceso de desarrollo de software que junto con el Lenguaje Unificado de Modelado (UML), conforma la metodología más utilizada en el mundo tanto para el desarrollo de pequeños proyectos, como para sistemas software complejos, pues está pensada para adaptarse a diferentes áreas de aplicaciones y diferentes organizaciones.

Como una de sus principales características está el hecho de que en cada ciclo de iteración se hace exigente el uso de artefactos.

1.7.2 Programación Extrema (XP).

Esta metodología consiste en una programación rápida, cuya particularidad es tener como parte del equipo al usuario final. Está diseñada para eliminar el problema de desarrollo por encargo, su objetivo es entregar a tiempo el producto que el cliente quiere.

La metodología se divide en tres partes fundamentales:

- Pruebas Unitarias.
- Re-fabricación.
- Programación en pares.

Entre sus principales características podemos mencionar que:

- Los diseñadores y programadores se comunican con el cliente y entre ellos mismos.
- Los diseños se mantienen libres de complejidad o pretensiones excesivas.
- El software es liberado en entregas frecuentes tan pronto como sea posible.
- Los cambios se implementan rápidamente.
- Es usada en proyectos de corto plazo.

1.7.3 Desarrollo Guiado por la Funcionalidad (FDD).

Esta metodología se considera a medio camino entre RUP y XP. Está diseñada para proyectos con tiempo de desarrollo relativamente cortos (menos de un año). Se divide en cinco partes fundamentales.

- Desarrollo de un modelo general.
- Construcción de la lista de funcionalidades.
- Plan de releases en base a las funcionalidades a implementar.
- Diseñar en base a las funcionalidades.
- Implementar en base a las funcionalidades.

Como parte de otras características que posee, están que en el proceso de implementar la funcionalidad se contemplan la preparación y ejecución de pruebas, así como revisiones del código e integración de las partes que componen el software.

Define métricas para seguir el proceso de desarrollo de la aplicación, útiles para el cliente y la dirección de la empresa, y pueden ayudar además para conocer el estado actual del desarrollo y a realizar mejores estimaciones en proyectos futuros.

1.7.4 Selección de la metodología a utilizar.

En este caso en particular se decidió que la metodología más apropiada para el desarrollo de esta aplicación es RUP, por todas las ventajas de organización que nos brinda. Por dividir el trabajo en roles, por ser adaptable a cualquier tipo de proyecto, ya sea grande o sencillo, por ser una metodología orientada a objetos y de conocimiento del autor. Además se tiene como antecedentes que se han construido varias Multimedia con esta metodología.

1.8 Lenguajes de modelado.

1.8.1 UML (Lenguaje Unificado de Modelado) y el Rational Rose.

UML es un lenguaje para visualizar, especificar, construir y documentar los artefactos de un sistema, que involucra una gran cantidad de software (Booch)

Está compuesto por diversos elementos gráficos que se combinan para conformar diagramas y cuenta con reglas para combinar tales elementos.

Sin embargo, desde el punto de vista puramente tecnológico, UML tiene una gran cantidad de propiedades que han sido las que realmente han contribuido a hacer de UML el estándar. Algunas de las propiedades de UML como lenguaje de modelado estándar son:

- Concurrencia: es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por OMG.
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.

Las estructuras más importantes que soportan tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos. Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.

UML no es una guía para el análisis y diseño orientado a objetos, es decir, no es un proceso, es un lenguaje que permite la modelación de sistemas con tecnología orientada a objetos.

En el modelo gráfico de UML se identifican:

- Elementos (abstracciones que constituyen los bloques básicos de construcción).
- Relaciones (Ligan los elementos).
- Diagrama (Es la representación gráfica de un conjunto de elementos. Visualizan un sistema desde diferentes perspectivas).

1.8.2 Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA-L).

Este lenguaje surge como una extensión de UML para la integración de especificaciones de sistemas multimedia, basado en el paradigma orientado a objeto y MVC (Modelo Vista Controlador) para la interfaz de usuario, siendo éste un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos.

OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son:

Vista Lógica: Ésta se modela a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias.

Vista de Presentación espacial: modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que éste último no contiene un diagrama apropiado para esta tarea. Estos diagramas tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización (texto, gráfico, video, animación) e interacción (scrolls, barras de menú, botones, campos de entrada y salida, hipertextos con hipervínculos).

Vista de Comportamiento temporal predefinido: Modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos; marcas de inicio y fin de ejecución que permiten soportar su reusabilidad; marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media; activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.

Vista de Control Interactivo: Modelado a través del Diagrama de Estado. Extendido a partir del diagrama de estado de UML, sintácticamente igual a este último, mas con la diferencia semántica de que en el orden de unir los controles interactivos y predefinidos no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados.

Para la propuesta de solución de este trabajo, se ha decidido utilizar OMMMA-L como extensión del lenguaje UML, basado principalmente en las ventajas que ofrece esta metodología sobre UML.

1.9 Lenguajes utilizados en la multimedia.

1.9.1 El lenguaje XML.

XML, (lenguaje de marcas extensible): Es un metalenguaje extensible de etiquetas. Una simplificación y adaptación del SGML permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto, XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

XML no ha nacido sólo para su aplicación en Internet, sino que se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo y casi cualquier cosa imaginable.

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en la actualidad, ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil, proporcionando numerosas ventajas a los programadores y usuarios:

- Búsquedas con más significado
- Programación de aplicaciones flexibles
- Integración de datos procedentes de fuentes dispares
- Computación y manipulación locales de los datos
- Varias vistas de los datos
- Actualizaciones granulares

XML proporciona interoperabilidad mediante un formato flexible y abierto basado en estándares, con formas nuevas de acceso a las bases de datos existentes y de entregar datos a clientes de la Web. Las aplicaciones se pueden generar más rápidamente, su mantenimiento es más sencillo, ofreciendo fácilmente varias vistas de los datos estructurados.

1.9.2 ActionScript.

ActionScript es un lenguaje de programación orientado a objetos para crear scripts en Flash. Mediante el amplio uso del lenguaje de programación interno, ayuda a reducir enormemente el tamaño de los archivos y películas.

Programar en ActionScript tiene varias ventajas para el diseñador de multimedia, ya que proporciona efectos que no son alcanzables con el simple diseño. Algunas de estas ventajas son:

- Controlar el tamaño del archivo, reduciendo su tamaño a una pequeña parte.
- Controlar la línea de tiempo de la película principal, de sus clips y de las películas que se encuentren en un paquete.
- Establecer el máximo grado de interactividad entre el usuario y la película.
- Controlar los videos y los sonidos desde la programación en tiempo real.
- Crear objetos y eliminarlos a voluntad, haciendo que su aparición en pantalla sea siempre agradable y sorprendente.

- Controlar la comunicación con el servidor.
- Realizar dibujos sofisticados con unas líneas de código, incluidos los degradados.
- Realizar películas que comuniquen a diferentes páginas Web, incluidas las ASP y JSP. Libros de visita.
- Juegos interactivos y sofisticados.
- Inclusión de documentos XML.

Otra de las grandes ventajas de ActionScript en Flash es que se puede ejecutar un dibujo enteramente en el reproductor, enviando solamente unas líneas de código. ActionScript no admite objetos específicos de navegador como Ventana, Documento o Ancla. Estos objetos están fuera del "escenario" de la película. No obstante, si el programador quiere referirse en alguna ocasión a estos objetos, dispone en ActionScript de una función especial FSCommand que le permitirá acceder a ellos.

Conclusiones del Capítulo 1:

En la realización de esta etapa del trabajo hemos abordado los principales conceptos que de una forma u otra se relacionan con el mismo, como han sido las definiciones de:

- Definición de calidad y riesgo.
- Qué es un (EVA) y su versión más utilizada en el mundo Moodle.
- La Multimedia y sus principales características.
- Características esenciales de la educación.

Partiendo de estos análisis y estudiando la situación problemática planteada, se propone realizar una multimedia para que contenga toda la información relacionada al proceso de gestión de riesgo de la calidad del software.

También se hizo un estudio de las principales tecnologías, metodologías y herramientas de diseño a considerar para el desarrollo de la aplicación, proponiéndose como metodología a utilizar RUP y la herramienta seleccionada es Flash 8.

Capítulo 2: Descripción de la Solución.

Introducción.

En este capítulo se presentan las características del sistema que se proponen para dar solución al problema existente. Se da una breve descripción del contenido de la multimedia. Se definen conceptos y se identifican entidades que se relacionan en un modelo de dominio, el cual permite determinar los requisitos necesarios que debe cumplir la aplicación para cubrir las necesidades que lo originan.

2.1 Identificación de la audiencia.

Esta multimedia está dirigida a todas las personas interesadas en obtener un conocimiento referente a la Gestión de Riesgos en el desarrollo del software, principalmente a los estudiantes de la Universidad de las Ciencias Informáticas (UCI), con el objetivo de que contribuya como complemento educativo para la asignatura de Ingeniería de Software que se imparte en el tercer año de la carrera y para los que reciben el curso del perfil Gestión de Riesgos.

2.2 Especificación del contenido.

El contenido de la multimedia “Gestión de Riesgos” cuenta con 7 temas divididos en subtemas, mapas conceptuales con los principales conceptos asociados al curso, evoluciones que permiten comprobar el conocimiento adquirido, una biblioteca con algunos trabajos realizados en la UCI sobre gestión de riesgo. También cuenta con una ayuda para facilitar la navegación en la multimedia.

A continuación se da una breve descripción del contenido de cada uno de los 7 temas:

Tema1: Introducción.

En este tema se da una introducción sobre lo que es la gestión de riesgos, sus principales componentes y las ventajas que nos brinda, además cuenta con dos subtemas que son:

- Estrategia de riesgo Reactiva y Proactiva.
- Riesgos del software.

Tema2: Planificación de la Gestión de Riesgo.

Este tema trata de cómo llevar a cabo una buena planificación del riesgo y las ventajas que ofrece en el transcurso del proceso de Gestión de Riesgos en un proyecto.

Tema3: Identificación del riesgo.

En este tema se trata la identificación del riesgo, los diferentes tipos de riesgos, las sub-categorías genéricas, la creación de una lista de comprobación, y los subtemas que aborda:

- Componentes y controladores del riesgo.
- Técnicas y Herramientas de Identificación.

Tema4: Proyección del riesgo.

Este tema aborda la proyección del riesgo o estimación del riesgo como también se le conoce; cómo medir estos riesgos y su impacto en el proyecto mediante la creación de una tabla de riesgos. También cuenta con los subcapítulos:

- Evaluación del impacto del riesgo
- Evaluación del riesgo

Tema5: Refinamiento del Riesgo.

En este tema se aborda el refinamiento del riesgo; cómo poder refinar un riesgo en un conjunto de riesgos más detallados, y por lo tanto más fácil de reducir, supervisar y gestionar.

Tema6: Reducción, supervisión y gestión del riesgos.

En este tema se trata de cómo reducir, supervisar y gestionar los riesgos del proyecto, y como desarrollar una estrategia para tratarlos. Cuenta con dos subtemas:

- Riesgos y peligros para la seguridad.
- El plan RSGR (Reducción, Supervisión y Gestión de Riesgos).

Tema7: Gestión de Riesgo en la etapa de Ingeniería de Requisitos.

Este tema trata sobre el tratamiento que se le debe dar a la gestión de riesgo en la etapa de Ingeniería de Requisitos. Cuenta con una breve introducción y los siguientes subtemas:

- Identificación del riesgo.
- Evaluación de los riesgos.
- Planificación de riesgos.
- Supervisión de los riesgos.

2.2.1 Mapa Conceptual.

Es una técnica usada para representar gráficamente un conocimiento. Consiste en una red de conceptos donde los nodos representan los conceptos, y los enlaces las relaciones entre ellos.

A continuación se muestran en las figuras (2.1, 2.2, 2.3, 2.4) los mapas conceptuales confeccionados a partir del curso de Análisis y Gestión de los Riesgos, (Creado por Rafael Menéndez) donde se muestra la relación entre los principales conceptos del curso.

2.2.2 Mapa conceptual General.

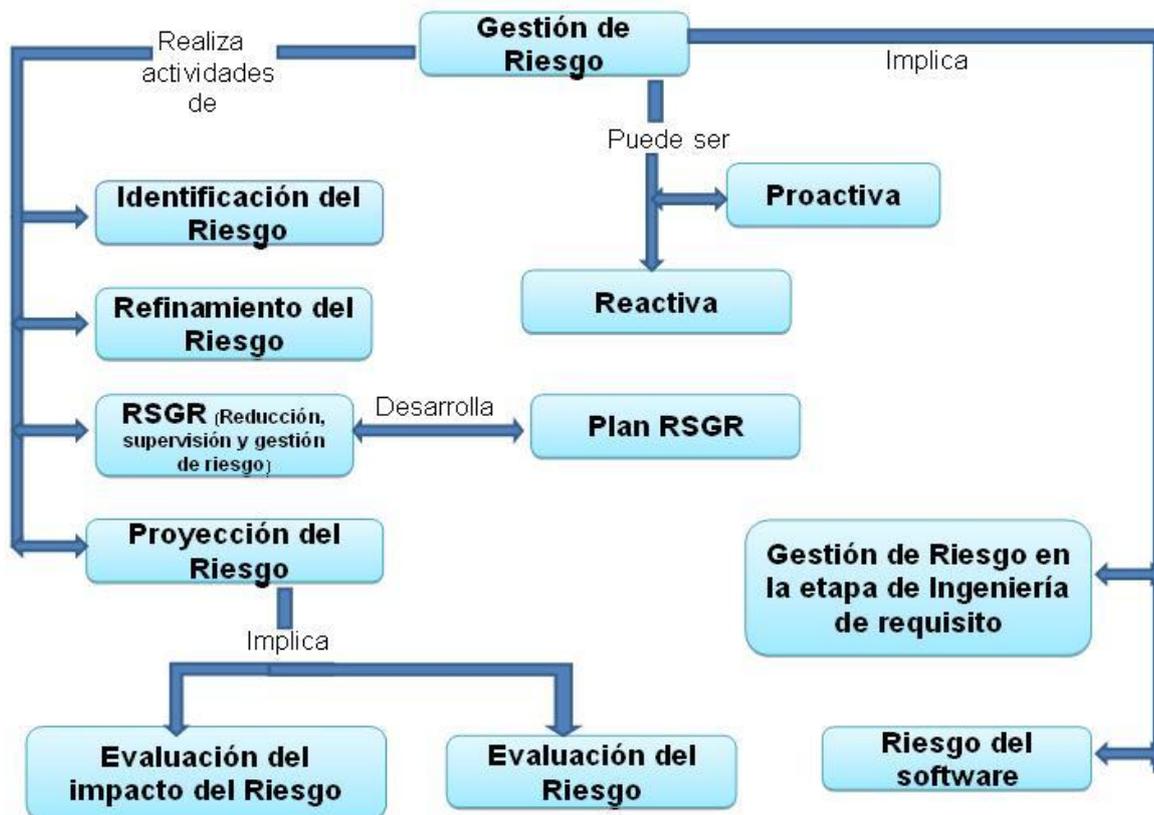


Fig. 2.1 Mapa conceptual general de Gestión de Riesgo.

2.2.3 Sub-mapa de Riesgo del software.

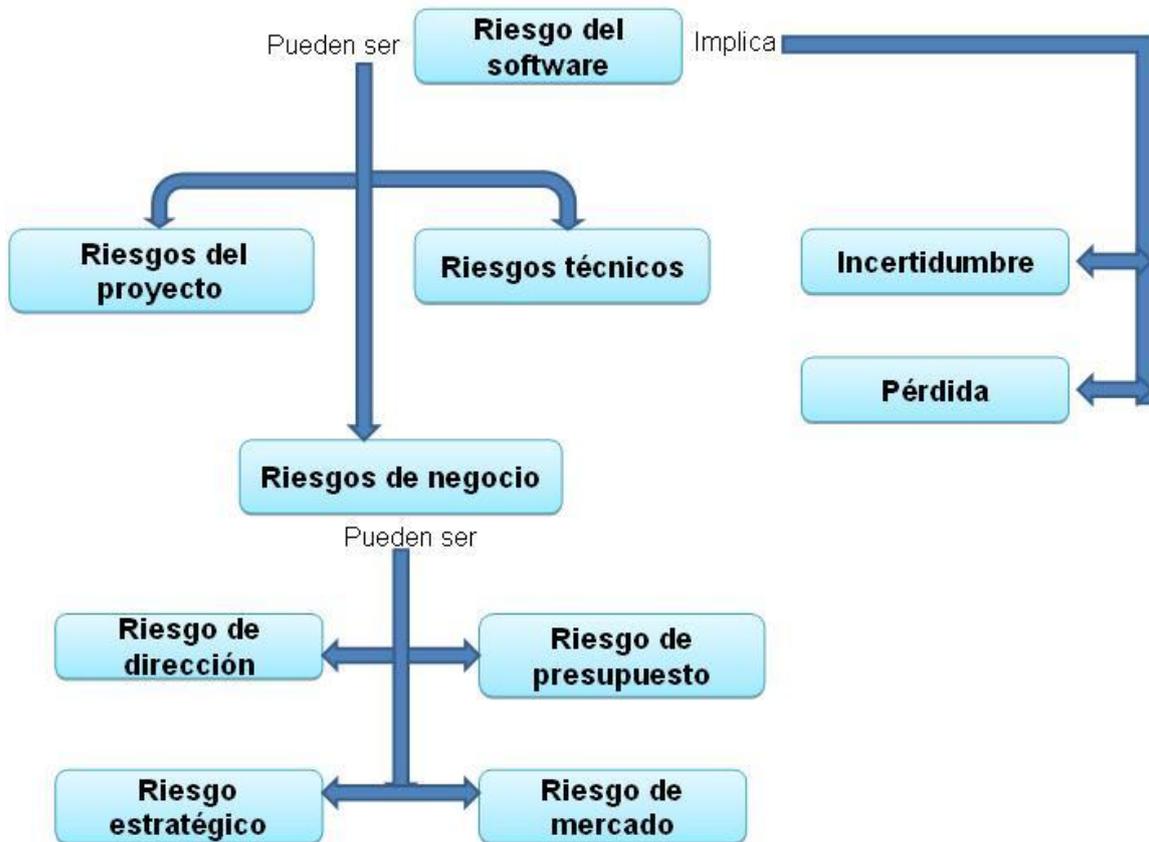


Fig. 2.2 Sub-mapa conceptual de Riesgo del software.

2.2.4 Sub-mapa de Identificación del Riesgo

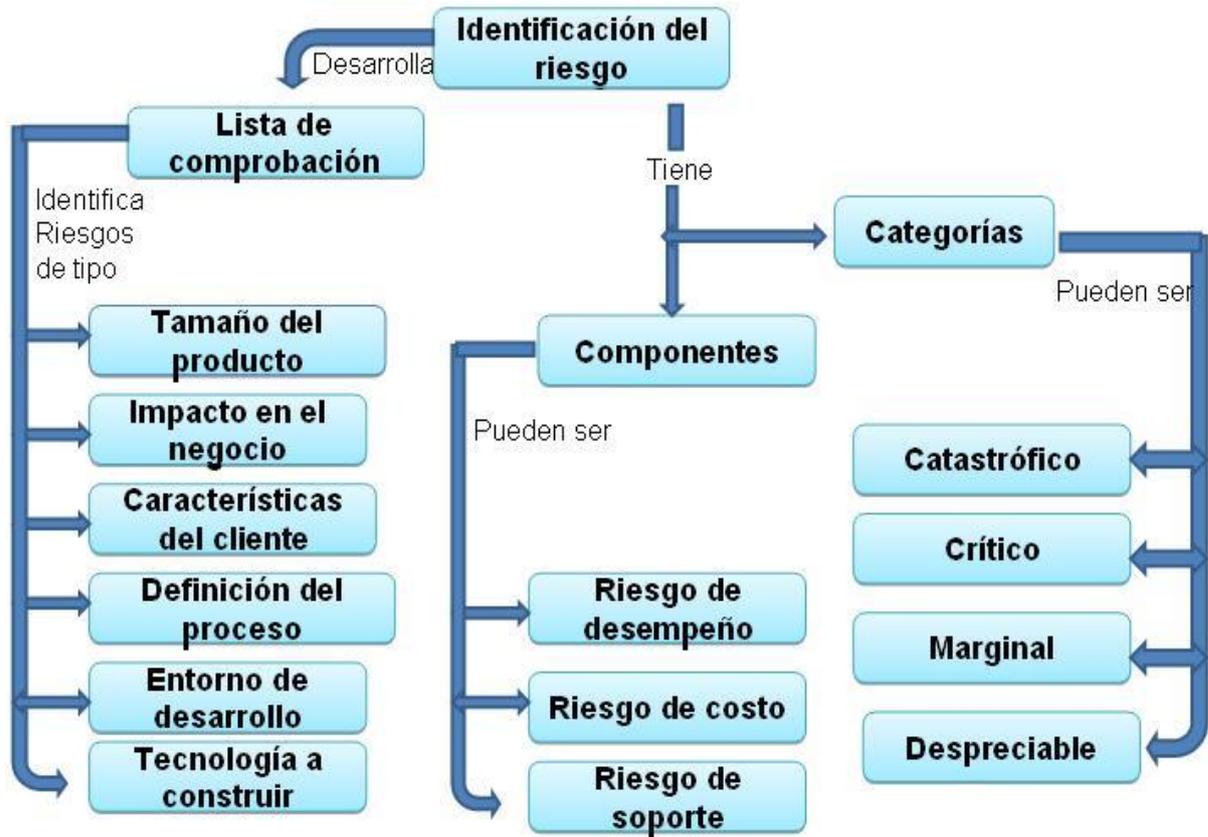


Fig. 2.3 Sub-mapa conceptual de la actividad de Identificación del Riesgo.

2.2.5 Sub-mapa de Gestión de Riesgo en la fase de Ingeniería de Requisitos.

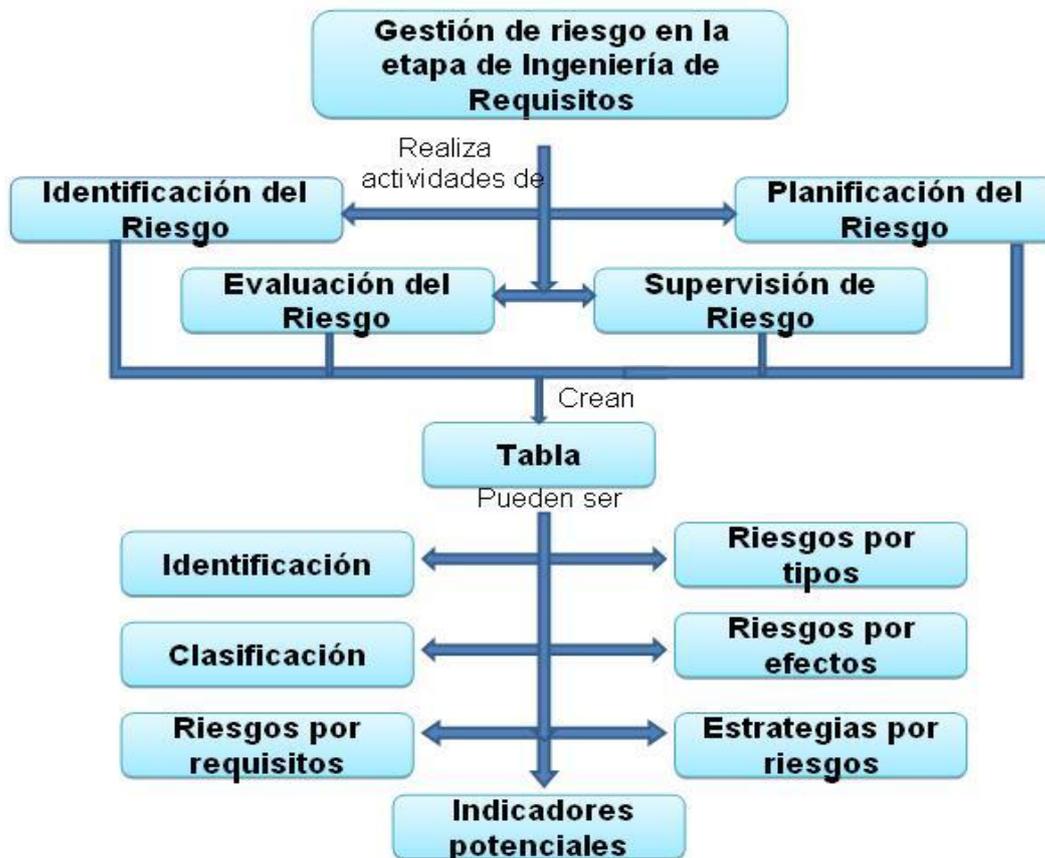


Fig. 2.4 Sub-mapa conceptual de la Gestión de Riesgo en la fase de Ingeniería de Requisitos.

2.3 Descripción del Modelo de Dominio.

Debido a que en la metodología del proceso de desarrollo del Software no está bien definido el negocio en la fase de inicio, se plantea el modelo de dominio mediante un diagrama de clases, siendo necesario identificar todos los conceptos que se utilizarán en el diagrama mediante un glosario de términos:

El usuario: Es aquella persona que posee los privilegios de acceder a la aplicación.

Juego: Es un objeto que, de una forma dinámica y entretenida, comprueba el conocimiento adquirido en el curso.

Tema: Se denomina a cada una de las unidades de contenido en que se divide el contenido del curso.

Mapa conceptual: Es la forma de representar varios conceptos relacionados entre sí.

Trabajos Realizados: Son trabajos que se han realizado en la UCI referentes al tema Gestión de Riesgo.

Autoevaluaciones: Son un conjunto de preguntas que comprueban el conocimiento adquirido en el tema.

Epígrafes: Son las secciones en las que están divididos los temas.

Imagen: Es la representación o gráfico de una figura o proceso.

Palabras Calientes: Es un hipertexto o palabra que, al accionar sobre ella, te muestra su significado o concepto más detallado.

2.3.1 Diagrama de clases del Modelo de Dominio.

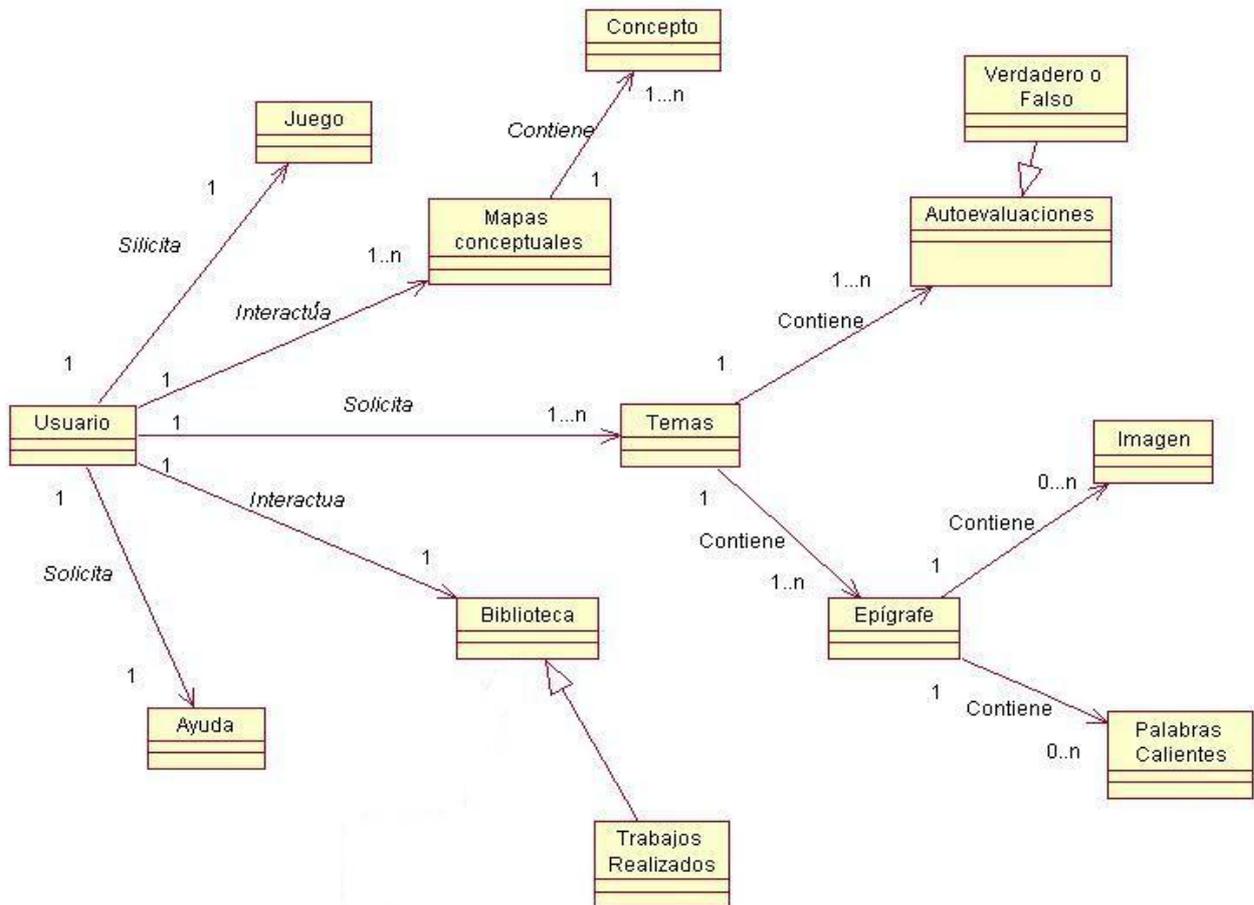


Fig. 2.5 Modelo de Dominio

2.4 Diagramas de Navegación.

Con la creación del mapa de navegación se trata de lograr un mejor entendimiento de la forma en que nos podemos mover por la aplicación.

El mapa se divide en un modelo de navegación general, en sub-modelos de navegación Temas, y Sub modelo de navegación Biblioteca, que se muestran a continuación en las figuras (2.6, 2.7, 2.8).

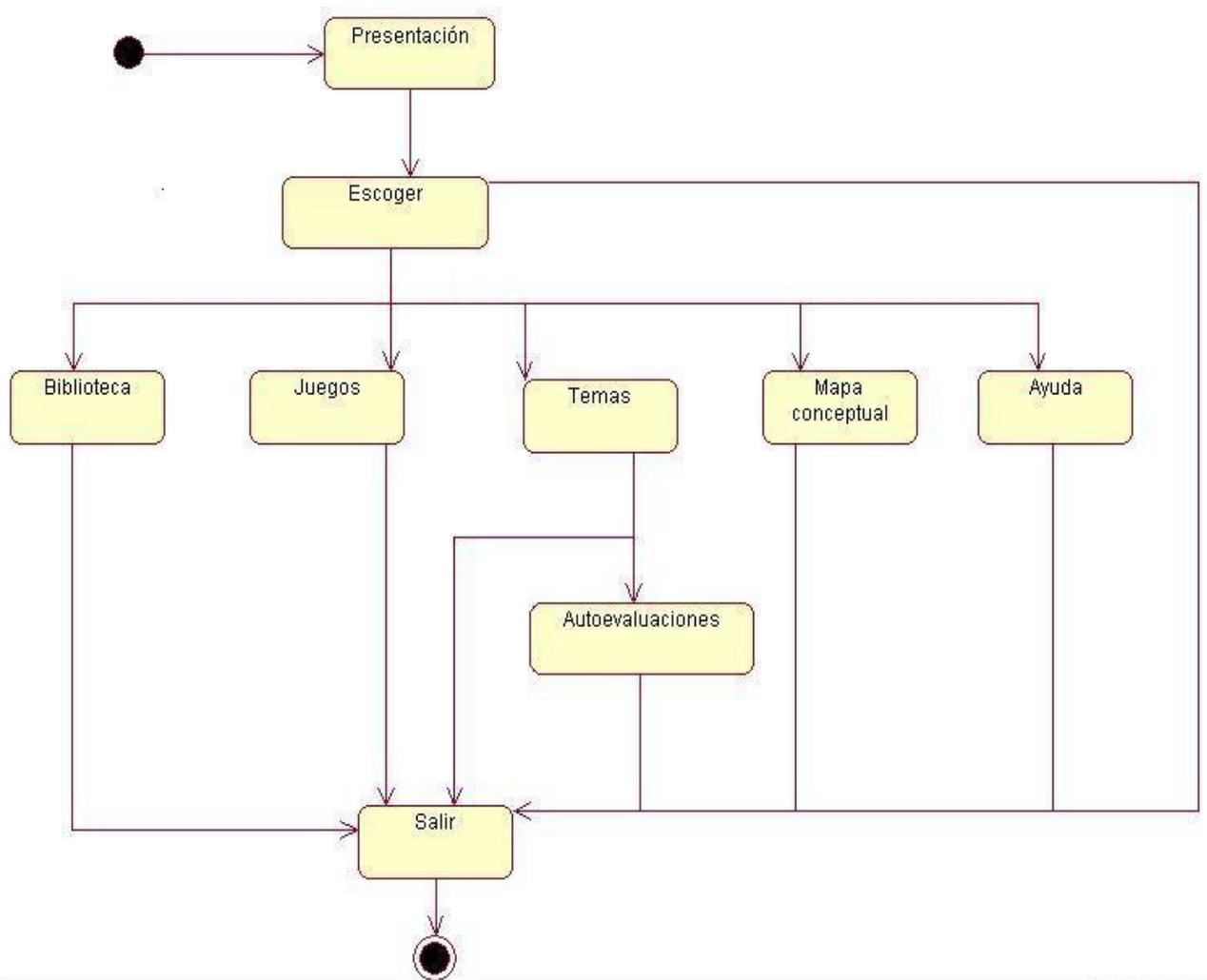


Fig. 2.6 Modelo de navegación general

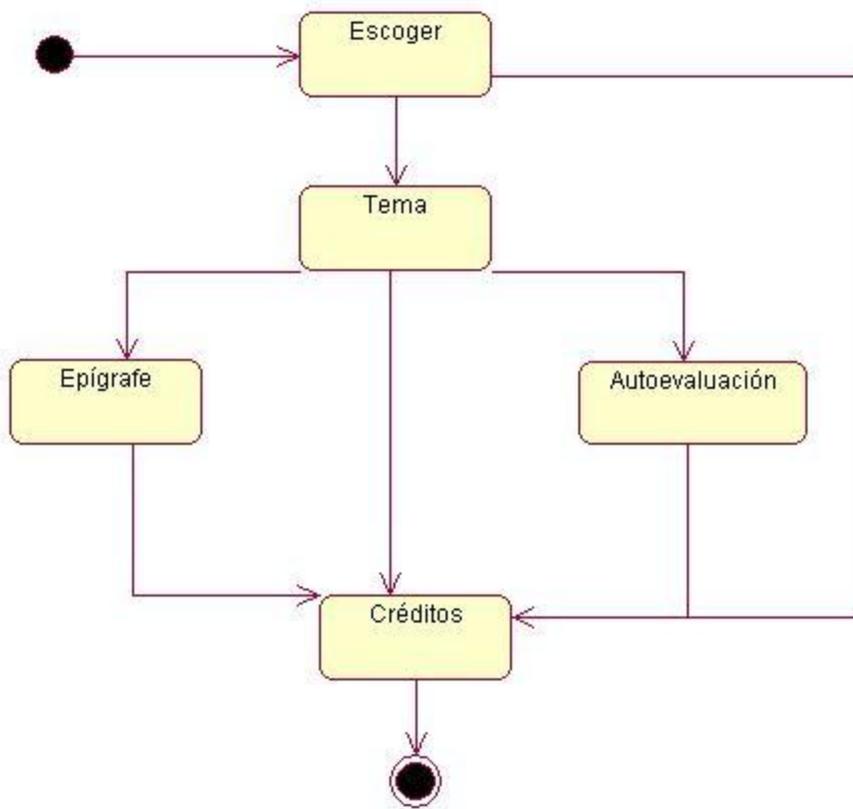


Fig. 2.7 Sub-modelos de navegación Temas



Fig. 2.8 Sub modelo de navegación Biblioteca.

2.5 Requerimientos Funcionales.

Los requerimientos funcionales son capacidades o condiciones que el sistema debe cumplir. Los requisitos funcionales que identifican este sistema son:

1. Mostrar pantalla principal de la multimedia.
2. Permitir la navegación por cada uno de los 7 temas y subtemas que componen la multimedia.
3. Mostrar el contenido de cada uno de los 7 temas y subtemas que componen el producto.
4. Mostrar el texto del contenido del subtema seleccionado por el usuario.
5. Mostrar en los casos requeridos imágenes asociadas al contenido que se muestra.
6. Mostrar palabras claves o calientes que aparecen en el contenido de la multimedia.
7. Mostrar, a través de ventanas emergentes, el concepto o definición de la palabra caliente seleccionada por el usuario.
8. Permitir, a través de ejercicios interactivos, comprobar los conocimientos adquiridos.
9. Permitir ver la respuesta correcta del ejercicio una vez que se haya terminado de responder.
10. Permitir el acceso a la sección biblioteca.
11. Permitir la navegación por cada una de las opciones (trabajos realizados) que componen la sección biblioteca.
12. Mostrar el mapa conceptual.
13. Mostrar el contenido del glosario de términos del mapa conceptual a partir de la selección efectuada por el usuario.
14. Permitir al usuario ejecutar el juego.
15. Mostrar el contenido de la ayuda
16. Permitir la salida de la aplicación desde cualquier pantalla en que se encuentre el usuario cuando ésta sea solicitada.
17. Mostrar créditos una vez que el usuario desea salir.

2.6 Requisitos no funcionales.

Los requerimientos no funcionales son propiedades o cualidades que hacen al producto atractivo, usable, rápido o confiable. Los requisitos no funcionales que identifican este sistema son:

2.6.1 Hardware.

Los requisitos para la ejecución de la aplicación son: Procesador Intel Pentium III de 800 MHz (o equivalente) y versiones posteriores y 128 de RAM. Pantalla de 16 bits de 1024 x 768 (se recomienda de 32 bits) y 710 MB de espacio en disco disponible.

2.6.2 Usabilidad.

Los usuarios que utilizarán el sistema deberán tener conocimiento básico del manejo de la computadora, así como del trabajo con sistemas operativos visuales.

La aplicación deberá poseer una interfaz y navegación asequibles y funcionales, tanto para usuarios expertos como para los que no tienen conocimientos profundos de informática.

2.6.3 Software.

Tabla 1: Sistemas Operativos y navegadores.

Windows	
Plataforma	Navegador
Microsoft® Windows® Vista	Microsoft Internet Explorer 7, Firefox 2.0, AOL 9, Safari 3.x o superior
Microsoft Windows XP	Microsoft Internet Explorer 6.0 o superior, Firefox 1.x, Firefox 2.x, Mozilla 1.x o superior, Netscape 7.x o superior, AOL 9, Opera 7.11 o superior, Safari 3.x o superior.
Windows Server® 2003	Microsoft Internet Explorer 6.0 o superior, Firefox 1.x, Firefox 2.x
Windows 2000	Microsoft Internet Explorer 5.x, Firefox 1.x, Firefox

	2.x, Mozilla 1.x, Netscape 7.x o superior, AOL 9, Opera 7.11 o superior
Windows Me	Microsoft Internet Explorer 5.5, Firefox 1.x, Mozilla 1.x, Netscape 7.x o posterior, AOL 9, Opera 7.11 o superior
Windows 98	Microsoft Internet Explorer 6.0 o superior, Firefox 1.x, Mozilla 1.x, Netscape 7.x o superior, Opera 7.11 o superior
Linux	
Plataforma Navegador Red Hat® Enterprise Linux® (RHEL) 3 actualización 8, RHEL 4 actualización 4 (AS/ES/WS)	Firefox 1.5.0.7 o superior; Mozilla 1.7.x o superior; SeaMonkey 1.0.5 o superior.
Novell SUSE 9.x o 10.1	Firefox 1.5.0.7 o superior; Mozilla 1.7.x o superior; SeaMonkey 1.0.5 o superior.

2.6.4 Apariencia.

El sistema tendrá una interfaz sencilla, intuitiva y amigable para sus usuarios.

Utilizar botones que expresen su función, ya sea que se intuya o expresados con texto.

El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

La aplicación debe utilizar como idioma principal el Español, excepto aquellas palabras técnicas que no puedan ser traducidas.

La opción salir de la aplicación estará disponible desde cualquier parte de la aplicación, haciendo clic sobre ella se saldrá de la misma.

2.6.5 Requisitos de diseño e implementación.

La herramienta de desarrollo de la aplicación será Macromedia Flash 8.0 y Macromedia Fireworks 8.

El lenguaje de programación será Action Script 2.0.

El contenido se cargará desde archivos XML.

2.6.6 Requisitos de soporte.

La aplicación es extensible a plataformas Web sin alterar de algún modo el contenido de sus datos.

Para su correcto funcionamiento, la computadora donde se ejecute la multimedia deberá tener tarjeta de video, tarjeta de sonido y demás aditamentos para la reproducción de sonido.

2.7 Modelo de casos de uso.

El usuario es el único actor que interactúa con la aplicación, iniciando todos los casos de usos del sistema. En la figura No.9 se observan los casos definidos para representar el flujo de los eventos.

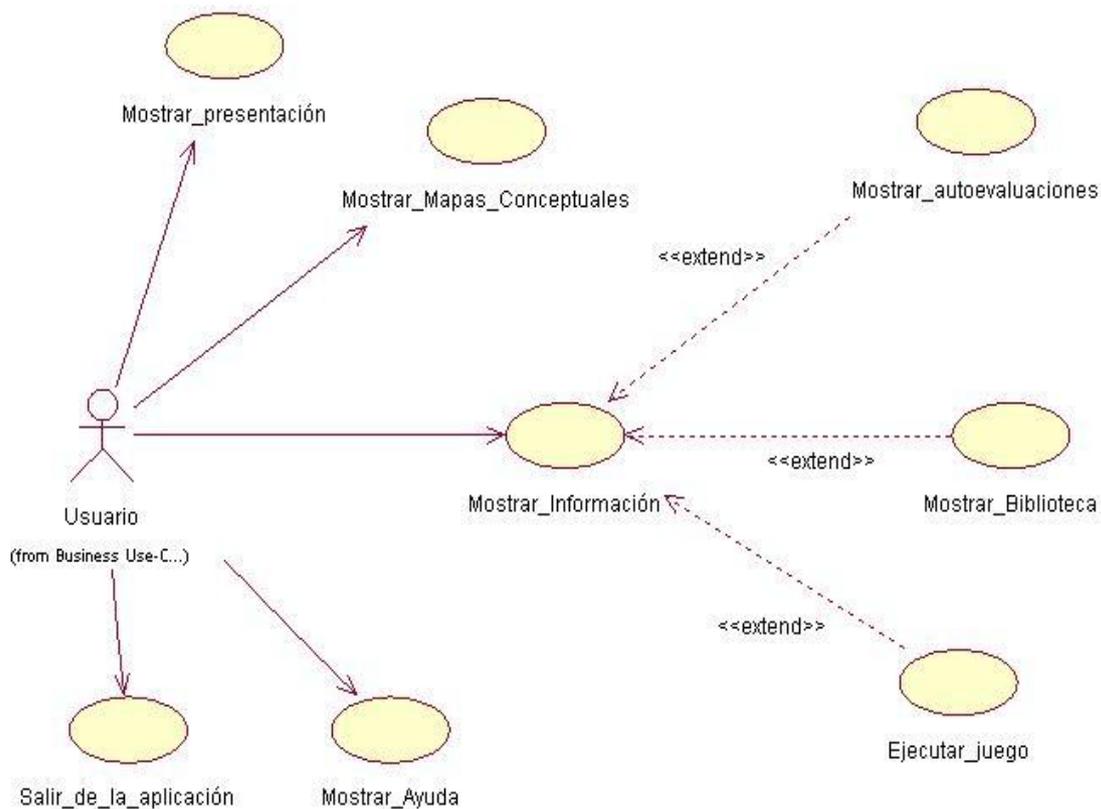


Fig. 2.9 Modelo de Caso de Uso

2.7.1 Actor presente en el sistema:

En la siguiente tabla se muestra el principal actor que interactúa con el sistema.

Actor	Descripción
Usuario	Es la persona que va a usar el sistema para buscar información referente al curso "Gestión de Riesgo".

2.7.2 Breve descripción de los casos de uso del sistema:

Caso de Uso:	Mostrar presentación de la multimedia
Autor(es):	Usuario
Propósito:	Mostrar presentación del producto
Resumen:	Al ejecutar la aplicación, se muestra la presentación del producto. El usuario puede detener la presentación si lo desea al presionar el botón que indica Saltar Presentación, pasando automáticamente a la pantalla principal.
Referencia:	R(1)

Caso de Uso:	Mostrar Información
Autor(es):	Usuario
Propósito:	Mostrar la información contenida en los temas, mapas conceptuales, evaluaciones, ayuda y en la biblioteca.
Resumen:	El usuario solicita la información, ya sea de los temas, mapas conceptuales, evaluaciones etc. Al accionar el botón, el sistema se encarga de mostrar la información solicitada.
Referencia:	R(2,3,4,5,6,7,13)

Caso de Uso:	Mostrar Mapas Conceptuales
Autor(es):	Usuario
Propósito:	Mostrar los mapas conceptuales con los principales conceptos asociados al contenido de la multimedia.
Resumen:	El usuario puede recorrer los diferentes mapas y ver cada uno de los conceptos, haciendo clic sobre los nodos. La información se muestra en una ventana flotante.
Referencia:	R(12)

Caso de Uso:	Mostrar Biblioteca
Autor(es):	Usuario
Propósito:	Mostrar el contenido de la biblioteca, que cuenta con una serie de trabajos realizados en la universidad.
Resumen:	Al usuario accionar el botón de Biblioteca, el sistema debe mostrar el contenido de la misma, permitiendo al usuario escoger el trabajo que sea de su interés.
Referencia:	R(10,11,)

Caso de Uso:	Ejecutar juego
Autor(es):	Usuario
Propósito:	La ejecución del juego disponible en la aplicación.
Resumen:	<p>El usuario selecciona el botón de juego. A aparece una pantalla con dos opciones: un crucigrama y una Sopa de Letras.</p> <p>Si opta por el crucigrama, aparece una serie de preguntas indicando en las verticales y horizontales las palabras correspondientes. Después de llenadas las casillas por el usuario, se podrá comprobar si la respuesta es correcta o no oprimiendo el botón comprobar. Si es incorrecto, le da la opción de comenzar nuevamente.</p> <p>Si su opción es la Sopa de letras, aparece una caja de letras donde el usuario debe localizar un conjunto de 8 palabras dadas. Las palabras se marcan dando clic en la primera y ultima letra. Al encontrar todas las palabras, sale un cartel de felicitación.</p>
Referencia:	R(14)

Caso de Uso:	Mostrar autoevaluaciones
Autor(es):	Usuario
Propósito:	Mostrar y ejecutar los ejercicios de autoevaluación de cada uno de los temas estudiados.
Resumen:	Después del usuario haber estudiado un tema, puede acceder a la autoevaluación, donde debe ir contestando las preguntas presentadas por el sistema. El mismo después le dará la cantidad de preguntas correctas e incorrectas.
Referencia:	R(8,9)

Caso de Uso:	Mostrar Ayuda
Autor(es):	Usuario
Propósito:	Mostrar la ayuda para una mejor comprensión del sistema.
Resumen:	El usuario solicita la ayuda. La aplicación le muestra el contenido de la misma, la cual contiene el Modelo de navegación general.
Referencia:	R(15)

Caso de Uso:	Salir de la aplicación
Autor(es):	Usuario
Propósito:	El usuario puede salir de la aplicación en cualquier momento.
Resumen:	Cuando el usuario solicita salir de la aplicación, el sistema le muestra un cartel de confirmación. Si el usuario no accede, vuelve a la pantalla en que se encontraba, de lo contrario, se muestran los créditos.
Referencia:	R(16,17)

Conclusiones del Capítulo 2:

Con la culminación del presente capítulo se describe la propuesta de solución a través del modelo de dominio, así como la especificación de los requisitos funcionales y no funcionales que debe cumplir el sistema a desarrollar. Se especificó la navegabilidad del producto a través de los diagramas de estructura de navegación. Concluidas estas funciones, ya se puede proceder al diseño e implementación de la aplicación, tratando de que se cumplan todos los requisitos y las funciones que se han considerado necesarias en este capítulo.

Capítulo 3: Solución de la Descripción Propuesta.

Introducción.

En el presente capítulo se analizan los principios de diseño a considerar en la creación del producto multimedia, mediante la realización de los **diagramas de presentación**. Éstos son un nuevo artefacto incorporado dentro del lenguaje UML a partir de la extensión del mismo para aplicaciones multimedia, planteada por OMMMA-L y que forman parte del modelo de diseño. Ellos brindan información sobre la estructura que tendrán las pantallas del sistema. Los diagramas de componentes dan a conocer la relación entre los archivos físicos que intervienen en el funcionamiento del sistema, que se muestran como parte de la implementación del producto.

3.1 Principios de Diseño.

El diseño de la multimedia implica una serie de elementos a considerar, teniendo en cuenta qué es lo que motiva al usuario a interactuar con dicho sistema. Su éxito depende de la calidad de las interfaces. El mismo cuenta con 8 principios a considerar:

1- Precedencia: Uno de los más importantes. Lograr que el lector mire a donde deseamos. Ejemplo de esto sería cuando ejecutamos una aplicación y lo primero que miramos es el logo.

2- Espacio: El espacio no es lugar perdido, ya que muchas veces hace que todo se vea ordenado y claro y por ende, más atractivo.

3- Navegación: Idear convenientemente, donde el lector podrá dirigirse por medio del menú de navegación, donde ubicarlo, y demás detalles a tener en cuenta para que cumpla su tarea.

5- Tipografía: Tener en cuenta que la elección de la letra, sus tamaños, espacio, color, etc., deben ser acorde a los colores a utilizar en cada unas de las pantallas donde se mostrarán los textos, teniendo en cuenta que el tipo de letra a utilizar permitirá al estudiante ver con mayor claridad y eficiencia las letras del texto.

6- Usabilidad: La aplicación debe ser lo más cómodamente accesible para nuestros lectores, logrando que el usuario se sienta cómodo y recorra la aplicación completamente, sin dificultades.

7- Alineación: Lo espléndido es no combinar varios tipos de alineación para mantener un orden lógico en el diseño.

8- Claridad: Lograr que nuestras imágenes no se vean pixeladas, que las fuentes se vean bien definidas, etc.

9- Consistencia: Que todo combine, las tipografías, los colores, los formatos. Todo se ve mejor cuando se encuentra dentro de un orden, así que por qué no buscarlo.

3.2 Diagramas de Presentación del Modelo de Diseño.

Los diagramas de presentación ofrecen una idea de cómo quedarán estructuradas las diferentes pantallas del sistema. Es un artefacto nuevo dentro del lenguaje UML, incorporado a éste a partir de la extensión del mismo planteada por OMMMA-L. Los mismos muestran los componentes que posee cada pantalla, lo que facilita el entendimiento de la estructura de la interfaces del sistema.

3.2.1 Diagrama de Presentación General

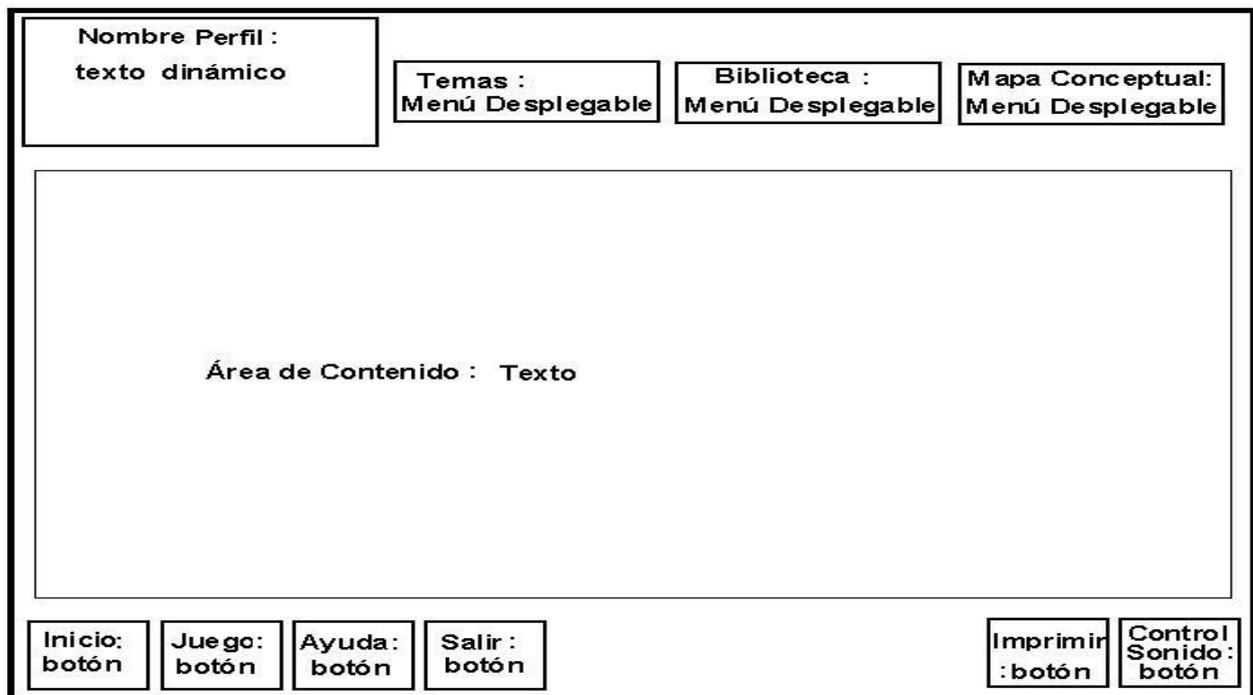


Fig. 3.1 Diagrama de Presentación General.

3.2.2 Diagrama de Presentación Autoevaluación

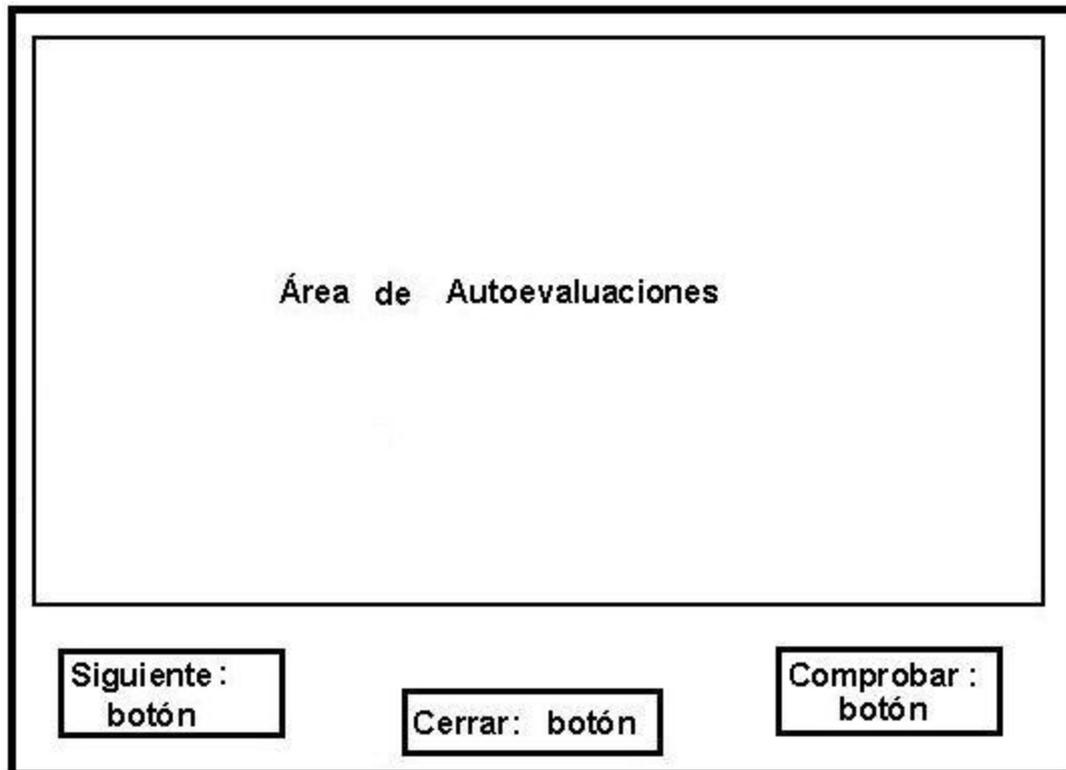


Fig. 3.2 Diagrama de Presentación Autoevaluación.

3.2.3 Diagrama de Presentación Juegos.



Fig. 3.3 Diagrama de Presentación Juegos.

3.2.4 Diagrama de Presentación Salir.

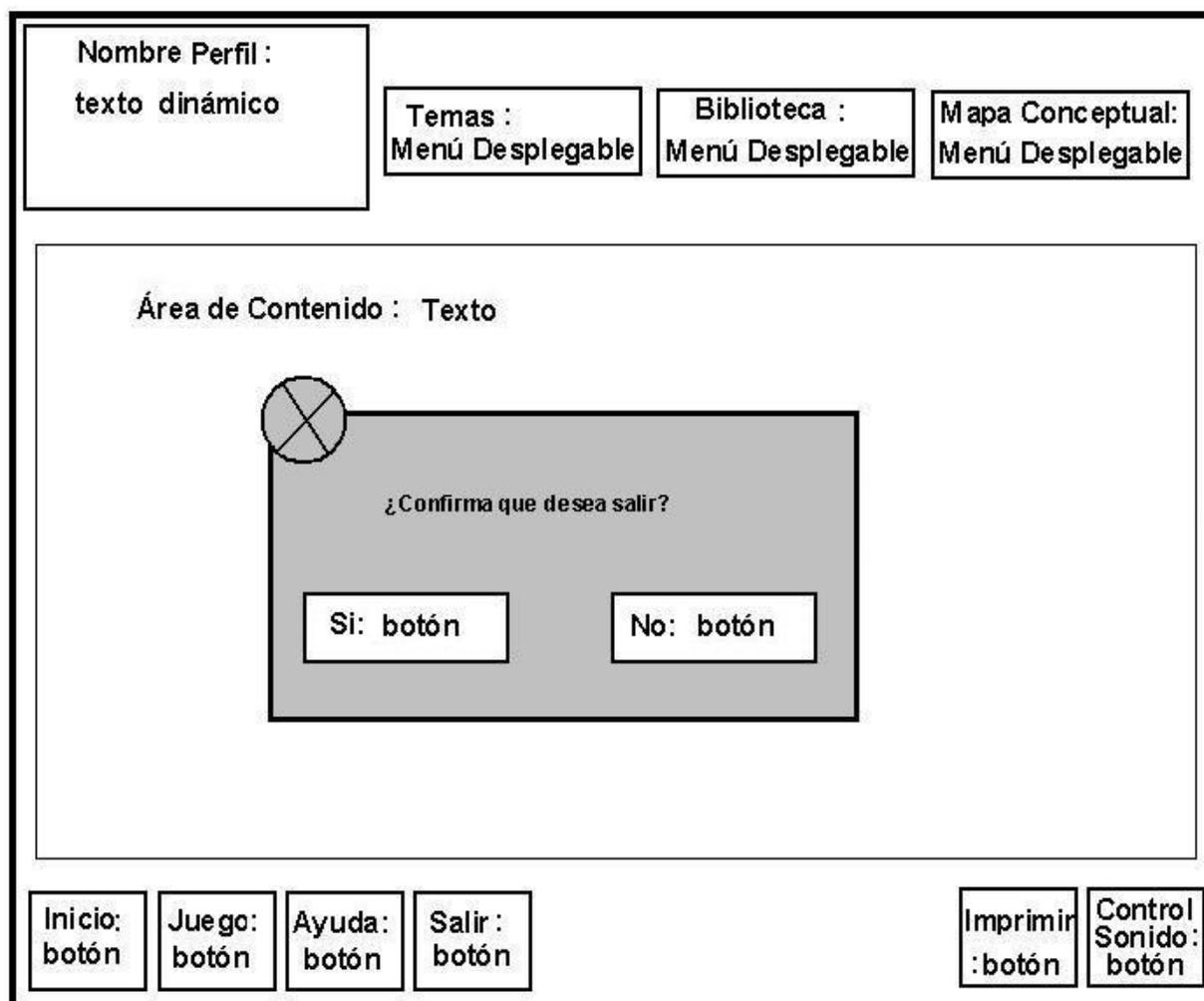


Fig. 3.4 Diagrama de Presentación Salir.

3.3 Descripción y Estructura de Principales Archivos XML.

Para almacenar los textos e imágenes de la multimedia se utiliza el metalenguaje XML, permitiendo la realización de modificaciones sin necesidad de interactuar con el diseño de la aplicación.

Archivos XML de Temas.

Para cada uno de los 7 temas que se tratan en la multimedia se crea un archivo XML con la siguiente estructura:

<navegacion>

<nombre del tema>

<![CDATA[<p class='e3'>se comienza un párrafo con un tipo de letra determinada anteriormente en una clase llamada estilo a la que se hace referencia, para facilitar el trabajo.

se pone el título en otro formato de letras, sin afectar el resto del párrafo

 no permite al igual que en HTML, cambiar de línea.

permite que se muestren palabras en negrita

</p>]]> se sierra el párrafo.

</nombre del tema>se sierra la etiqueta del tema.

<Sub_tema>se crea una nueva etiqueta para almacenar la información de los subtemas, se utiliza nuevamente la etiqueta <p class='e3'>, además:

esto nos permite cargar una imagen, pasando la dirección url, y las dimensiones.

</Sub_tema>se cierra la etiqueta de los subtemas.

</navegacion>se cierra la navegación.

Esto se repite para cada uno de los temas y subtemas, al igual que para la ayuda, y los conceptos de los mapas conceptuales.

Archivos XML del Crucigrama.

Para no correr el riesgo de que un usuario pueda ver las respuestas del crucigrama o cambiarlas el siguiente archivo se crea dentro del script de flash.

mi_xml.parseXML('<?xml version="1.0" encoding="iso-8859-1"?>' se parseo el xml, lo meto en una variable objeto de tipo xml

+'<acrostico>'

+'<palabra>'

+'<texto>se escribe la palabra</texto>'

+'<direccion>se especifica si la palabra es Horizontal o Vertical</direccion>'

+'<fila>se especifica el # de fila donde debe comenzar la palabra</fila>'

+'<col> se especifica el # de fila donde debe comenzar la palabra</col>'

+'</palabra>'

Esta acción para cada una de las palabras que componen el crucigrama.

```
+ '</acrostico>');
```

Archivos XML de la Sopa de letras.

Al igual que en el crucigrama para no correr el riesgo de que un usuario pueda ver las respuestas o cambiarlas el siguiente archivo se crea dentro del script de flash.

```
('<?xml version="1.0" encoding="UNICODE" ?>'
```

```
+ '<dato>'
```

```
+ '<fil>se especifica el # de filas </fil>'
```

```
+ '<pal>'
```

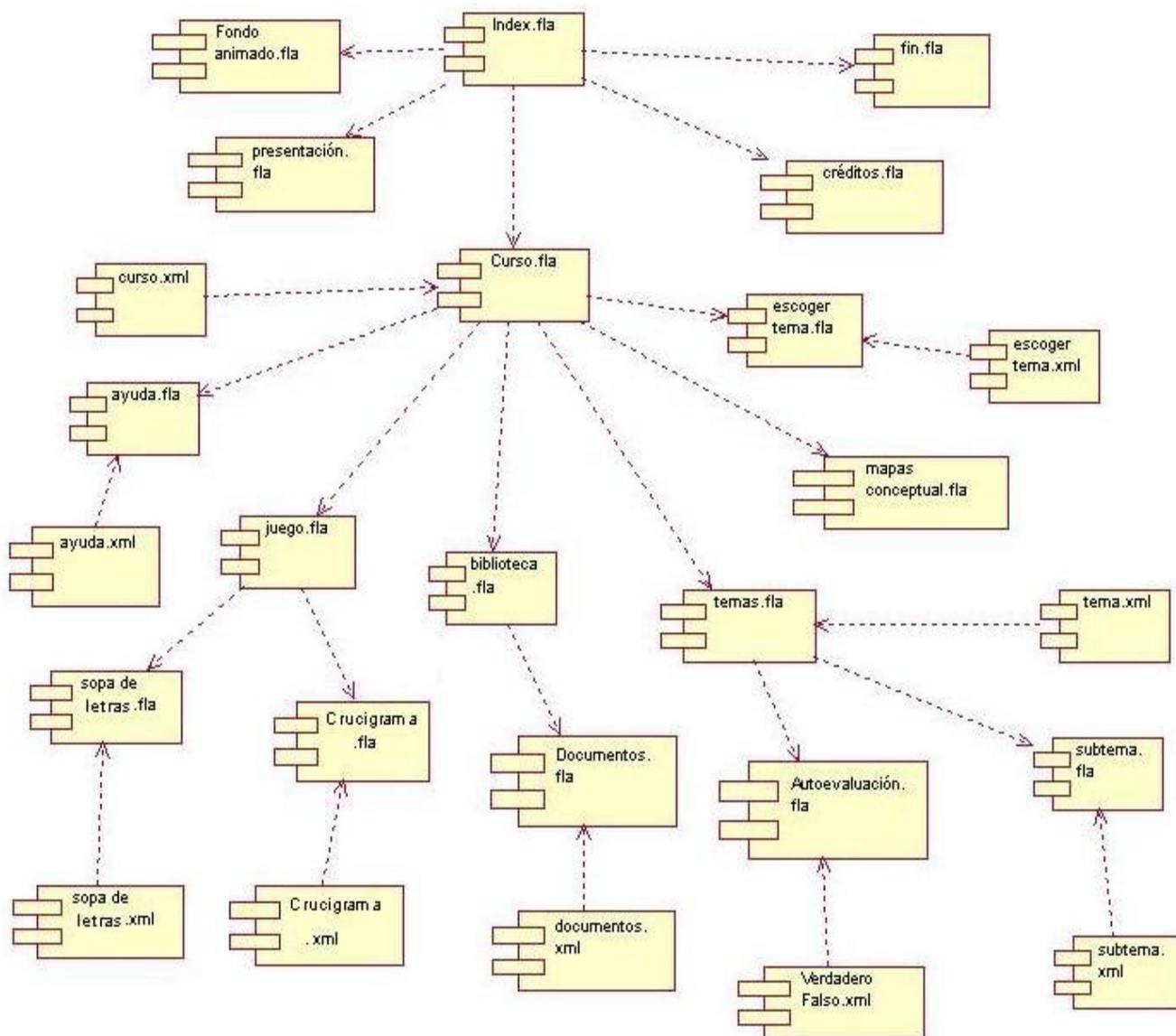
```
+ '<p c=" se escribe la palabra correspondiente" s="se escribe un # que indica el  
ángulo de la palabra" x="la posición en las x donde debe comenzar" y=" la posición en  
las y donde debe comenzar " />'
```

```
+ '</pal>'
```

```
+ '</dato>');
```

3.4 Diagrama de Componentes.

El diagrama de Componente se representa como un grafo de componentes software, unidos por medio de relaciones de dependencia, ya sean de compilación (los componentes de código binario) o de ejecución (y los componentes ejecutables), pudiendo mostrarse las interfaces que éstos soporten.



Conclusiones del Capítulo 3:

En este capítulo se obtuvieron los artefactos fundamentales correspondientes a los flujos de trabajo de diseño e implementación. Se realizó un análisis de los principios de diseño en los que se basó la aplicación. Se realiza la descripción de los archivos XML que se utilizan para el tratamiento de datos. Mediante el lenguaje de modelado para aplicaciones con tecnología multimedia propuestos por OMMM-L se elaboraron los diagramas de presentación, además del diagrama de componentes.

Conclusiones Generales

Después de realizar un amplio estudio de la documentación existente en el mundo sobre el proceso de Gestión de Riesgos y las herramientas para la creación de software multimedia, se concluye con la elaboración de la multimedia “Gestión de Riesgos”.

La aplicación cuenta con una interfaz amigable y un sistema de navegación placentero para el usuario, donde se integra la utilización del lenguaje XML para gestionar y agrupar los datos.

La Multimedia contiene el curso “Gestión de Riesgos”, el cual brinda con el objetivo de apoyar el proceso de enseñanza -aprendizaje, donde su contenido está distribuido por temas y subtemas, con ejemplos concretos de tablas y listas de riesgo, además cuenta con mapas conceptuales que agrupan los principales conceptos de la disciplina para facilitar a los usuarios la comprensión de las diferentes actividades del curso.

Se utilizó el Proceso Unificado del Software (RUP) con el lenguaje UML y la extensión del mismo para aplicaciones multimedia OMMMA-L, para la modelación de la aplicación, garantizando la obtención de un producto con calidad. Se les dio así cumplimiento a los objetivos propuestos y todas las tareas a desarrollar fueron cumplidas con éxito.

Recomendaciones

Se recomienda:

- Poner a disposición de los estudiantes la Multimedia Interactiva Gestión de Riesgos para la Calidad del Software e introducirla como material de apoyo al curso de Perfil Gestión de Riesgos impartido por la Dirección de Calidad de Software en la UCI.

- Integrar en un único producto las aplicaciones multimedia: Gestión de Riesgos, Pruebas de Software, Introducción a CMMI, Curso Ingeniería de Requisitos, etc. con el objetivo de que los estudiantes puedan acceder desde una misma aplicación a cualquiera de los cursos que el mismo desee consultar sin necesidad de tener que ejecutar diferentes aplicaciones.

Referencias

1. **Boehm, Barry W.** *A Spiral Model of Software Development and Enhancement*. 1988.
2. **Boehm, Barry W.** *Software Risk Management: Principles and practices*. 1, s.l. : Software IEEE, 1991, Vol. 8.
3. **Charette, Robert N.** *Software Engineering Risk Analysis and Management*. 1989.
4. **Fernández Carrasco, Oscar M.** *Investigador Agregado. Centro de Desarrollo Informático. SOFTCAL, SIME*. La Habana : s.n., 1995.
5. **ISO.** [translate.google.com.cu. translate.google.com.cu](http://translate.google.com.cu/translate?hl=es&sl=en&u=http://). [En línea] 1994. [Citado el: 16 de febrero de 2009.] <http://translate.google.com.cu/translate?hl=es&sl=en&u=http://>.
6. **Marcelo, Julian.** De la Gestión de los Riesgos en los Proyectos a la Gestión de los Proyectos por sus Riesgos. *QUATIC 2001*. [En línea] 13 de 01 de 2001. [Citado el: 12 de Marzo de 2009.] <http://www-ctp.di.fct.unl.pt/QUASAR/quatic2001/documents/JulianCocho.pdf>.
7. **Pressman, Roger S.** *Ingeniería de software. Un enfoque práctico. Cuarta Edición*. Madrid : McGraw_Hill Interamericana De España, 1998.
8. **Pressman, Roger S.** *Ingeniería del Software Un Enfoque Práctico 5ta edicion*. s.l. : MC Graw Hil, 2002.
9. **Ropponen, Janne y Lyytinen, Kalle.** *Components of Software Development Risk*. 2, NJ, EEUU : IEEE Press Piscataway, 2000, Vol. 26.
10. **Rubinstein, David.** [www.sdtimes.com. www.sdtimes.com](http://www.sdtimes.com). [En línea] 28 de noviembre de 2007. [Citado el: 30 de enero de 2009.] <http://www.sdtimes.com/content/artide.aspx?ArtideID=30247>.
11. **Salcedo Lagos, Pedro.** Univercidad De Consepcion. *Univercidad De Consepcion*. [En línea] 28 de agosto de 2002. [Citado el: 22 de febrero de 2009.] <http://www.inf.udec.d/%7Erevista/ediciones/edicion6/isetm.PDF>.
12. **vaughan, try.** *Todo el poder de la Multimedia. Segunda Edición*. Mexico : MC Graw, 1994.
13. **Veliz.** *Modelo para Gestión de Riesgos en proyectos de desarrollo de software en la Universidad de las Ciencias Informáticas*. La Habana : s.n., 2007.
14. **Booch, Jacobson.** *Lenguaje Unificado de Modelado UML*. s.l. : PEARSON .
15. **Adobe Systems Incorporated.** [Online] 2009. <http://www.adobe.com/>.