

# Universidad de las Ciencias Informáticas

Facultad # 5



**Título: Multimedia Interactiva de Métricas de Software**

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

**Autor: Yanisley Sánchez Piloto**

**Tutores: MSc. Yamilis Fernández Pérez**

**Ing. Miguel Ángel del Pino Zincke**

Ciudad de la Habana

Junio 2009

**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. Ocupó el cargo de Jefa del Departamento Docente Central de Ingeniería y Gestión de Software de la UCI desde su fundación. Actualmente directora de postgrado de la Universidad de las Ciencias Informáticas.

Tutor: Ing. Miguel Ángel del Pino Zincke

Correo electrónico: mdelpino@uci.cu

Graduado de Ingeniero en Ciencias Informáticas en 2008 en la Universidad de las Ciencias Informáticas. Profesor adjunto desde 2008 en la UCI. Ha desarrollado trabajos de investigación y desarrollo sobre pruebas de software, creando una aplicación multimedia que sirve de apoyo al proceso de enseñanza aprendizaje de pruebas de software en la UCI. Líder de proyecto en el centro de desarrollo de software UCI- MININT ubicado en la UCI.



*La verdadera ignorancia no es la ausencia de conocimientos, sino el hecho de rehusarse a adquirirlos.*

*Karl Popper*



### Agradecimientos

*Antes que todo quiero agradecer a la Revolución y a nuestro Comandante en Jefe Fidel.*

*A mi **Madre**: Por apoyándome, confiar en mí y haberme guiado siempre por el camino correcto (con muchísimos regaños), sin tu amor, cariño y comprensión este momento nunca hubiese sido posible. Te lo agradezco.*

*A mi **Hermano**: Por cuidarme, apoyarme, quererme, y estar siempre ahí cuando lo necesito. Además porque lo quiero mucho.*

*A mi **Abuela**: Por apoyarme cada vez que tenía un examen y darme fuerzas para confiar en mí misma. Por enseñarme que todo es posible si te lo propones. Y sobre todo por el amor que me has dado.*

*A mi **Tío Chino**: Por ser para mí como el padre que me faltó, por quererme como a una hija más, y sobre todo por el amor tan grande que me has dado a pesar de tener dos hijos.*

*A mi **Tía Arelis**: Porque gracias a ella tengo mi computadora, de verdad gracias te quiero.*

*A mis **Tías y Tíos** maternos: Por estar ahí cada vez que los necesito y por que supieron apoyarnos y darnos el cariño que la vida nos quito a mi mamá, a mi hermano y a mí. Además por ayudarme de una forma u otra cada vez que me daban algo para traer para la escuela.*

*A mis **Primas y Primos**: Por ser tan unidos para todo (incluyendo las fiestas que Aileen siempre inventa) y lograr una gran armonía y unión en nuestra pequeña familia.*

*A mi **Eddy**: Por ser mi novio y amigo, por entenderme siempre, por quererme tanto y aguantar mis caprichos y malcriadeces estos años, por estar conmigo en las buenas y en las malas en fin por todo lo que hemos pasado juntos, gracias.*

*A mi **Anabel**: Por estar a mi lado cada vez que lo necesité, por apoyarme y por ser como la hermana que no tuve.*

*A mis **Amistades**: Liset, Yahima, Yanay, Yadira, Carlos, Oliva y Yanexy en fin a todos por aguantarme estos 5 años.*

*A mis **Compañeros de la universidad**: Por apoyarme en las buenas y en las malas entre estos están Alexander, Johan, Anniel, Meylin, Amed, María Elena, Yany.*

*A mis **Compañeras de apartamento**: Por hacerme sentir como si estuviera en familia todo este tiempo.*

*A mis **Tutores**: Por ayudarme a desarrollar este trabajo corrigiéndome los errores, enseñándome cada día algo nuevo con paciencia y dedicación, en ocasiones dejando de hacer sus cosas para atenderme. De verdad muchas gracias.*

## **Dedicatoria**

*Les dedico este trabajo de diploma especialmente a mi papá y a mi abuelo (Pipo) que aunque físicamente no estén hoy aquí estarían muy orgullosos de este triunfo. También a mi mamá que se lo merece por haberse sacrificado tanto para que yo pudiera estudiar y que no me faltara nada. Y a mi hermano porque sé que está muy orgulloso de mí.*

### **Resumen**

Con la aparición de las Tecnologías de la Información y las Comunicaciones (TICs) se ha marcado un cambio significativo en el mundo. La Universidad de las Ciencias Informáticas (UCI) no se ha quedado atrás en este sentido. Desde un principio, se ha destacado en la creación de multimedia para el apoyo de los medios de enseñanza del país y de este centro en específico.

En este trabajo se realiza una multimedia para apoyar el proceso de enseñanza-aprendizaje de Métricas de Software en la UCI, apoyándose en la confección de mapas conceptuales que tengan los principales conceptos relacionados con esta disciplina. Para el desarrollo de la misma, se hizo un análisis detallado de las diferentes herramientas y metodologías existentes para la creación de aplicaciones multimedia, así como UML en su extensión OMMMA - L (lenguaje representativo), ActionScript (lenguaje de implementación) y XML (lenguaje para gestionar y agrupar los datos en volúmenes compactos de información).

### **Palabras claves**

Métricas, multimedia, multimedia interactiva, herramienta, metodología y lenguaje.

Índice

**Introducción ..... 1**

**Capítulo 1: “Fundamentación Teórica” ..... 7**

    Introducción ..... 7

    1.1 Fundamentos de métricas ..... 7

    1.2 Proceso de medición de software ..... 8

        1.2.1 Establecer y mantener compromiso de medición ..... 9

        1.2.2 Planificar el Proceso de Medición ..... 10

        1.2.3 Realizar el proceso de medición ..... 11

        1.2.4 Evaluar las Mediciones ..... 13

    1.3 Software educativo ..... 14

        1.3.1 Características de los programas educativos ..... 15

    1.4 Enseñanza de Métricas de Software en la actualidad ..... 15

    1.5 Multimedia ..... 16

        1.5.1 Clasificación de Multimedia ..... 17

    1.6 Tendencias tecnológicas actuales ..... 19

        1.6.1 Herramienta a utilizar para la creación del Software Multimedia ..... 19

        1.6.2 Metodología a utilizar para el desarrollo del software ..... 21

        1.6.3 Lenguajes de Modelado ..... 23

        1.6.4 Lenguajes utilizados en la multimedia ..... 24

    Conclusiones parciales ..... 25

**Capítulo 2: Descripción de la solución ..... 26**

    Introducción ..... 26

2.1 Especificación de la audiencia .....	26
2.2 Especificación del Contenido.....	26
2.2.1 Mapa Conceptual.....	27
2.3 Descripción del Modelo de Dominio.....	28
2.3.1 Diagrama de clases del dominio .....	29
2.3.2 Análisis de los conceptos del Dominio.....	29
2.4 Descripción del sistema propuesto .....	30
2.4.1 Requisitos Funcionales .....	30
2.4.2 Requisitos no funcionales .....	32
2.5 Diagrama de navegación .....	34
2.6 Modelo de Casos de uso del sistema.....	35
2.6.1 Determinación y justificación de los Actores del Sistema .....	35
2.6.2 Diagrama de Casos de Uso.....	36
2.6.3 Breve descripción de los casos de uso del sistema .....	36
Conclusiones Parciales .....	42
<b>Capítulo 3: Solución de la descripción propuesta.....</b>	<b>43</b>
Introducción .....	43
3.1 Análisis de la Arquitectura utilizada .....	43
3.1.1 Principios de diseño .....	43
3.1.2 Estándares para el diseño de la interfaz de usuario .....	44
3.2 Diagrama de Presentación del modelo de análisis.....	45
3.3 Descripción de Archivos XML.....	48
3.4 Modelo de Implementación.....	51

3.4.1 Diagrama de Componentes ..... 51

Conclusiones Parciales ..... 52

**Conclusiones Generales .....53**

**Recomendaciones .....54**

**Referencias Bibliográficas .....55**

**Bibliografía .....57**

**Anexos.....58**

**Glosario de Términos .....64**

**Índice de Figuras**

Fig. 1: Muestra los proyectos que usan métricas según su área (Fuente: Diagnóstico 2008).....3

Fig. 2: Muestra el uso de métricas en los proyectos a nivel UCI (Fuente: Diagnóstico 2008).....3

Fig. 3: Estructura de la información seleccionada de los libro el SWEBOK y SEEK.....8

Fig. 4: Mapa Conceptual.....28

Fig. 5: Diagrama de dominio .....29

Fig. 6: Diagrama de navegación .....34

Fig. 7: Diagrama de CU .....36

Fig. 8: Diagrama de Presentación Pantalla Principal.....45

Fig. 9: Diagrama de Presentación Pantalla Tema .....46

Fig. 10: Diagrama de Presentación Pantalla Ejercicio .....46

Fig. 11: Diagrama de Presentación Pantalla Juego.....47

Fig. 12: Diagrama de Presentación Pantalla Salir .....47

Fig. 13: Diagrama de componente general por paquetes.....52

Anexo 1: Paquete Juego.....58

Anexo 2: Paquete Mapa.....58

Anexo 3: Paquete Presentación.....58

Anexo 4: Paquete Tema .....59

Anexo 5: Paquete Video .....59

Anexo 6: Paquete Glosario .....60

Anexo 7: Pantalla ejercicios .....62

Anexo 8: Pantalla mapa conceptual con ventana flotante .....62

Anexo 9: Pantalla juego .....63

Anexo 10 : Pantalla Video.....63

**Índice de Tablas**

Tabla 1: Comparación entre las metodologías estudiadas .....22

Tabla 2: Requisitos funcionales .....30

Tabla 3: Sistemas Operativos y navegadores necesarios para ejecutar la aplicación .....32

Tabla 4: Actor y justificación .....35

Tabla 5: CU Mostrar Presentación .....37

Tabla 6: CU Mostrar Ayuda.....37

Tabla 7: CU Visualizar Glosario de Términos.....38

Tabla 8: CU Mostrar Información .....38

Tabla 9: CU Visualizar Mapa Conceptual.....39

Tabla 10: CU Ejecutar Juego .....39

Tabla 11: CU Visualizar Video .....40

Tabla 12: CU Salir de la Aplicación.....40

Tabla 13: CU Realizar Ejercicios.....41

Tabla 14: CU Mostrar Materiales Complementarios .....41

Tabla 15: CU Reproducir Música .....42

### Introducción

La industria del software, a diferencia de otras, tiene muy poco tiempo de existencia. Lo que ha llamado la atención del mercado hacia ella han sido esencialmente dos factores: la velocidad con que ha crecido y su alcance. En la actualidad, ésta ocupa uno de los lugares más atractivos del mercado y un espacio de creciente demanda. Cubrir dicha demanda se hace difícil debido a que se incrementa la competencia, por lo que se imponen productos cada vez con mayor calidad. La creciente complejidad de los productos de software y la necesidad de desarrollarlos en periodos de tiempo cada vez más cortos, está impulsando a la comunidad de Ingeniería del Software a diseñar y adoptar nuevas tecnologías. Cada vez es mayor la necesidad de justificar de una manera razonada tanto los diseños como la propia programación.

En el mundo actual, el desarrollo de software está presentando muchos problemas. Desde el año 1994, Standish Group<sup>1</sup> realiza investigaciones que son publicadas bianualmente, donde se muestran las cifras de los proyectos rechazados, aceptados y cancelados.

Según una publicación realizada por Córdoba software factory sobre el estudio realizado por la compañía norteamericana denominada Standish Group, conocido como "Chaos Report", en su última publicación, en el 2006: Según el estudio del total de proyectos evaluados: el 16% son completados con el alcance esperado, en el tiempo planificado y dentro del presupuesto asignado, el 53% son completados con menor alcance, y/o sobrecosto y/o fuera de término, y el 31% son cancelados antes de terminar. Del total de proyectos que se completan: el 70% terminan fuera de plazo, el 54% sufren sobrecostos, el 66% no son considerados exitosos y el 30% son cancelados antes de terminar. (*Javo 2008*)

En comparación con las estadísticas de 1994 se ha mejorado, pero aún hay muchísimos problemas en el desarrollo de software. A nivel mundial, de los proyectos que se inician son muy pocos los que se terminan.

---

<sup>1</sup> Standish Group: Compañía norteamericana, conformada por un grupo de profesionales con años de experiencia y práctica.

En la industria del software en Cuba existen problemas con la productividad de los trabajadores, los tiempos de entrega de los productos y las documentaciones. Las causas fundamentales de estos problemas están dadas porque no hay disciplina en el cumplimiento de los calendarios del proceso de desarrollo; no se realizan seguimientos de la evolución de los procesos, ni de la productividad de los desarrolladores; no se estimula a crear un espíritu de equipo; el software excede el tiempo y el costo planificado, existen problemas en la administración de los proyectos, la cual debe contar con planes claros, precisos y que sirvan para cuantificar y medir el estado en cada momento del desarrollo del proyecto. (PÉREZ, 2002)

Teniendo en cuenta que la Universidad de las Ciencias Informáticas (UCI) tiene como misiones: producir software para la industria nacional e incluir al país en el mercado internacional, se hace necesario crear las estrategias que garanticen un adecuado desarrollo de software con el que sea posible limar las dificultades existentes y así lograr productos de alta calidad. La producción de software en la UCI se ve afectada, de varias maneras en la actualidad, debido a la insuficiente calidad de los procesos empleados para la elaboración, desarrollo y mantenimiento de sus productos.

Con el objetivo de mejorar la producción de software, se diseñan y aplican métricas que nos ayudan a entender tanto el proceso técnico que se utiliza para desarrollar un producto, como el propio producto. Una métrica es “La aplicación continua de mediciones basadas en técnicas para el proceso de desarrollo del software y sus productos para suministrar información relevante a tiempo. Así el planificador, junto con el empleo de estas técnicas, mejorará el proceso y sus productos (Brito, 2007).

Según un diagnóstico realizado en el 2008 a los proyectos productivos de la UCI, conducido por la Dirección de Calidad de Software, se determinó lo siguiente:

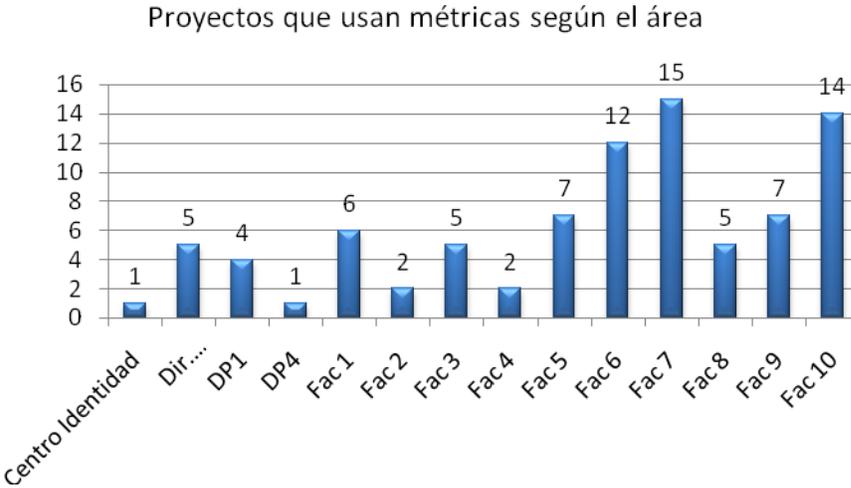


Fig. 1: Muestra los proyectos que usan métricas según su área (Fuente: Diagnóstico 2008)

Proyectos según el uso de métricas a nivel UCI



Fig. 2: Muestra el uso de métricas en los proyectos a nivel UCI (Fuente: Diagnóstico 2008)

En la UCI existe un 44% de los proyectos productivos que aún no usa métricas. A partir de las entrevistas realizadas a líderes de proyecto, jefes de polos, vicedecanos de producción y asesores de calidad, por varios consultores del SIE Center de Monterrey<sup>2</sup>, se detectaron algunas problemáticas en los proyectos

<sup>2</sup> SIE Center de Monterrey: Instituto Tecnológico y de Estudios Superiores de Monterrey, México.

productivos y se detallaron por área de proceso. En el área de medición y análisis se detectaron las siguientes:

En la meta específica Alinear actividades de medición y análisis, no se especifican mediciones, ni procedimientos de recolección y almacenamiento de datos, ni tampoco procedimientos de análisis y los objetivos de medición se establecen solo en algunos proyectos.

En la meta específica Proveer resultados de Medición, solo algunos proyectos coleccionan datos de medición, analizan los mismo, almacenan datos y resultados y se comunican los resultados.

En la meta genérica Institucionalizar un Proceso Gestionado, no se establece una Política Organizacional, no se entrena al personal, es decir, no se le da la capacitación necesaria y no se revisa el estado con la alta gerencia.

Ante esta realidad y debido al carácter urgente de realizar un proceso de desarrollo del software con la calidad requerida, se hace necesario que el personal dedicado a este proceso esté suficientemente capacitado en el tema de Métricas de software, el cual es un eslabón fundamental en el proceso de desarrollo del mismo. El tema métricas de software no está definido como un curso de pregrado, es decir, los estudiantes pasan 5 años en la universidad, se hacen ingenieros y no saben que es métricas de software. Además, la UCI está actualmente en un programa de mejora para llegar al nivel 2 de madurez de CMMI<sup>3</sup>, para lo cual es necesario personal suficientemente capacitado. Por esto surge la necesidad de brindarles la capacitación necesaria a los estudiantes que participan en el proceso de desarrollo de software, en la definición de métricas de software en la UCI.

Por lo anteriormente planteado, tenemos como **Problema científico**: ¿Cómo sustentar la inclusión de buenas prácticas en el proceso de enseñanza-aprendizaje de métricas de software, a través de los medios interactivos?

**Objeto de Estudio**: Los medios interactivos en el proceso de enseñanza-aprendizaje de métricas de software. El mismo tiene como **Campo de Acción**: El proceso de desarrollo de una multimedia interactiva que apoye la enseñanza- aprendizaje de métricas de software en la UCI. El objetivo entonces es:

---

<sup>3</sup> CMMI: Capability Maturity Model Integration (Modelo de Capacidad y Madurez Integrado). Modelo que constituye un marco de referencia de la capacidad de desarrollo de software de diferentes empresas.

Implementar una multimedia interactiva como medio de enseñanza-aprendizaje de métricas de software en la UCI.

Para cumplir el objetivo propuesto, se realizaron las siguientes **tareas**:

- ✓ Recopilar y analizar toda la información necesaria sobre métricas de software, para la elaboración del contenido del producto a desarrollar.
- ✓ Confeccionar un mapa conceptual que contenga los conceptos necesarios para la comprensión del tema de métricas de software.
- ✓ Analizar de forma crítica los diferentes lenguajes como XML (Lenguaje de Marcas Extensibles), ActionScript (Lenguaje de programación), y los lenguajes de modelado UML (Lenguaje Unificado de Modelado) y su extensión OMMMA-L (Lenguaje de Modelado Orientado a Objetos para Aplicaciones Multimedia).
- ✓ Seleccionar la metodología de desarrollo de software que mejor se ajuste a las características del producto a desarrollar.
- ✓ Analizar y diseñar una multimedia que contenga un mapa conceptual y diversos medios que ayuden a la comprensión del tema, utilizando la herramienta y el lenguaje de modelado seleccionado.

Entre los **Métodos Científicos** utilizados se destacan:

### **Métodos teóricos:**

Análítico-sintético: Para procesar la información de métricas de software recopilada durante la investigación y lograr una mejor comprensión del fenómeno.

Histórico-lógico: Permite realizar una investigación que comience con los orígenes de la métricas en el mundo y las tendencias que actualmente existen.

### **Método empírico**

Entrevista: Se realizarán entrevistas a profesores y especialistas en la rama que poseen conocimientos acerca del tema de métricas del software en la UCI.

El presente trabajo está estructurado de la siguiente manera:

**Capítulo 1:** “Fundamentación teórica.”

Éste contiene todos los conceptos relacionados con el objeto de estudio para lograr una mejor comprensión del problema. Se reflejan las tendencias y tecnologías actuales consideradas para la elaboración del producto. Se describen los lenguajes de programación y sistemas para realizar el análisis y diseño de la aplicación.

### **Capítulo 2:** “Descripción de la solución propuesta”

En este capítulo se realiza una breve especificación del contenido de la multimedia y el modelado del negocio del sistema mediante el modelo de dominio. Se hace el levantamiento de los requisitos funcionales y no funcionales, así como la descripción de los casos de uso correspondiente para el modelado del sistema. Se representan los mapas de navegación.

### **Capítulo 3:** “Solución de la descripción propuesta.”

Éste capítulo está dedicado principalmente a la construcción del sistema a desarrollar, incluyendo los diagramas de presentación y el diagrama de componentes.

# Capítulo 1: “Fundamentación Teórica”

## Introducción

En este capítulo se abordan los principales conceptos de métricas y los aspectos más importantes seleccionados en dos libros, el SWEBOK (Guía de la Ingeniería de Software. Cuerpo del Conocimiento) (IEEE, 2004) y el SEEK (Curricula de la carrera de Ingeniería de Software) (IEEE, ACM, 2004) para elaborar el contenido de la multimedia. Además, se tratan los aspectos relacionados con el software educativo. Se describen los principales conceptos de Multimedia y Multimedia Interactiva. Se exponen aspectos sobre las tecnologías y tendencias actuales, haciendo énfasis en la metodología, lenguajes y herramientas a utilizar para el desarrollo de la aplicación.

### 1.1 Fundamentos de métricas

Para entender mejor el concepto de métrica, es necesario aclarar que los términos métricas, medición y medida no tienen el mismo significado.

**Medida:** Proporciona una indicación cuantitativa de la extensión, cantidad, dimensiones, capacidad o tamaño de algunos atributos de un proceso o producto. (Muñoz M. C., 2007)

**Medición:** La medición es el acto de determinar una medida. (Muñoz M. C., 2007)

**Métrica:** El concepto de métrica es el término que describe muchos y muy variados casos de medición. Siendo una métrica una medida estadística, ésta es un buen medio para entender, controlar, predecir, probar y valorar la calidad de productos, procesos y software (Álvarez, 2007).

En general, la medición persigue tres objetivos fundamentales:

- ✓ Entender qué ocurre durante el desarrollo y el mantenimiento del software.
- ✓ Controlar qué es lo que ocurre en proyectos.
- ✓ Mejorar procesos y productos.

Tras un estudio realizado a partir de los libros mencionados anteriormente, se determinó parte del contenido a presentar en la multimedia. El libro SWEBOK está dividido en 11 áreas de procesos, de éstas se hace énfasis en dos que son: Gestión de la Ingeniería de Software y Proceso de Ingeniería de

Software. Dentro de cada una de estas áreas se seleccionó una subárea que es la que interesa en realidad: Medición de Ingeniería de Software y Medición de Producto y Proceso respectivamente.

Luego del análisis de los puntos esenciales que tienen en común estos dos libros, se determinó que la medición del producto y proceso puede incluirse en medición de ingeniería de software, por lo tanto la información queda estructurada de esta forma:

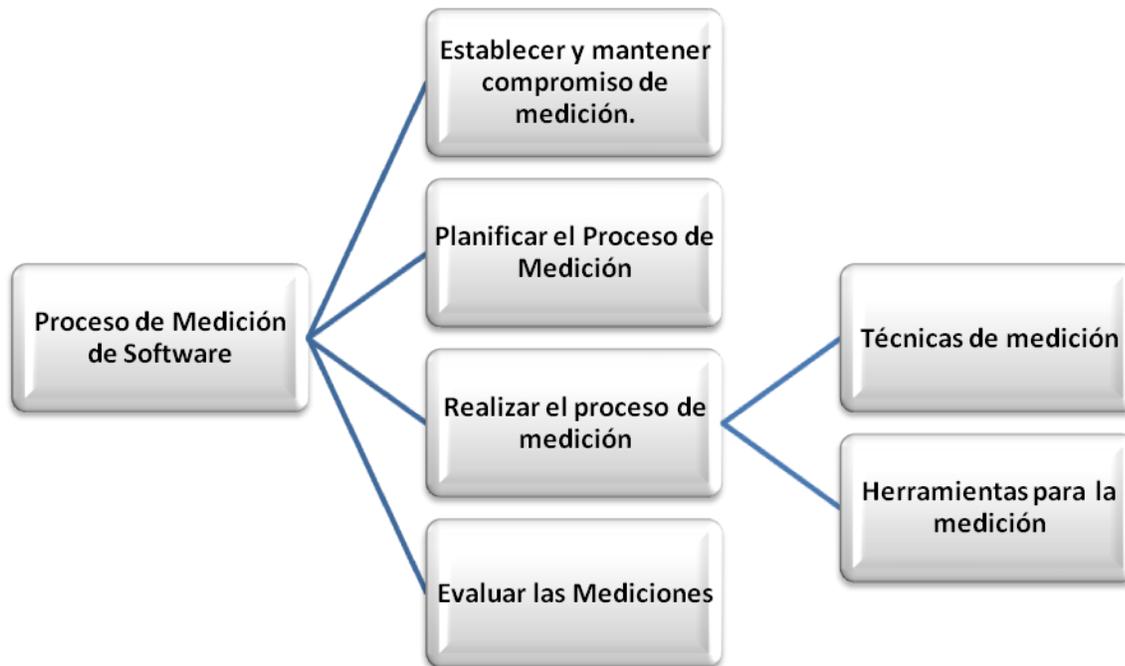


Fig. 3: Estructura de la información seleccionada de los libros el SWEBOK y SEEK

### 1.2 Proceso de medición de software

El proceso de medición de software tiene importancia no sólo para las grandes organizaciones, éste también puede ser realizado con éxito por pequeñas organizaciones, equipos e individuos. La importancia de la medición y su papel en mejores prácticas es ampliamente reconocida, por lo que sólo puede aumentar en los próximos años.

Se utiliza la medición para identificar las fuerzas y las debilidades de los procesos, productos y proyectos, y para evaluarlos después de que hayan sido implementados y/o cambiados.

Para realizar un buen proceso de medición, es necesario llevar a cabo una serie de actividades como las que se mencionan a continuación:

- ✓ Establecer y mantener compromiso de medición.
- ✓ Planificar el proceso de medición.
- ✓ Realizar el proceso de medición.
- ✓ Evaluar medición.

Cada una de estas actividades se explica en los epígrafes que siguen a continuación.

### 1.2.1 Establecer y mantener compromiso de medición.

Para llevar a cabo un proceso de medición como es debido, lo primero que hay que hacer es establecer y mantener un compromiso de medición, y para esto es necesario realizar una serie de actividades como se detallan a continuación:

- ✓ Aceptar los requisitos de medición: Cada medición o esfuerzo debe guiarse por los objetivos de la organización, impulsada por un conjunto de requisitos de medición establecidos por la organización y el proyecto.
- ✓ Definir el alcance de la medición. Se debe establecer la unidad organizacional a la que se va a aplicar cada requisito de medición. Esto puede consistir en un área funcional, un solo proyecto, un sitio único, o incluso toda la empresa. Todas las subsiguientes tareas relacionadas con este requisito deben estar dentro del alcance definido.
- ✓ Compromiso de la dirección y el personal a medición. El compromiso debe ser oficialmente establecido, comunicado, y con el apoyo de recursos.
- ✓ Comprometer recursos para la medición. El compromiso de la organización con la medición es un factor esencial para el éxito, como lo demuestra la asignación de recursos para la aplicación del proceso de medición. El asignar recursos incluye el reparto de responsabilidades para las diferentes tareas del proceso de medición y el proporcionar una financiación, entrenamiento, herramientas, y apoyo adecuados para dirigir el proceso de un modo perdurable.

### 1.2.2 Planificar el Proceso de Medición

Para planificar el proceso de medición es necesario realizar una serie de pasos como son:

- ✓ Caracterizar la unidad organizativa. La unidad organizativa proporciona el marco para la medición, por lo que es importante hacer explícito este contexto y articular las presunciones que éste incluye y las restricciones que va imponiendo. La caracterización puede ser en términos de los procesos organizacionales, ámbitos de aplicación, tecnología de organización y las interfaces.
- ✓ Identificar las necesidades de información. Las necesidades de información se basan en los objetivos, las limitaciones, riesgos y problemas de la unidad organizacional. Deben ser identificadas y priorizadas. Luego, un subconjunto a tratar debe ser seleccionado y los resultados documentados, comunicados, y revisados por el cliente.
- ✓ Seleccionar las mediciones. Se deben elegir las mediciones candidatas al puesto, claramente vinculadas a las necesidades de información. Las mediciones deben ser seleccionadas sobre la base de las prioridades de las necesidades de información y otros criterios como el costo de recogida, el grado de perturbación durante el proceso de recogida, la facilidad de análisis, la facilidad de obtención de información precisa de datos, entre otras.
- ✓ Definir la recopilación de datos, análisis y los procedimientos para informar. Esto incluye los procedimientos de recolección y los cuadros, el almacenamiento, verificación, análisis, informes, configuración y gestión de datos.
- ✓ Revisar, aprobar y proporcionar recursos para tareas de medición:  
El plan de medición debe ser revisado y aprobado por los contratistas. Esto incluye todos los procedimientos de recopilación de datos, almacenamiento, análisis y presentación de informes, procedimientos de evaluación, criterios de horarios y responsabilidades. Los criterios para la revisión de estos artefactos se deben establecer a nivel de unidad organizacional o superior y deben ser utilizados como base para estas mediciones.
- ✓ Adquirir y desplegar tecnologías de apoyo. Esto incluye la evaluación de las tecnologías disponibles, la selección de la tecnología más adecuada, la adquisición de esas tecnologías, y despliegue de las mismas.

### 1.2.3 Realizar el proceso de medición

Para realizar un buen proceso de medición hay que tener en cuenta algunas actividades, técnicas y herramientas de medición. A continuación se muestran estas actividades y se explican las técnicas utilizadas para apoyar las mismas, además se citan dos de las herramientas automatizadas que pueden ser utilizadas para realizar dicho proceso.

- ✓ Integrar los procedimientos de medición con los procesos pertinentes. Los procedimientos de medición tales como la recopilación de datos, deben integrarse en los procesos que se están midiendo. Esto puede implicar el cambio de curso en los procesos para dar cabida a la recopilación de datos o la generación de actividades. También puede implicar el análisis de los actuales procesos para reducir al mínimo esfuerzo adicional y la evaluación del efecto en los empleados para asegurarse de que el procedimiento de medición será aceptado.
- ✓ Recopilar datos. Los datos deben ser recogidos, verificados y almacenados.
- ✓ Analizar los datos y desarrollar productos de información. Los datos pueden ser recopilados, transformados, o decodificados. Los resultados de este análisis son típicamente indicadores tales como gráficos, números, u otras indicaciones que deben ser interpretados, dando lugar a conclusiones iniciales que han de presentar los contratistas.
- ✓ Comunicar los resultados. La información de los productos debe ser documentada y comunicada a los usuarios y contratistas.

El proceso de medición se realiza a través de técnicas de medición y herramientas.

#### 1.2.3.1 Técnicas de medición

Las técnicas de medición pueden ser utilizadas para analizar el proceso de ingeniería de software y para identificar las fortalezas y debilidades. Las técnicas de medición se han clasificado en dos tipos generales: análisis y evaluación comparativa. Los dos tipos de técnicas pueden usarse conjuntamente, ya que están basadas en distintos tipos de información.

### ✓ Técnicas de análisis

Las técnicas analíticas se caracterizan por basarse en pruebas cuantitativas. El tipo de análisis es ejemplificado por el mejoramiento de la calidad del paradigma (proyectos de efecto rápido) que consiste en un ciclo de comprensión, evaluación. A continuación se presentan algunas técnicas de análisis.

- Estudios Experimentales: La experimentación consiste en la creación de experimentos controlados en la organización para evaluar los procesos. Por lo general, un nuevo proceso se compara con el actual proceso para determinar si el primero tiene un mejor proceso de resultados.
- Definición de Proceso de Revisión: Es una manera por la cual una definición de proceso (ya sea uno descriptivo o uno prescriptivo, o ambos) es revisada, y las deficiencias y las mejoras del proceso de potencial identificadas.
- Clasificación de defecto Ortogonal: es una técnica que se puede utilizar para vincular las fallas encontradas con las posibles causas. Se basa en una cartografía de los tipos entre la culpa y la culpa desencadenantes.
- Análisis de Causa Raíz: es otra técnica analítica común que se utiliza en la práctica. Esto implica seguimiento posterior de los problemas detectados (defectos) para identificar el proceso de causas, con el objetivo de cambiar el proceso para evitar estos problemas en el futuro.
- Control Estadístico de Procesos: es una forma eficaz de determinar la estabilidad, o la falta de ella en el proceso, mediante el uso de gráficos de control y sus interpretaciones.

### ✓ Técnicas de evaluación comparativa

El segundo tipo de técnica, "depende de la identificación de una excelente organización en un campo y documentar sus prácticas y herramientas". Bench marking supone que si una organización menos competente adopta la prácticas de una excelente organización, también es excelente. La evaluación comparativa se ocupa de la evaluación de la madurez de una organización o la capacidad de sus procesos.

#### 1.2.3.2 Herramientas automatizadas para la medición del proceso

Entre las herramientas automatizadas para realizar la medición del proceso de software se encuentran:

**Proceso de Dashboard:** Herramienta de apoyo al PSP y al TSP. Fue desarrollada originalmente en 1998 por la Fuerza Aérea de Estados Unidos y ha seguido evolucionando en el marco del modelo de código abierto.

El proceso de Dashboard apoya:

- ✓ **Recogida de datos** - Tiempo, defectos, tamaño del plan en comparación con los datos reales.
- ✓ **Planificación** - Integrado de scripts, plantillas, formularios, y los resúmenes, valor ganado.
- ✓ **Análisis de Datos** - Gráficos e informes de la ayuda en el análisis de los datos históricos de las tendencias.
- ✓ **Exportación de datos** - Los datos de exportación a Excel, o exportar datos a formato de texto para su uso con herramientas externas.

**PSM Insight:** Es una aplicación basada en Windows que permite un alto grado de flexibilidad en la gestión de datos, incluyendo la modificación, la navegación, y sofisticadas capacidades gráficas. Proporciona la capacidad para adaptar el software de medición a un proyecto único y permite el uso de los datos ya disponibles a partir de un proceso de desarrollo de software existente. Desarrollada para apoyar el PSM<sup>4</sup>

Está patrocinada por el Departamento de Defensa y el Ejército de los EE.UU, específicamente en la Oficina de Métricas de Software.

### 1.2.4 Evaluar las Mediciones

Para evaluar una medición es importante regirse por algunas actividades esenciales.

- ✓ Evaluar productos de información. Evaluar la información de productos especificados en contra de los criterios de evaluación y determinar los puntos fuertes y débiles de la información de productos. Esto puede ser realizado por un proceso interno o una auditoría externa y debe incluir información

---

<sup>4</sup> PSM: Practical Software Measurement.

de medición de los usuarios. Las lecciones aprendidas deben guardarse en una base de datos adecuada.

- ✓ Identificar posibles mejoras. Estas mejoras pueden ser los cambios en el formato de los indicadores, los cambios en unidades de medida, o la reclasificación de las categorías. Se deben determinar los costos y beneficios de las mejoras potenciales y seleccionar las acciones de mejora adecuadas. Además debe ser informada la propuesta a los contratistas y dueños del proceso de medición, para su revisión y aprobación.

La medición es muy común en el mundo de la ingeniería. Se mide potencia de consumo, pesos, dimensiones físicas, temperaturas, voltajes, señales de ruidos por mencionar algunos aspectos. Pero desgraciadamente como se ha podido ver, la medición se aleja de lo común en el mundo de la ingeniería del software. La medición de software es una disciplina relativamente joven y se encuentran muchas dificultades en ponerse de acuerdo sobre qué medir y cómo evaluar las medidas. Para esto, como se explicó anteriormente se debe regir por los fundamentos de los libros antes mencionados.

### 1.3 Software educativo

A través de los años el hombre ha presentado un cambio radical en su nivel de vida. Los conocimientos que él ha logrado acumular y aplicar han sido para su beneficio, lo cual ha cambiado radicalmente su modo de vivir. Con el desarrollo de las nuevas tecnologías, la educación no se ha quedado atrás. Ésta también tiene su avance, ya que tiene como objetivo adecuar las estrategias de enseñanza – aprendizaje a los nuevos tiempos.

Durante la década de los años 1980, la "explosión" tecnológica cambió el mundo de la educación y los aportes de las ciencias y la industria fueron llevados a las clases. Surgieron entonces materiales como las diapositivas, películas y otros medios para sumarse a los que existían anteriormente. Más adelante apareció el concepto de "software educativo", que no es más que cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. Un concepto más restringido de Software Educativo lo define como aquel material de aprendizaje especialmente diseñado para ser utilizado con un computador en los procesos de enseñar y aprender. (Dellamea, 2006).

### 1.3.1 Características de los programas educativos

Los programas educativos pueden tratar las diferentes materias (matemáticas, idiomas, geografía, dibujo...), de formas muy diversas (a partir de cuestionarios, facilitando una información estructurada a los alumnos, mediante la simulación de fenómenos...) y ofrecer un entorno de trabajo más o menos sensible a las circunstancias de los alumnos y más o menos rico en posibilidades de interacción, pero todos comparten cinco **características esenciales**: (Marquèz, 2004)

- ✓ Son materiales elaborados con una **finalidad didáctica**, como se desprende de la definición.
- ✓ **Utilizan el ordenador** como soporte en el que los alumnos realizan las actividades que ellos proponen.
- ✓ **Son interactivos**, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- ✓ **Individualizan el trabajo** de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los mismos.
- ✓ **Son fáciles de usar**. Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, es decir, son mínimos, aunque cada programa tiene reglas de funcionamiento que son de necesario conocimiento.

### 1.4 Enseñanza de Métricas de Software en la actualidad

En la carrera de informática es una necesidad la enseñanza de métricas de software para poder llevar a cabo la medición del software como es debido.

En algunas universidades del mundo tales como la Universidad de Carlos III de Madrid, Universidad de la Sabana de Colombia, la Universidad de Concepción de Canadá, la universidad de Sevilla de España, las métricas se encuentran en el plan de estudios de la carrera de Informática como asignaturas opcionales o incluidas en alguna asignatura, y no como un curso obligatorio que es como debía ser, debido a que algunos de los problemas que se presentan en los proyectos, están dados por la falta de capacitación del personal vinculado al proyecto en cuanto a análisis y medición.

Sin embargo existe un curso de verano en España donde se incluye métricas de software como asignatura obligatoria dentro del plan de estudio de la carrera, pero éste hay que pagarlo, lo cual nos imposibilita acceder al mismo. También existen centros como la Escuela Nacional de Posgrados, San Marcos (Universidad de Perú), donde para la maestría de Ingeniería de Sistemas e informática es necesario cursar varias asignaturas y entre los cursos especializados de la maestría está Métricas de software; y también está el Centro Universitario de Ciencias Exactas e Ingenierías de Guadalajara (México) el cual imparte las materias de calidad de la informática y métricas de calidad de software, las cuales son obligatorias.

La Universidad de las Ciencias Informáticas, dentro de su plan de estudio, tiene la asignatura Ingeniería de Software, la cual tiene incluida algo de métricas de software en la parte de estimación. No cuenta con un curso de pregrado de métricas de software.

En estas universidades se imparten asignaturas las cuales tienen en sus contenidos métricas de software, pero en ninguna de éstas se utiliza la multimedia como medio de enseñanza.

### 1.5 Multimedia

Cuando se habla de Multimedia, se refiere a la integración de múltiples elementos de comunicación en un mismo programa o aplicación, tales como: Texto, sonido, imagen, vídeo, animación, etc. Otra característica sería la posibilidad de avanzar de forma no lineal por la aplicación. A diferencia de un libro en el que tienes que comenzar por la página uno y acabar en la x, en un programa con tecnología multimedia tienes la posibilidad de avanzar a saltos, pues todas sus partes están interconectadas.

Es importante aclarar que existen tres conceptos que, aunque estén muy relacionados, son diferentes y es necesario conocerlos para un mejor entendimiento de lo que es cada uno de ellos. Estos son: hipertexto, multimedia e hipermedia, éste último es una combinación de los dos anteriores (**hipertexto + multimedia = hipermedia**).

**Multimedia** es una tecnología digital que, a través de la computadora, integra diversos tipos de datos como gráficas, sonido, imagen fija e imagen en movimiento. Consiste en la integración de varias tecnologías de comunicación a través de la computación, dando lugar a datos digitales, sonidos digitales, imagen digital, etc. (Diaz, 1994)

El **hipertexto** es una tecnología que organiza una base de información en bloques distintos de contenidos conectados a través de una serie de enlaces, cuya activación o selección provoca la recuperación de información. (Adelaide Bianchini, 2007)

El hipertexto ha sido definido como un enfoque para manejar y organizar información, en el cual los datos se almacenan en una red de nodos conectados por enlaces. (Adelaide Bianchini, 2007)

La hipermedia surge como resultado de la fusión de dos tecnologías, el hipertexto y la multimedia, por tanto, es la tecnología que nos permite estructurar la información de una manera no-secuencial, a través de nodos interconectados por enlaces. La información presentada en estos nodos podrá integrar diferentes medios. (Texto, sonido, gráficos...). (Yusef Hassan, 2002)

### 1.5.1 Clasificación de Multimedia

Con el desarrollo y evolución de las tecnologías se ha desarrollado ampliamente un conjunto de aplicaciones denominadas **aplicaciones multimedia interactivas**, las cuales nos permiten interactuar con el ordenador utilizando diferentes códigos en la presentación de la información (texto, imagen, sonido,... ). Estas aplicaciones son las más utilizadas en la educación y en los procesos de intervención en logopedia.

El rasgo más significativo y novedoso de una obra multimedia es la interactividad como modo esencial de realizarse el proceso de comunicación. El modo de propiciar esta interactividad es instrumentando sistemas de búsqueda y de navegación. Entre estos últimos están los índices o menús en variantes tales como listas, botones e iconos. También muy difundida es la técnica de hipertexto y con mayor complejidad la de los hipermedio. (Pérez, 2007)

Para tener una visión general de los diferentes tipos de programas, podemos clasificarlos en función de diferentes criterios como: Sistema de navegación y Nivel de control que tiene el profesional.

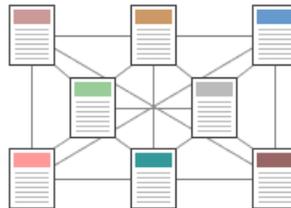
✓ **Según su sistema de navegación** (Ortí, 2006):

Los sistemas de navegación más usuales en relación a la estructura de las aplicaciones son:

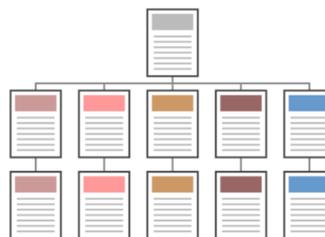
- **Lineal.** El usuario sigue un sistema de navegación lineal o secuencial para acceder a los diferentes módulos de la aplicación, de tal modo que únicamente puede seguir un determinado camino o recorrido. Esta estructura es utilizada en gran parte de las aplicaciones multimedia de ejercitación y práctica o en libros multimedia.



- **Reticular.** Se utiliza el hipertexto para permitir que el usuario tenga total libertad para seguir diferentes caminos cuando navega por el programa, atendiendo a sus necesidades, deseos, conocimientos, etc. Sería la más adecuada para las aplicaciones orientadas a la consulta de información, por ejemplo para la realización de una enciclopedia electrónica.



- **Jerarquizado.** Combina las dos modalidades anteriores. Este sistema es muy utilizado, pues combina las ventajas de los dos sistemas anteriores (libertad de selección por parte del usuario y organización de la información atendiendo a su contenido, dificultad, etc.).



✓ **Según el nivel de control que tiene el profesional** (Ortí, 2006):

Los tipos de software según el menor o mayor nivel de control por parte del profesional son:

- **Programas cerrados:** Se componen por los programas informáticos que trabajan sobre un determinado contenido, y el profesional no tiene posibilidad de modificarlo y/o adaptarlo a las características de las personas con las que trabaja.
- **Programas semiabiertos:** Estas aplicaciones permiten que el profesional modifique algunas de las características del programa o tome decisiones sobre el itinerario a seguir.
- **Programas abiertos:** Son programas informáticos que, partiendo de un conjunto de posibilidades de actuación, permiten que el profesional fije el contenido concreto a desarrollar, pudiendo adaptarlo a las necesidades de las personas concretas que lo van a utilizar.

Se pretende realizar una multimedia que esté clasificada, según su sistema de navegación, como reticular, ya que se desea que el usuario tenga total libertad para seguir diferentes caminos cuando navega por el programa, atendiendo a sus necesidades, deseos y conocimientos. Así se logrará una mayor interactividad con el sistema. También estaría clasificada según el nivel de control del profesional, como programas abiertos, ya que ésta podrá adaptarlo a las necesidades de las personas que lo van a utilizar.

### 1.6 Tendencias tecnológicas actuales

Como parte de las nuevas tecnologías en la enseñanza están los Entornos Virtuales de Aprendizaje (EVA), las clases por televisión, las presentaciones en Power Point, la multimedia, las bibliotecas electrónicas, etc. La UCI cuenta con un EVA para la enseñanza- aprendizaje de los estudiantes y profesores de la universidad. Éste está dividido en dos categorías: formación de pregrado y formación de posgrado. Los cursos se montan a partir de un diseño instruccional, el cual soporta la colaboración entre aprendices, y entre el profesor y los estudiantes.

#### 1.6.1 Herramienta a utilizar para la creación del Software Multimedia

Para la creación de la Multimedia Interactiva Métricas de Software, primero se realizó un estudio de un total de 3 herramientas, las cuales son:

**ToolBook:** Sistema para desarrollo de aplicaciones destinadas a Windows que emplea una metáfora de "página y libro", análoga a la de "tarjeta y pila" de HyperCard. Utiliza un lenguaje de programación propio: OpenScript. En él se puede interactuar con parámetros que controlan el aspecto y el comportamiento de

la aplicación, tales como la composición de la página, las puntuaciones y el método de distribución, entre otros. Una interesante característica de este asistente es que permite ser configurado a gusto propio, de forma tal que la siguiente vez que se tenga que crear un nuevo proyecto, el asistente trabajará acorde a lo que se haya especificado.

Aunque ToolBook es una aplicación muy flexible, su principal desventaja radica en el hecho de tener muy poca compatibilidad y mucha programación.

**Macromedia director:** Programa para la creación de todo tipo de aplicaciones multimedia, utilizado principalmente para crear CD y DVD interactivos. Permite incorporar a las películas múltiples formatos, como imágenes (JPEG, BMP, PNG, GIF...), vídeos (MOV, AVI...), sonidos (WAV, AIFF...) o animaciones Flash. Incluye editores básicos para texto, mapa de bits, vectores, sonido.

Éste está firmemente integrado con otros productos y servidores de la familia Macromedia MX. Además de añadir soporte para Flash MX 2004, Director también tiene la capacidad de presentación y edición en Flash y Fireworks para hacer posible un flujo de trabajo racionalizado.

Pero a pesar de todas sus ventajas, tiene como principal desventaja que no puede ser visualizado en una plataforma con sistema operativo Linux instalado, por lo que los usuarios de este tipo de sistema no tendrían acceso al contenido hecho utilizando esta herramienta.

**Macromedia Flash 8:** Potente herramienta creada por Macromedia que ha superado las mejores expectativas de sus creadores. En sus versiones recientes se ha ampliado más allá de las animaciones simples, convirtiéndola en una herramienta de desarrollo completa para crear principalmente elementos multimedia e interactivos para Internet. Esta herramienta permite a los diseñadores y desarrolladores integrar video, texto, audio y gráficos en experiencias dinámicas que le permiten al cliente adentrarse en su vivencia y que producen resultados superiores para marketing y presentaciones interactivas, aprendizaje electrónico e interfaces de usuario de aplicaciones.

Luego de un estudio de cada una de estas herramientas, ha sido seleccionada como herramienta a utilizar Macromedia Flash, específicamente la versión 8.0, ya que además de que este trabajo es una continuidad de un proyecto que tiene ya definida como herramienta flash, ésta es la más usada a nivel

mundial por todas las mejoras que aporta: facilidad de manejo, mayor potencia gráfica y de integración con programas de edición de imágenes, facilidad para importar vídeo entre otras. Flash tiene muchas ventajas (Aula Clic, 2006):

- ✓ **Diseños más atractivos:** Flash 8 permite el uso de efectos visuales que nos facilitarán la creación de animaciones, presentaciones y formularios más atractivos y profesionales.
- ✓ **Optimización de fuentes:** Incorpora también opciones de legibilidad para fuentes pequeñas, haciendo más agradable la lectura de nuestros textos y de alta legibilidad.
- ✓ **Mayor potencia de animación:** Flash 8 permite un mayor control de las interpolaciones, habilitando un modo de edición desde el que se podrá modificar independientemente la velocidad en la que se apliquen los diferentes cambios de rotación, forma, color, movimiento, etc., de nuestras interpolaciones.
- ✓ **Mayor potencia gráfica:** Evita la repetición innecesaria de la representación de objetos vectoriales simplemente señalando un objeto como mapa de bits.
- ✓ **Mejoras en la importación de vídeo:** Para facilitar el resultado con formatos de vídeo, Flash 8 incluye un códec independiente de calidad superior capaz de competir con los mejores códecs de vídeo actuales con un tamaño de archivo mucho más pequeño.
- ✓ **Asistente de ActionScript:** El Asistente de ActionScript ha vuelto. Fue eliminado en la versión anterior, pero se ha vuelto a recuperar, y de forma mejorada, en esta última.

### 1.6.2 Metodología a utilizar para el desarrollo del software

Para hacer la selección de la metodología a utilizar, se realizó un estudio de 3 de las metodologías existentes en el mundo, realizando una comparación entre ellas a partir de los siguientes criterios de comparación:

- ✓ Carga de trabajo.
- ✓ Obtención de los requisitos.

- ✓ Relación con el cliente.
- ✓ Conocimiento del autor.
- ✓ Antecedentes en colección multimedia.

**Tabla 1: Comparación entre las metodologías estudiadas**

Criterios	FDD <sup>5</sup>	XP <sup>6</sup>	RUP <sup>7</sup>
Carga de trabajo	Genera la documentación mínima necesaria para comprender el código y depende totalmente del jefe del proyecto	Proceso ligero, no se le asignan roles organizativos al equipo	Basado en la documentación, la cual varía según las exigencias del cliente
Obtención de requisitos.	Define cómo se va a proceder después de recoger los requisitos definidos por el usuario, es decir, él no define requisitos.	Se Basa en la historia de casos de uso, definen los detalles técnicos sin meterse con los detalles de implementación	Se basa en los casos de uso donde se describen los requerimientos de la aplicación desde el punto de vista del usuario
Relación con el cliente	A diferencia del XP, existe el modelo general desarrollado en la primera fase, la	Después de cada iteración el cliente recibe un pedazo de programa funcional, así	Al final de cada fase, se le presentan al cliente los artefactos finales de dicha fase, para que sean

<sup>5</sup> FDD: Feature Driven Development

<sup>6</sup> XP: Extreme Programming

<sup>7</sup> RUP: Rational Unified Process

	que provee un marco general dentro del cual evolucionará el proyecto	el cliente está informado permanentemente	evaluados por éste y se puedan generar las iteraciones necesarias para la siguiente fase.
Conocimiento del autor	Escaso	Escaso	Abundante
Antecedentes en colección multimedia	No	No	Si

Después del análisis de las metodologías comparadas, ha sido elegida la metodología RUP como metodología de desarrollo de la aplicación multimedia, ya que además de ser la que está en la cima a nivel mundial, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. RUP permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Esto permite a los grupos de desarrollo producir aplicaciones informáticas más robustas y flexibles que se adaptan a las necesidades de los usuarios.

### 1.6.3 Lenguajes de Modelado

#### UML (Lenguaje Unificado de Modelado)

El Lenguaje unificado de modelado es un lenguaje gráfico para visualizar, especificar y documentar cada fase del software orientado a objetos. Éste ofrece una forma de modelar elementos conceptuales tales como los procesos de negocio y funciones de sistema, además de aspectos concretos como escribir clases en un lenguaje determinado, esquemas de base de datos y componentes de software reusables.

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

El Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L) se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos. Además el 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 de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. (SAUER, 2004).

OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia un tipo de diagrama en particular. Estas vistas son: (BRUNO, 2005)

- ✓ **Lógica** : Modelada a través del Diagrama de Clases de OMMMA-L extendido del Diagrama de Clases de UML.
- ✓ **Presentación Espacial**: Modelada a través de los diagramas de presentación.
- ✓ **Comportamiento Temporal Predefinido**: Modelada por el diagrama de secuencia. Este modela una secuencia de una presentación predefinida dentro de una escena.
- ✓ **Control Interactivo**: Modelado a través del Diagrama de Estado.

#### **1.6.4 Lenguajes utilizados en la multimedia**

**XML (Lenguaje de Marcas Extensible)**: Lenguaje de marcas para documentos que contengan información estructurada. La Información estructurada contiene (palabras, imágenes, etc.) y alguna indicación sobre el papel que desempeña el contenido, (por ejemplo, el contenido en una partida tiene un significado distinto al contenido en una nota a pie de página, lo que significa algo diferente que el contenido en una figura título o el contenido en una tabla de base de datos, etc.). Un lenguaje de marcas es un mecanismo para identificar estructuras en un documento. (Walsh, 2006)

La especificación XML define una manera estándar de añadir marcas a los documentos. XML ni especifica semántica, ni una etiqueta de conjunto. De hecho, XML es realmente un meta-lenguaje para

describir lenguajes de marcas. En otras palabras, XML proporciona un mecanismo para definir las etiquetas y relaciones estructurales. (Walsh, 2006)

**ActionScript** es el lenguaje de programación orientado a objetos (OOP) utilizado en Macromedia Flash. Hoy en día los lenguajes ActionScript permiten hacer con Flash interactividad a niveles antes impensables. ActionScript es un lenguaje de script, esto significa que no requiere la creación de un programa completo para que la aplicación alcance los objetivos. El lenguaje está basado en especificaciones de estándar de industria ECMA-262, un estándar para Javascript, de ahí que ActionScript se parezca tanto a Javascript. (Curso ActionScript, 2009)

### Conclusiones parciales

En este capítulo se realizó un estudio sobre las principales tendencias tecnológicas actuales para aplicaciones multimedia, así como los lenguajes para el desarrollo de las mismas. Luego de este estudio, se determinó utilizar RUP como metodología, como herramienta de desarrollo, Flash 8, como lenguaje de programación, ActionScript 2.0 y XML como lenguaje de marcas, además se propuso como metodología de desarrollo RUP y como lenguaje de modelado OMMMA-L. Se analizaron también los conceptos necesarios para una mejor realización del trabajo, tales como los de software educativo, multimedia, hipertexto e hipermedia. También se dieron algunos conceptos esenciales de métricas de software. Se realizó un estudio de las posibles soluciones existentes en el mundo, y se llegó a la conclusión de que las universidades estudiadas no usan la multimedia como medio de enseñanza.

# Capítulo 2: Descripción de la solución.

## Introducción

Este capítulo está destinado a describir la solución propuesta para dar cumplimiento al problema planteado. En éste se hace una descripción del contenido a presentar en la multimedia y se presenta el mapa conceptual para un mejor entendimiento del contenido. Se realiza una descripción del modelo de dominio y se analizan los conceptos del mismo. Se muestra además el diagrama de navegación de la multimedia y también el diagrama de casos de uso del sistema y una breve descripción de los mismos.

### 2.1 Especificación de la audiencia

Lo primero es preguntarse ¿A quién va dirigida la aplicación? Esta pregunta permite identificar el usuario final a quien va dirigido el producto, teniendo siempre en cuenta que el diseño del sistema debe satisfacer las necesidades del usuario. Por eso, la identificación de la audiencia es uno de los aspectos fundamentales que se debe tener en cuenta a la hora de desarrollar un producto software.

La aplicación “Métricas de software” está dirigida no sólo a los estudiantes de 3er año de la universidad que estén involucrados en el proceso de desarrollo de software; sino también a otros profesionales que se desarrollan en la misma rama.

### 2.2 Especificación del Contenido

Para su estudio la temática se dividió en 4 temas y éstos en subtemas, como se describe a continuación:

Tema 1: Fundamentos de métricas.

Este tema está conformado por cuatro subtemas, los cuales le brindan al usuario los principales conceptos, características, importancia de las métricas de software.

Tema 2: Medición del proceso de software.

El tema dos está compuesto por cinco subtemas principales: Introducción, Establecer y mantener compromiso de medición, Plan de Medición del proceso, Realizar la Medición del proceso y Evaluar medición. El usuario podrá conocer cómo establecer y mantener el compromiso de medición, los pasos a seguir para realizar el plan de medición y también cómo realizar y evaluar el proceso de medición. Además, el epígrafe “Realizar la medición del proceso”, contendrá dos subtemas: Técnicas de medición y Herramientas de medición. En éstos se explican las técnicas de medición que existen y las herramientas más usadas para la realizar la medición del proceso. Además de los ejercicios del tema.

Tema 3: Clasificación de Métricas

Este tema está compuesto por cuatro subtemas: Introducción, Métricas del Producto, Métricas del Proyecto, y Métricas del Proceso. En ellos se explican cada una de estas clasificaciones y además en tema cuenta con ejercicios para autoevaluarse.

Tema 4: Ejemplos de métricas.

Este tema está compuesto por varios ejemplos de métricas, para lograr que el usuario tenga una idea más clara de lo que es una métrica de software y cómo funciona. Consta de dos subtemas: Métricas de calidad (ISO 9126-3) y Resumen de métricas de PSP.

### 2.2.1 Mapa Conceptual

Además de los temas antes mencionados, el producto cuenta también con un mapa conceptual para una mejor comprensión del contenido. Un mapa conceptual es una agrupación de conceptos que se relacionan entre sí.

Con un mapa conceptual se relacionan conceptos que ya poseemos con otros nuevos, logrando de esta forma un aprendizaje significativo y no memorístico.

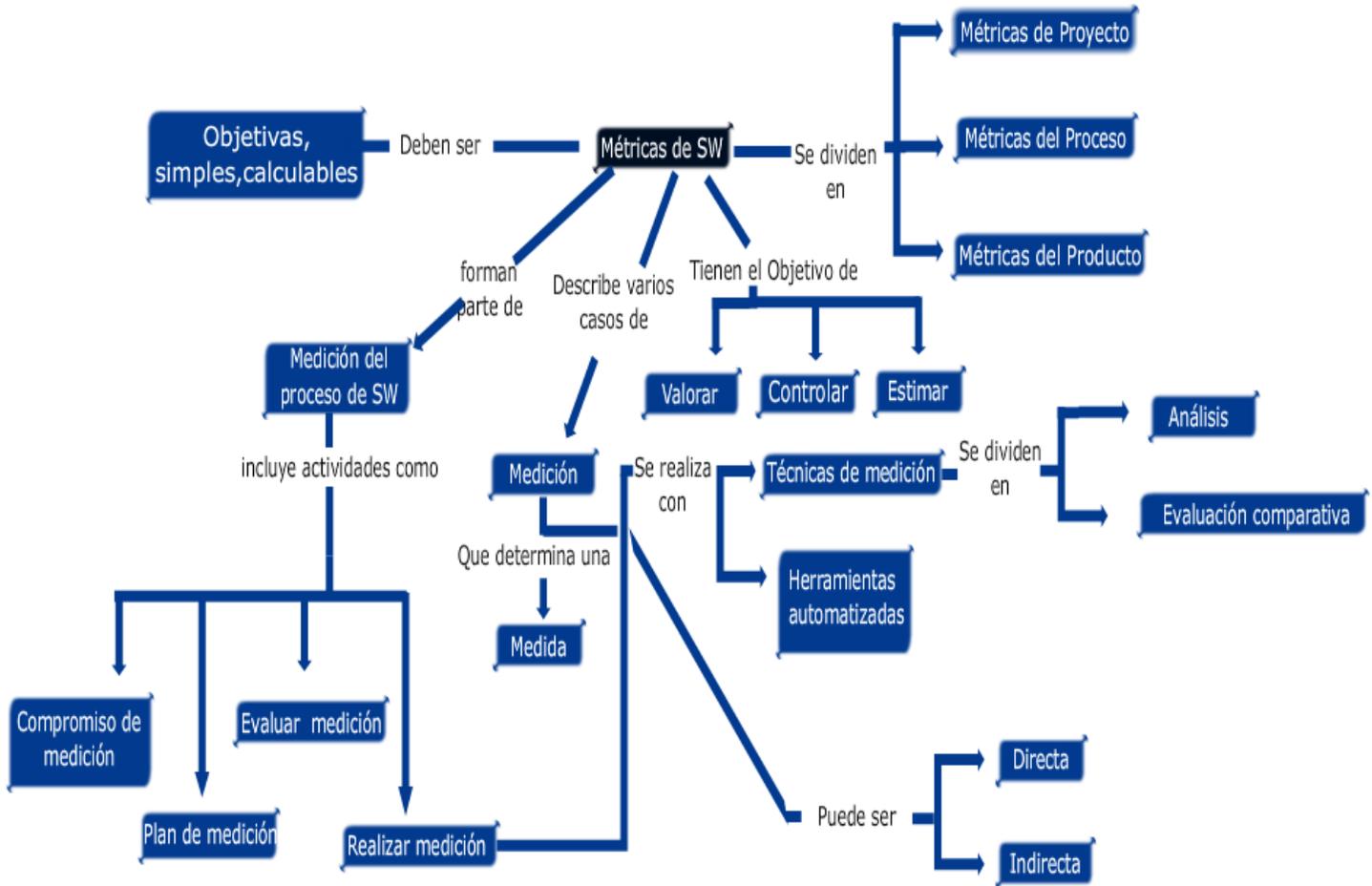


Fig. 4: Mapa Conceptual

### 2.3 Descripción del Modelo de Dominio

Debido al bajo nivel de estructuración del negocio, se propone un modelo de dominio que permite, mediante la representación visual de conceptos, un mejor entendimiento del sistema.

La descripción del modelo de dominio se realiza a través de un modelo conceptual que se representa en un diagrama de clases UML.

### 2.3.1 Diagrama de clases del dominio

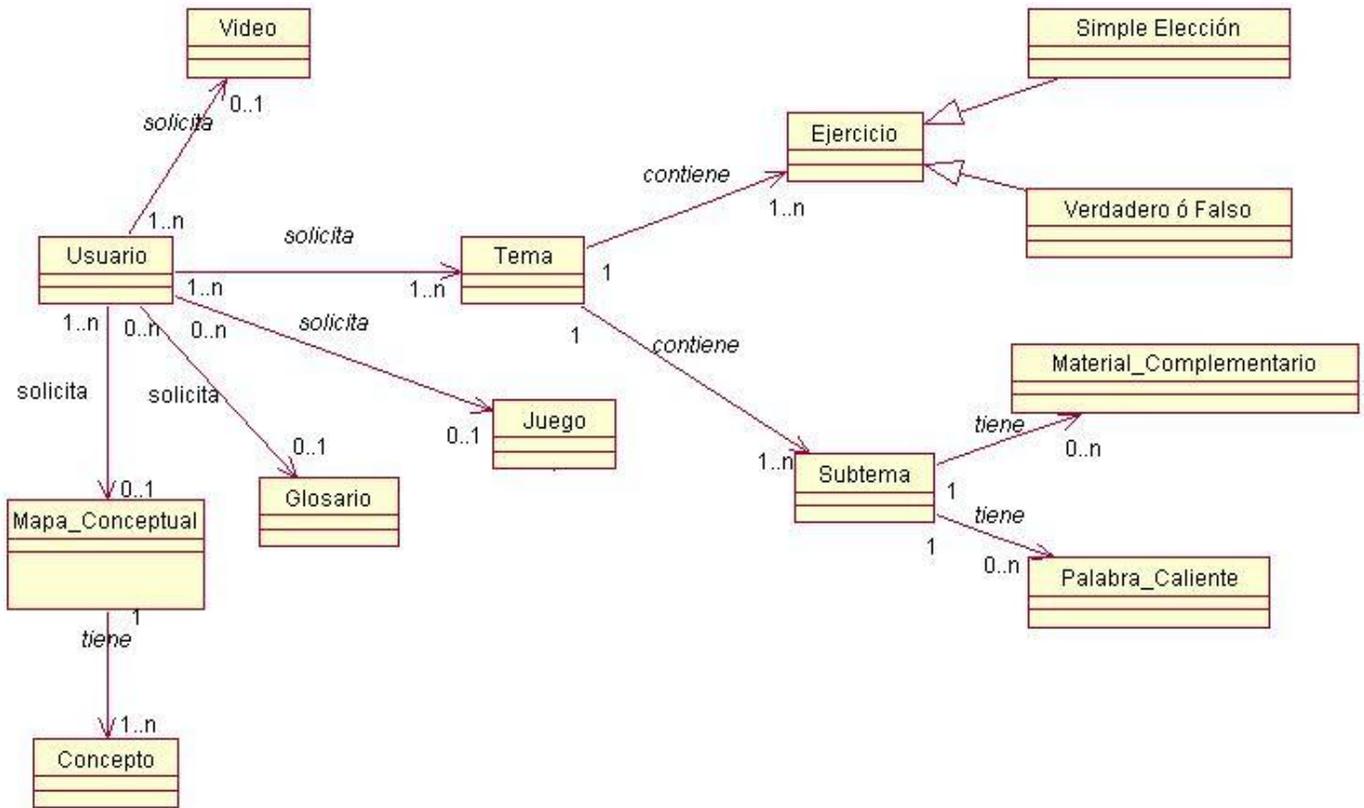


Fig. 5: Diagrama de dominio

### 2.3.2 Análisis de los conceptos del Dominio

Descripción de los conceptos utilizados en el diagrama de clases del modelo de dominio.

**Usuario:** Toda persona que interactúe con la aplicación.

**Video:** Videoconferencia relacionada con el tema de métricas de software.

**Mapa Conceptual:** Representación gráfica de conceptos relacionados entre sí.

**Temas:** Contenido de la multimedia que está dividido en 4 áreas: Fundamentos de métricas, Medición del proceso de software, Ejemplos de métricas y Métricas definidas en el método de estimación en la UCI.

**Subtemas:** Diferentes contenidos mostrados en cada tema.

**Material Complementario:** Representa conferencias, documentos o presentaciones; relacionadas con alguna de las temáticas que se abordará.

**Ejercicios:** Conjunto de ejercicios que sirven para evaluar cada uno de los temas.

**Palabras Calientes:** Hipertexto que abre una ventana flotante que profundiza en el significado de alguna palabra o concepto tratado en un subtema.

**Glosario:** Representación textual de conceptos organizados alfabéticamente.

**Juego:** Juego de conocimiento que sirve para profundizar los conceptos del mapa.

### 2.4 Descripción del sistema propuesto

Para lograr un buen producto, es necesario conocer las preferencias del usuario para realizar una aplicación que se ajuste a las necesidades de éste. A continuación se presenta un listado de los requisitos funcionales y no funcionales que se necesitan para realizar la multimedia.

#### 2.4.1 Requisitos Funcionales

En la siguiente tabla se exponen los requisitos funcionales planteados por el usuario.

**Tabla 2: Requisitos funcionales**

Referencia	Función
R1	Mostrar una pantalla de presentación con el nombre del producto que de inicio a la pantalla principal.
R2	Mostrar la pantalla de ayuda en la cual se explicará el curso que el usuario estudiará y cómo navegar la multimedia.
R3	Reproducir música de fondo automática de forma predeterminada.
R4	Permitir al usuario pausar y reanudar música de fondo.
R5	Permitir la navegación por los temas y subtemas.

## Capítulo 2: Descripción de la solución

R6	Mostrar los temas que conforman la aplicación.
R7	Permitir al usuario seleccionar un subtema deseado.
R8	Mostrar información contenida dentro del subtema seleccionado por el usuario.
R9	Permitir al usuario desplazar el scroll de texto para una completa visualización de la información en aquellas pantallas que requieran el uso del mismo.
R10	Permitir al usuario realizar ejercicios sobre los contenidos estudiados en la aplicación.
R11	Mostrar mensaje de respuesta correcta e incorrecta.
R12	Permitir al usuario ejecutar el juego que se encuentra disponible.
R13	Permitir al usuario visualizar materiales complementarios.
R14	Visualizar video conferencia.
R15	Controlar sonido del video visualizado por el usuario.
R16	Permitir al usuario maximizar video visualizado.
R17	Permitir visualizar mapa conceptual.
R18	Mostrar concepto asociado al nodo seleccionado por el usuario en el mapa conceptual.
R19	Permitir siempre ver los temas desde cualquier pantalla en que se encuentre el usuario.
R20	Visualizar información referente a palabras calientes consultadas por el usuario.
R21	Visualizar glosario de términos.
R22	Permitir salir de la aplicación en cualquier momento que el usuario desee.
R23	Mostrar créditos al salir de la aplicación.

### 2.4.2 Requisitos no funcionales

Los requisitos no funcionales son propiedades o cualidades que el producto debe cumplir. Son las cualidades que hacen que el producto sea usable, rápido, confiable y atractivo.

#### ✓ Requisitos de Apariencia

- Utilizar botones que expresen su función, ya sea que se intuya o expresados con texto.
- El sistema debe contener una interfaz amigable al usuario, brindándole todas las facilidades necesarias para una mayor comodidad.
- 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.

#### ✓ Requisitos de Usabilidad

Los usuarios que utilicen la aplicación deberán tener conocimiento previo de trabajo con sistemas operativos visuales.

#### ✓ Requisitos de Software

En la siguiente tabla se enuncian los sistemas operativos (Windows y Linux) sobre los cuales la aplicación podrá ser ejecutada, así como los navegadores con los cuales la misma podrá ser visualizada.

**Tabla 3: Sistemas Operativos y navegadores necesarios para ejecutar la aplicación**

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
<b>Linux</b>	
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.

### ✓ Requisitos de Hardware

Los requisitos mínimos 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.

### ✓ Requisitos de Diseño e Implementación

- La herramienta a usar en el diseño gráfico de la aplicación será la Macromedia Flash 8.
- El contenido se cargará desde archivos XML y el lenguaje de programación será ActionScript 2.0.

### ✓ 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.

## 2.5 Diagrama de navegación

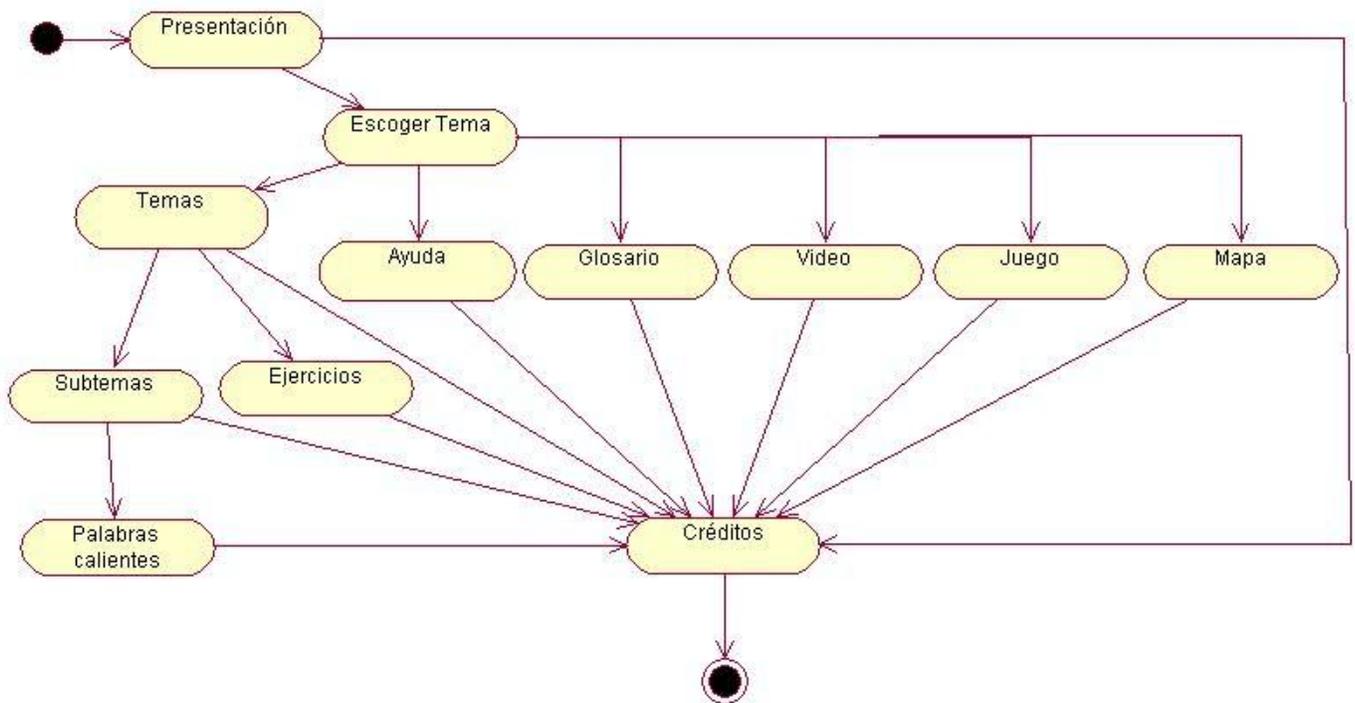


Fig. 6: Diagrama de navegación

### 2.6 Modelo de Casos de uso del sistema

El Modelo de Casos de Uso del sistema contiene actores, casos de uso y sus relaciones. Representa un esquema donde se recogen las funcionalidades que se automatizan y determina cómo será utilizado desde el punto de vista del usuario (Actor), pues se construye sobre la base de sus necesidades.

#### 2.6.1 Determinación y justificación de los Actores del Sistema

A continuación se muestra la justificación del actor del sistema.

Tabla 4: Actor y justificación

Actor	Justificación
Usuario	Es la persona que va a usar el sistema para buscar información sobre métricas de software.

### 2.6.2 Diagrama de Casos de Uso

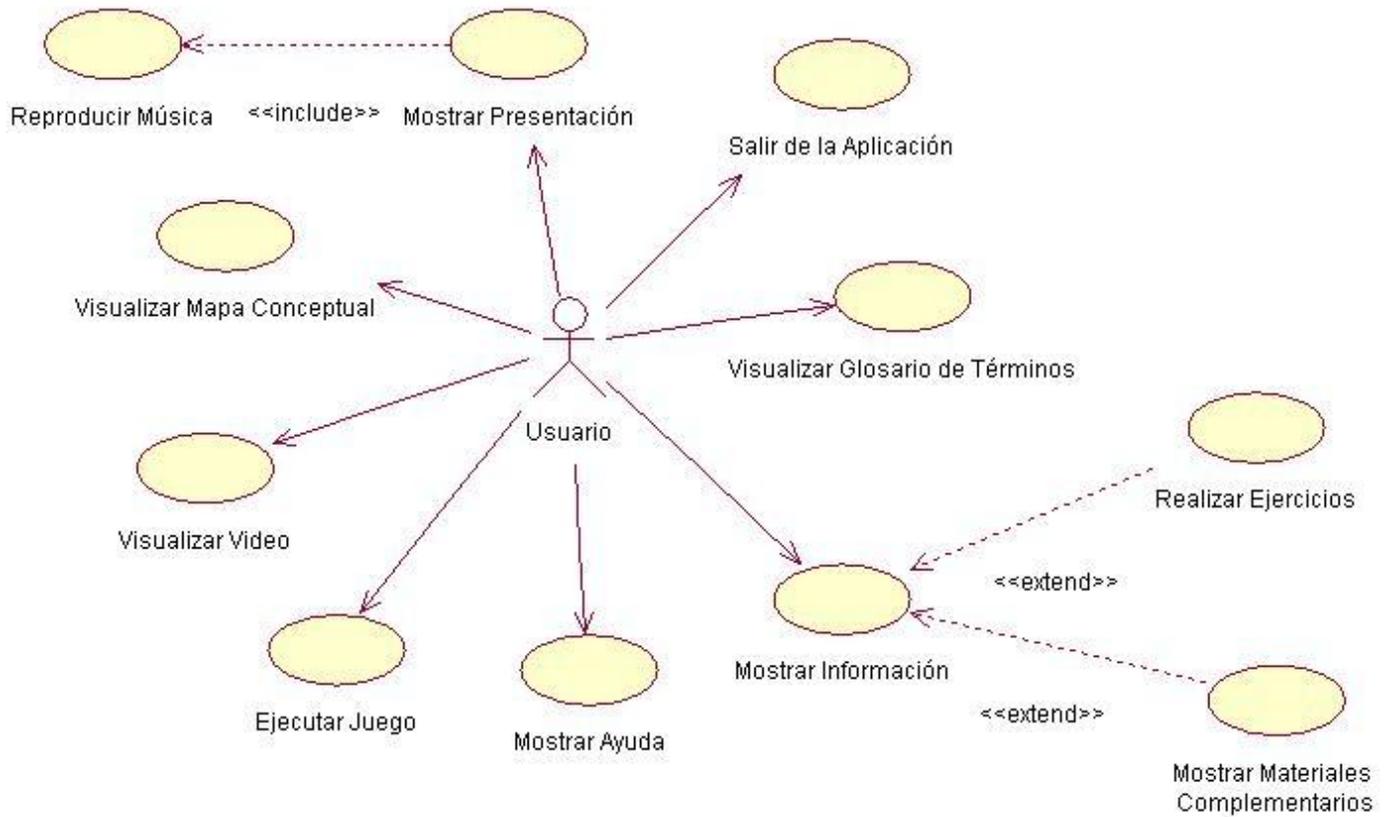


Fig. 7: Diagrama de CU

### 2.6.3 Breve descripción de los casos de uso del sistema.

Para lograr un mayor entendimiento del funcionamiento de cada uno de los casos de uso (CU), se realiza una breve descripción de cada uno, lo cual se muestra a continuación.

**Tabla 5: CU Mostrar Presentación**

<b>Caso de Uso:</b>	<b>Mostrar Presentación</b>
Actor(es):	Usuario
Propósito:	Mostrar la presentación de la multimedia.
Resumen:	Al ejecutar la multimedia se muestra la presentación, la cual muestra el nombre del curso. Al concluir, pasa automáticamente a la pantalla principal del curso.
Referencias:	R1

**Tabla 6: CU Mostrar Ayuda**

<b>Caso de Uso:</b>	<b>Mostrar Ayuda</b>
Actor(es):	Usuario
Propósito:	Mostrar una ayuda de la multimedia.
Resumen:	El caso de uso comienza cuando el usuario da clic en el botón "Ayuda" y automáticamente se carga una pantalla que contiene la ayuda al usuario. Ésta muestra una explicación al usuario sobre el contenido de la multimedia, además les explica como navegar en la misma. El caso de uso termina cuando el usuario cierra la ayuda.
Referencias:	R2

**Tabla 7: CU Visualizar Glosario de Términos**

<b>Caso de Uso:</b>	<b>Visualizar Glosario de Términos</b>
Actor(es):	Usuario
Propósito:	Mostrar el glosario de términos que contiene las palabras que puedan crear duda tratadas en el curso y sus conceptos.
Resumen:	El caso de uso comienza cuando el actor hace clic sobre el botón “Glosario de Términos” y automáticamente se carga una pantalla que contiene el glosario. Éste muestra el abecedario, el mismo da la posibilidad de escoger la letra con la que empieza la palabra que desee buscar, cuando de clic sobre esta letra se mostrarán los conceptos relacionados con esta. El caso de uso termina cuando el usuario cierra el glosario.
Referencias:	R21

**Tabla 8: CU Mostrar Información**

Actor(es):	Usuario
Propósito:	Mostrar la información contenida en la multimedia.
Resumen	El usuario de la multimedia selecciona la información que desea. Puede ser subtemas dentro de cada uno de los temas, conceptos de las palabras calientes, materiales complementarios, ejercicios, etc. Para poder mostrar la información, el usuario tendrá que hacer clic sobre los botones que contienen la información deseada y el sistema se encargará de obtener y mostrar la información, cargándola dinámicamente desde archivos XML.
Referencias:	R5,R6,R7,R8,R9,R19,R20

**Tabla 9: CU Visualizar Mapa Conceptual**

<b>Caso de Uso:</b>	<b>Visualizar Mapa Conceptual</b>
Propósito:	Mostrar el mapa conceptual que contiene los principales conceptos del curso.
Actor(es):	Usuario
Resumen:	El caso de uso comienza cuando el actor hace clic sobre el botón Mapa y automáticamente se carga una pantalla que contiene el mapa conceptual. Este permite visualizar el significado de cada palabra en una ventana flotante. El caso de uso termina cuando el usuario cierra el mapa.
Referencias:	R17, R18

**Tabla 10: CU Ejecutar Juego**

<b>Caso de Uso:</b>	<b>Ejecutar juego</b>
Actor(es):	Usuario
Propósito:	Abrir el juego que está disponible en la aplicación.
Resumen	El caso de uso se inicia cuando el usuario da clic en el botón "Juego" que se encuentra en la barra que contiene todos los botones. A continuación aparecerá el juego del ahorcado, el cual consiste en adivinar una palabra determinada relacionada con el tema tratado en el curso. Éste irá guardando en la memoria de la máquina un historial de valores positivos o negativos, dependiendo de la cantidad de aciertos o fallas con respecto a las palabras a adivinar. El caso de uso termina cuando el usuario decide salir del juego y da clic en el

	botón salir.
Referencias:	R12

**Tabla 11: CU Visualizar Video**

<b>Caso de Uso:</b>	<b>Visualizar Video</b>
Actor(es):	Usuario
Propósito:	Muestra la video conferencia que aparece en la aplicación
Resumen	El caso de uso se inicia cuando el usuario da clic en el botón "Video", automáticamente se carga la pantalla con dicho video, la cual debe permitir opciones como: parar, reproducir, pausar, subir o bajar volumen del mismo.
Referencias:	R14, R15, R16

**Tabla 12: CU Salir de la Aplicación**

<b>Caso de Uso:</b>	<b>Salir de la Aplicación</b>
Actor(es):	Usuario
Propósito:	Permite al usuario salir de la aplicación en cualquier momento
Resumen	El caso de uso se inicia cuando el usuario da clic en el botón de "Cerrar", automáticamente se carga la pantalla de cerrar presentación la cual muestra la pregunta que si desea salir realmente de la misma. Si da clic en sí, muestra los créditos y, si da clic en no, vuelve para la pantalla en la que estaba anteriormente.

Referencias:	R22, R23
--------------	----------

**Tabla 13: CU Realizar Ejercicios**

<b>Caso de Uso:</b>	<b>Realizar Ejercicios</b>
Actor(es):	Usuario
Propósito:	Permite al usuario hacer ejercicios relacionados con el tema
Resumen	El caso de uso se inicia cuando el usuario selecciona la opción realizar ejercicios, automáticamente se carga la pantalla de los ejercicios y, cuando el usuario responda los mismos, se le informará en cada uno si la respuesta es correcta o incorrecta.
Referencias:	R10, R11

**Tabla 14: CU Mostrar Materiales Complementarios**

<b>Caso de Uso:</b>	<b>Mostrar Materiales Complementarios</b>
Actor(es):	Usuario
Propósito:	Muestra los materiales complementarios contenidos en la aplicación
Resumen	El caso de uso se inicia luego de que el usuario estudia algún tema y se le recomienda ver algún documento o presentación que está disponible en la aplicación. Éste da clic encima de la palabra que indica que ahí está el material y éste se muestra automáticamente. El caso de uso termina cuando el usuario cierra el material recomendado.

Referencias:	R13
--------------	-----

**Tabla 15: CU Reproducir Música**

<b>Caso de Uso:</b>	<b>Reproducir Música</b>
Actor(es):	usuario
Propósito:	Reproducir música automática
Resumen	El caso de uso comienza cuando el usuario ejecuta la multimedia, la aplicación reproduce la música de forma automática y le da la posibilidad al usuario de pausar, reanudar y controlar el volumen de la misma.
Referencias:	R3, R4

### Conclusiones Parciales

Con el desarrollo de este capítulo, se ha logrado una mayor comprensión del sistema a través de la realización de un modelo de dominio donde se representan de forma visual los conceptos más importantes para un mejor entendimiento del sistema. También se creó un mapa conceptual. Se identificaron los requisitos funcionales y no funcionales, además se identificaron y describieron los casos de uso para darle cumplimiento a los requisitos. Con la culminación de este capítulo, ya se puede pasar al diseño e implementación de la aplicación.

### **Capitulo 3: Solución de la descripción propuesta**

#### **Introducción**

En el presente capítulo se estudiarán las normas y principios de diseño que se tendrán en cuenta para el desarrollo de la aplicación. Se muestran los diagramas de presentación como parte de OMMMA-L, el diagrama de componentes como parte de la implementación. También se presenta la estructura de los archivos XML que se utilizarán para mostrar el contenido del producto.

#### **3.1 Análisis de la Arquitectura utilizada**

El diseño de interfaces es una labor importante dentro del proceso de desarrollo de un software. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.

##### **3.1.1 Principios de diseño**

Con el objetivo de desarrollar la interfaz del sistema, se analizaron aspectos importantes que posibilitan el bienestar de todo usuario. Para ello se tuvo en cuenta la organización, la distribución de la información, el público al que va dirigido y la temática que se aborda.

Los usuarios que interactúan con el sistema lo hacen por medio de una interfaz gráfica, por tal razón la misma es amena, sencilla, agradable, presenta facilidad de uso, logrando que el usuario se sienta identificado con ella.

Las pantallas del sistema contienen la información necesaria para evitar la sobre carga, además de mantener las opciones principales en el mismo lugar de la interfaz para una mejor interacción y adaptabilidad del usuario con la aplicación.

El sistema tiene colores oscuros y poco llamativos, pues son los que están a fin con la temática. Tiene como color predeterminado el azul.

### **3.1.2 Estándares para el diseño de la interfaz de usuario**

La interfaz de la aplicación fue elaborada teniendo en cuenta algunos de los estándares comunes para aplicaciones multimedia. La aplicación contiene navegación, ayuda, opción de salir y opción de controlabilidad del sonido.

#### **Navegación**

En primer lugar, aparece un menú principal, el cual permite el libre acceso a los contenidos lineales de forma rápida.

En segundo lugar se considera el árbol de contenidos, puesto que al profundizar en las diferentes ramas, mediante diferentes pulsaciones del ratón, permitirá acceder a los diferentes subtemas y, al pulsar sobre uno de ellos, acceder a la página de contenidos del mismo.

#### **Ayuda**

El usuario puede consultarla desde cualquier pantalla ya que está disponible siempre. Ésta muestra una explicación del curso y cómo navegar en la multimedia para evitar que el usuario se pierda.

#### **Salir**

La opción salir es accesible desde cualquier pantalla, por lo que el usuario puede salir cuando lo desee.

#### **Controlabilidad del sonido**

La salida y entrada del fondo musical puede ser activada y desactivada por el usuario o puede ser controlado el volumen del mismo según el gusto del usuario.

#### **Color**

Los colores que predominan son el azul oscuro, negro y blanco, debido a que el sistema debe usar colores serios y poco llamativos. (Ver Anexos 8-14)

### 3.2 Diagrama de Presentación del modelo de análisis

Este es un nuevo artefacto dentro de la extensión OMMMA – L del lenguaje UML y sirve para describir la parte estática del modelo a través de una descripción intuitiva de la distribución espacial de objetos visuales de la interfaz de usuario.

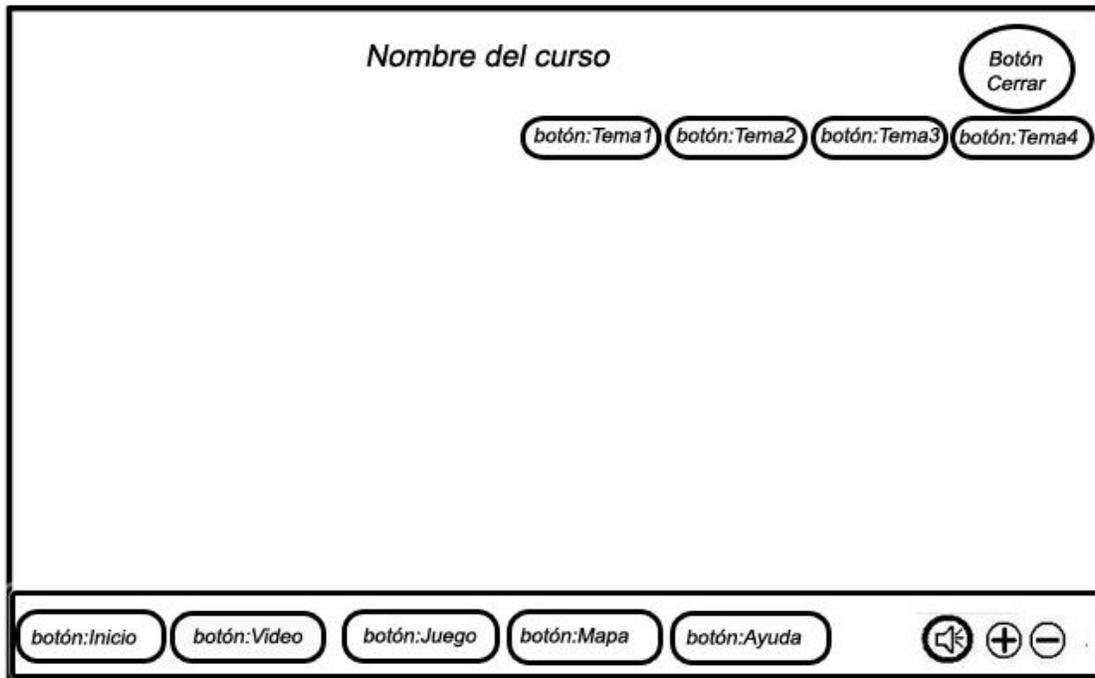


Fig. 8: Diagrama de Presentación Pantalla Principal

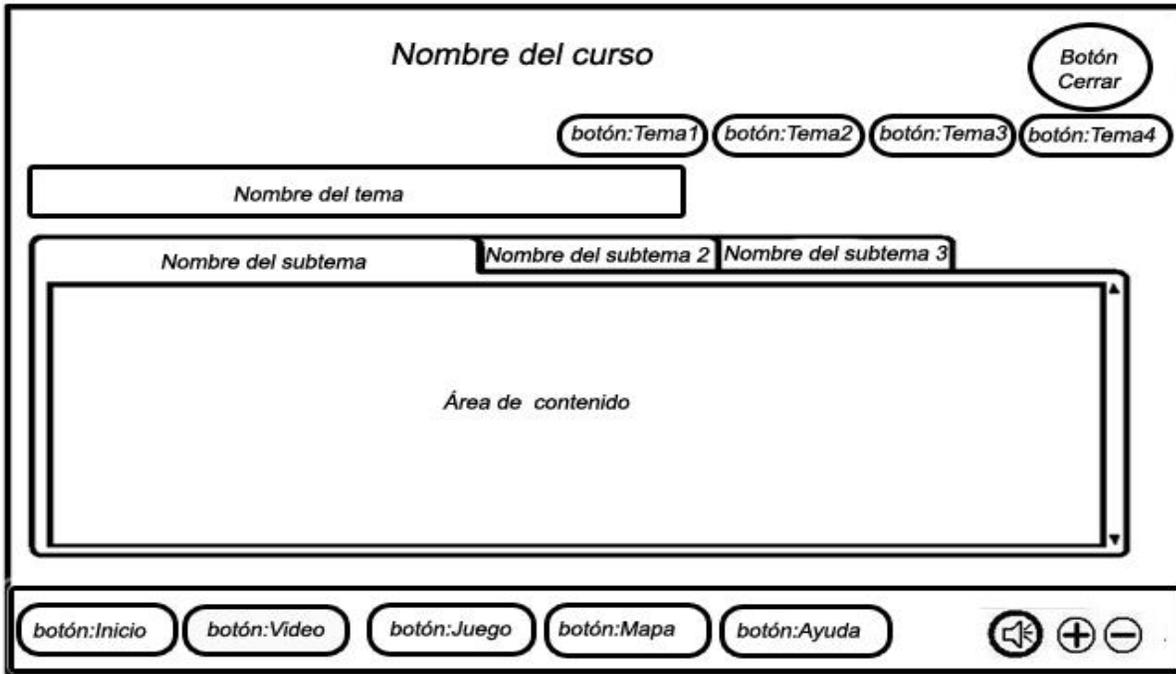


Fig. 9: Diagrama de Presentación Pantalla Tema

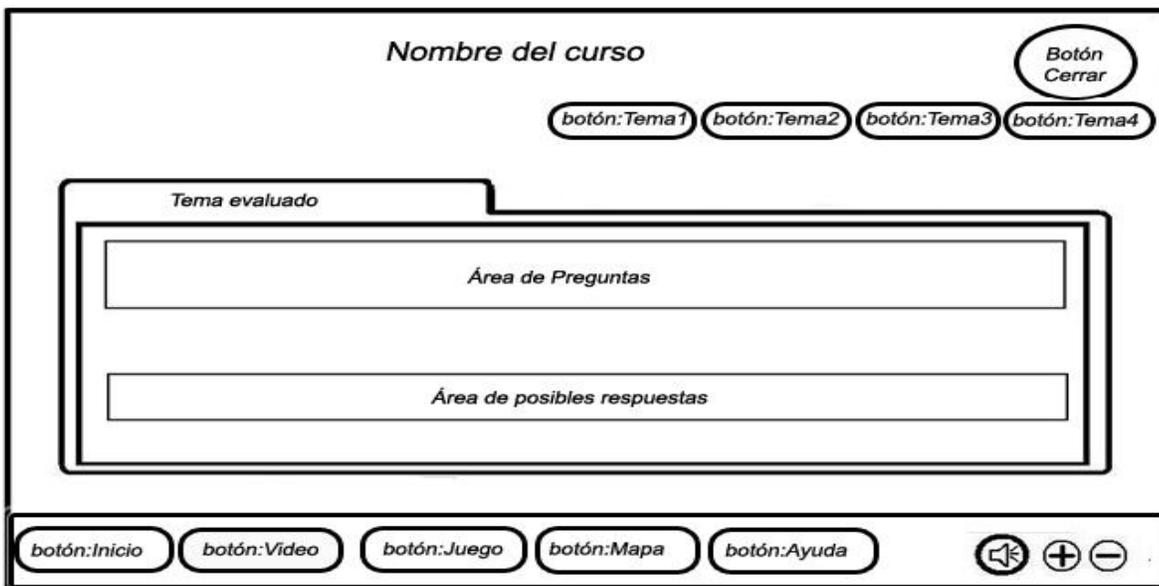


Fig. 10: Diagrama de Presentación Pantalla Ejercicio

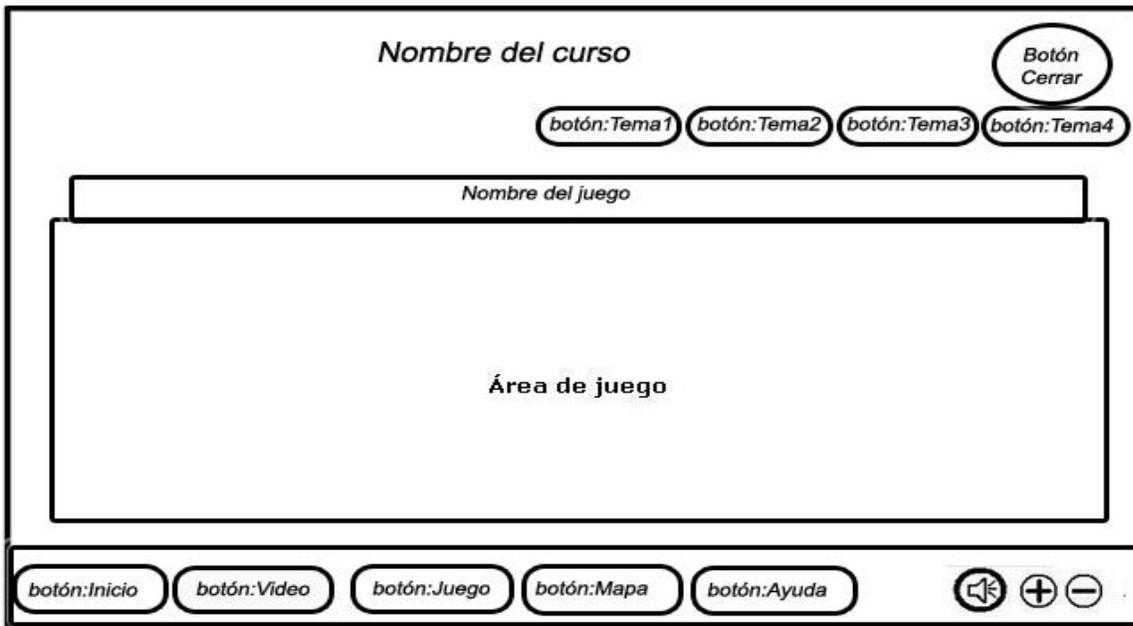


Fig. 11: Diagrama de Presentación Pantalla Juego

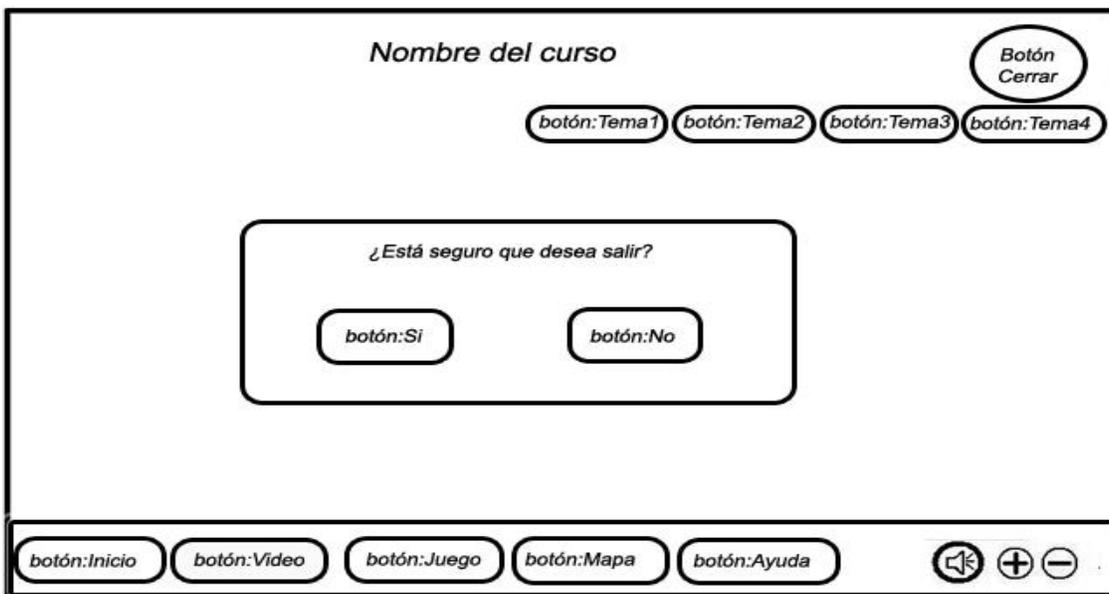


Fig. 12: Diagrama de Presentación Pantalla Salir

### 3.3 Descripción de Archivos XML

Para almacenar la información (textos e imágenes) de la multimedia se utiliza la tecnología XML, lo que permite modificar la información sin necesidad de interactuar con la aplicación.

#### Descripción de archivos XML Tema

Para cada tema se crea un archivo XML con la siguiente estructura:

```
<tema>
  <nombreTema><p align='center'><i><b>Nombre del tema</b></i>
</p></nombreTema>
  <subtema1><b><i>Nombre del subtema 1</i></b> </subtema1>
  .
  < subtema_n><b><i>Nombre del subtema n</i></b> </ subtema_n>
<texto_ subtema1> Texto que pertenece al subtema 1 del tema</texto_ subtema1>
  .
  .
<texto_ subtema_n> Texto que pertenece al subtema 1 del tema</texto_ subtema_n>
</tema>
```

Los XML de cada tema están estructurados por un nodo padre, por n nodos hijos donde n representa la cantidad de subtemas que tenga cada tema, y por n nodos hijos más que contendrán el texto de cada uno de estos n subtemas.

### Descripción de los archivos XML ayuda

```
<ayuda>
    <bienvenida><p align='center'><font color='#000000' size='22'></font></p>
Bienvenido al curso de Métricas de Software!!! </bienvenida>
<descripción><p align='center'><font color='#FF0000' size='15'></font></p>
Se explica al usuario el contenido que encontrará dentro de la multimedia y además como
puede navegar por la misma.
</descripción>
</ayuda>
```

### Descripción de archivos XML ejercicios

```
<quiz>
    <title> Nombre del título del ejercicio </title>
    <items>
        <item1>
            <question>Aquí estará la pregunta que se realizará en el ejercicio que será al
estilo Verdadero o Falso </question>
            <answer correct="y">Falso</answer>
            <answer>Verdadero</answer>
        </item1>
        <item2>
```

## Capítulo 3: Solución de la Descripción Propuesta

```
<question> Aquí estará la pregunta que se realizará en el ejercicio donde el
usuario seleccionará la respuesta correcta</question>

<answer>Aquí se pondrán las opciones de respuesta para el usuario </answer>

<answer correct="y">Aquí se pondrá la respuesta correcta </answer>

</item2>

<ítem n>.....</ítem n>

</items>

</quiz>
```

El XML está estructurado de la siguiente manera:

`<quiz>` este nodo da inicio al XML de contenido del ejercicio.

`</quiz>` cierre de nodo.

`<title>` nombre del título del ejercicio.

`<ítems>` nodo que va a contener todos los elementos de las preguntas de los ejercicios

`</ítems>` cierre de nodo.

`<ítem>` nodo que va a contener la pregunta y las respuestas como elemento.

`</ítem>` cierre de nodo.

`<question>` nodo que va a contener la pregunta en cuestión.

`</question>` cierre de nodo.

<answer> nodo que va a contener una posible respuesta a seleccionar.

<answer correct = "y"> nodo que va a contener la respuesta correcta, ya sea una selección de falso o verdadero o una selección de respuesta simple.

</answer> cierre de nodo.

### 3.4 Modelo de Implementación

El modelo de implementación denota la implementación actual del sistema en términos de componentes y subsistemas de implementación. Es natural mantener el modelo de implementación a lo largo de todo el ciclo de vida del software. Éste describe cómo se organizan los componentes de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y en el lenguaje o lenguajes de programación utilizados, y cómo dependen los componentes unos de otros.

#### 3.4.1 Diagrama de Componentes

El diagrama de componentes representa la separación de un sistema de software en componentes físicos. Se usa para modelar la estructura del software, incluyendo la dependencia entre los componentes del software. Éste muestra las dependencias lógicas entre componentes software, sean éstos componentes fuentes, binarios o ejecutables.

Para la realización de este diagrama, los componentes han sido agrupados por paquetes (Ver Anexos1-7), incluyendo el paquete de archivos XML, ya que esta tecnología fue usada para mostrar la información del producto.

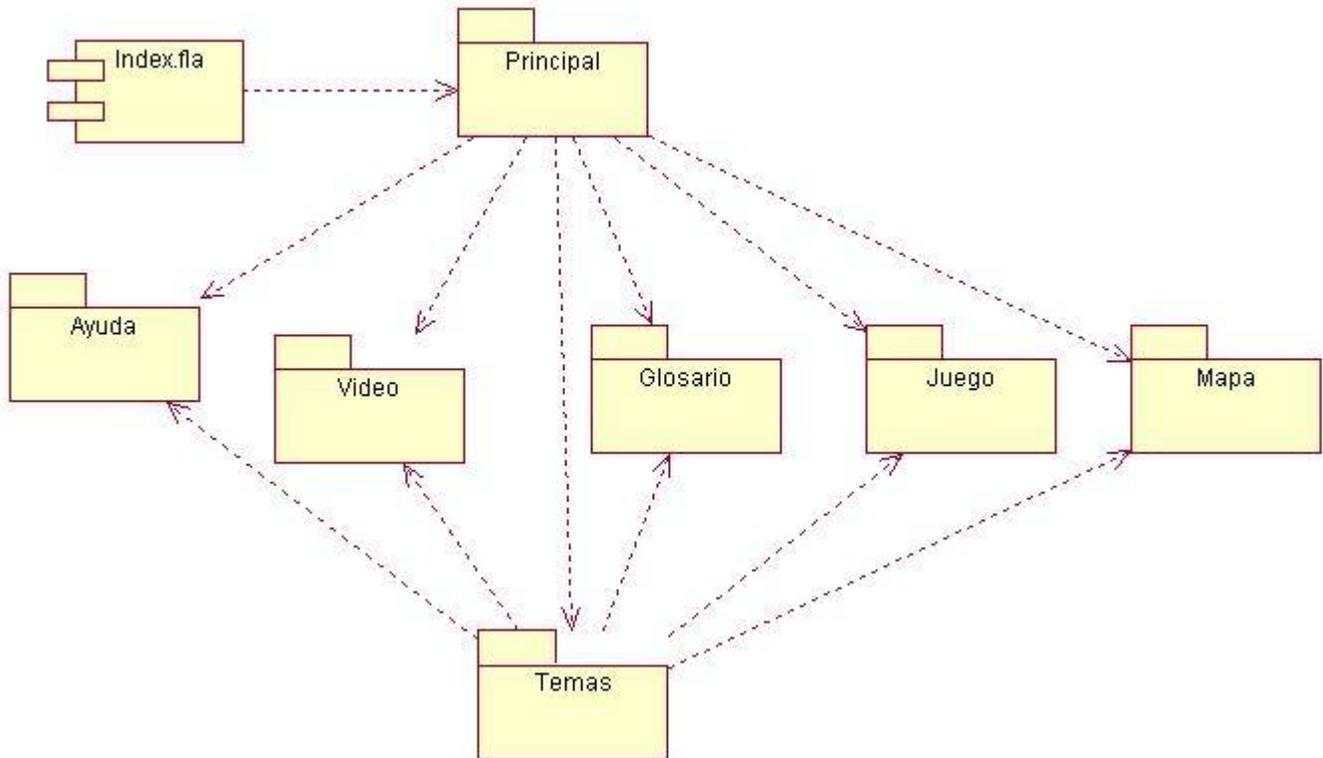


Fig. 13: Diagrama de componente general por paquetes

### Conclusiones Parciales

En este capítulo se realizó un análisis de los principios y estándares de diseño sobre los cuales se basó la aplicación, así como la estructura de los archivos XML usados. Se dio una visión clara de la multimedia a través de los diagramas de presentación, así como los artefactos de la implementación. Se representó la manera en que quedaron distribuidos los diferentes ficheros de la multimedia y las relaciones existentes entre éstos.

### Conclusiones Generales

Con la realización de este trabajo, se presenta un nuevo material de apoyo que le brinda a los estudiantes una nueva vía de estudio para las métricas de software. Durante el desarrollo de la aplicación, se fue venciendo cada uno de los objetivos trazados en el inicio de la investigación como son:

- ✓ Estudio del arte relacionado a todo lo concerniente a aplicaciones existentes en el mundo, en Cuba y en la Universidad de Ciencias Informáticas, que dieran solución a la problemática planteada.
- ✓ Se realizó la recopilación y análisis de toda la información necesaria para confeccionar el contenido de la multimedia.
- ✓ Se realizó un estudio de las herramientas que podrían ser utilizadas en el desarrollo de la aplicación con tecnología multimedia, así como de las metodologías y herramientas de desarrollo de software más utilizadas a escala mundial.
- ✓ Se realizó el análisis y diseño de todo el producto utilizando el Proceso Unificado de Desarrollo de Software (RUP) con el lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA-L) como extensión de UML.
- ✓ Se desarrolló con éxito la Multimedia Interactiva Métricas de Software, cumpliendo con los requisitos funcionales establecidos por el cliente.

Todos estos objetivos vencidos dieron como resultado final el cumplimiento del objetivo general “Implementar una multimedia interactiva que apoye la enseñanza aprendizaje de métricas de software”, obteniendo un producto económicamente factible que cuenta con una interfaz amigable, de fácil uso y que centraliza toda la información existente acerca del curso optativo de Métricas de Software, permitiendo de esta forma una mejor vía de aprendizaje a los estudiantes y las personas interesadas en este tema.

## **Recomendaciones**

Se puede plantear que el software implementado con tecnología multimedia cumple con los objetivos planteados al inicio del trabajo, pero sin embargo se recomienda:

- ✓ Enriquecer el contenido de cada uno de los temas que conforman el producto.
- ✓ Utilizar el software propuesto como material de apoyo del curso de métricas de software.
- ✓ Extender el producto a otras universidades como material de apoyo para la enseñanza-aprendizaje de métricas de software.
- ✓ Agregarle nuevas funcionalidades al producto que posibiliten una mayor interactividad.
- ✓ Construir objetos de aprendizaje para enriquecer el contenido de la multimedia.
- ✓ Construir medios que mejoren el contenido de la multimedia.

### Referencias Bibliográficas

- Abran, A. 2008.** SOFTWARE ACQUISITION GOLD PRACTICE.METRICS-BASED SCHEDULING. *The Data & Analysis Center for Software*. 2008.
- Actionscript, Curso. 2009.** Cursos Actionsript. [En línea] 2009. [Citado el: 02 de 03 de 2009.] <http://www.killersites.com.ar/cursos-actionscript.htm..>
- Adelaide Bianchini. 2007.** Depto. de Computación y Tecnología, Universidad Simon Bolivar. Conceptos y definiciones de hipertexto. [En línea] 2007. [Citado el: 15 de 1 de 2009.] <http://www ldc.usb.ve/~abianc/hipertexto.html>.
- Alvarez, J. C. G. 2007.** "Controles y Métricas Técnicas del Software". 2007.
- Bianchini, A. 2007.** Hipertexto. [En línea] 2007. [Citado el: 15 de 01 de 2009.] <http://www ldc.usb.ve/~abianc/hipertexto.html..>
- Brito, Dayami Rodriguez 2008.** BASES DEL MÉTODO DE ESTIMACIÓN UCI V1.0. Ciudad de la Habana: UCI, 2008.
- BRUNO, Leighton. 2005.** Una propuesta Metodológica para el Diseño de Interfaces y Mapas de Navegación en Aplicaciones Hipermediales. [En línea] 2005. [http://www.inf.udec.cl/revista/edicion4/paper\\_metodologia.htm...](http://www.inf.udec.cl/revista/edicion4/paper_metodologia.htm...)
- Clic, Aula. 2006.** Aula Clic. [En línea] Marzo de 2006. [Citado el: 20 de 01 de 2009.] <http://www.aulaclic.es/index.html>.
- Calero, Dra. Coral. 2006.** Conceptos básicos, definición y formalización. s.l. : Universidad de Castilla-La Mancha, 2006. pág. 13.
- Dellamea, Esther. 2006.** Informatica I. [En línea] 2 de junio de 2006. [Citado el: 10 de noviembre de 2008.] <http://proftecnologia.blogspot.com/>.
- Diaz, Carlos. 1994.** LA TECNOLOGIA MULTIMEDIA: Una Nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones.,. [En línea] 1994. [Citado el: 11 de noviembre de 2008.] <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm>.
- FEBLES, Ailyn. 2001.** Case Corporativo para el proceso de control de cambios. Tesis presentada en opción al título de Máster en Informática Aplicada. 2001.
- FEBLES, Ailyn. 2006.** UCI, Calidad del software. Ciudad de la Habana : s.n., 2006.
- IEEE. 2004.** SWEBOOK (Guide to the Software Engineering, Body of Knowledge). 2004.

- Javo. 2008.** Córdoba software factory. [En línea] 16 de Septiembre de 2008. <http://cbasqa.wordpress.com/2008/09/16/the-chaos-report/>.
- Karel López Duardo, Darlyn Milán Aguilar. Junio, 2007..** SOFTWARE EDUCATIVO PARA LA ASIGNATURA CONTABILIDAD Y FINANZAS EN LA UCI. Ciudad de la Habana : UCI, Junio, 2007.
- Logroño. 2008.** Adrformacion.com. Adrformacion.com. [En línea] 2008. [Citado el: 15 de 11 de 2008.] <http://www.adrformacion.com/>.
- Marquèz, Pere 2004.** El software educativo. s.l. : Universidad Autónoma de Barcelona, 2004. pág. 16.
- MENJIVAR, E. 2007.** [En línea] 2007. [Citado el: 15 de noviembre de 2008.] <http://www.maestrosdelweb.com/editorial/multime/>.
- Muñoz, Mariela Cepero. 2007.** Monografias. [En línea] 2007. <http://www.monografias.com/trabajos55/proceso-de-desarrollo-software/proceso-de-desarrollo-software.shtml>.
- Pérez, Henry Ernesto Bermúdez. 2007.** Multimedia Historia Universal Volumen I. Ciudad de la Habana : UCI, 2007.
- PÉREZ, Isabel. 2002.** Métricas para el control de proyectos de software. 2002.
- Sánchez, Rafael Torregrosa.** Calidad Conceptos y Generalidades. Valencia : Consorcio Hospital General Universitario. pág. 48.
- SAUER, S. 2004.** Extending UML for Modeling of Multimedia Applications. [En línea] 2004. <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>.
- Software Tech, DoD Software Technology Challenges.** Measurement : Practical Software and Systems Measurement (PSM) . [En línea] [Citado el: 11 de 03 de 2009.] <https://www.thedacs.com/databases/url/key/4902>.
- Walsh, Norman. 2006.** Una Introducción a XML . [En línea] 2006. [Citado el: 02 de 02 de 2009.] [http://74.125.93.100/translate\\_c?hl=es&sl=en&tl=es&u=http://www.xml.com/pub/a/98/10/guide0.html%3Fpage%3D1&usg=ALkJrhiRX7FdhwwHTa2ujakbD3nPo2YLuQ](http://74.125.93.100/translate_c?hl=es&sl=en&tl=es&u=http://www.xml.com/pub/a/98/10/guide0.html%3Fpage%3D1&usg=ALkJrhiRX7FdhwwHTa2ujakbD3nPo2YLuQ).

## **Bibliografía**

**IEEE. 1990.** *IEEE Standard Glossary of Software Engineering Terminology.*

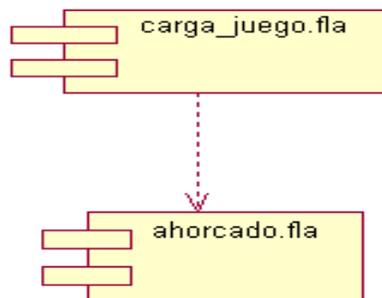
**ISO/IEC. 1991.** *Norma internacional ISO/IEC-9126..*

**IEEE. 2004.** *SWEBOK (Guide to the Software Engineering, Body of Knowledge).*

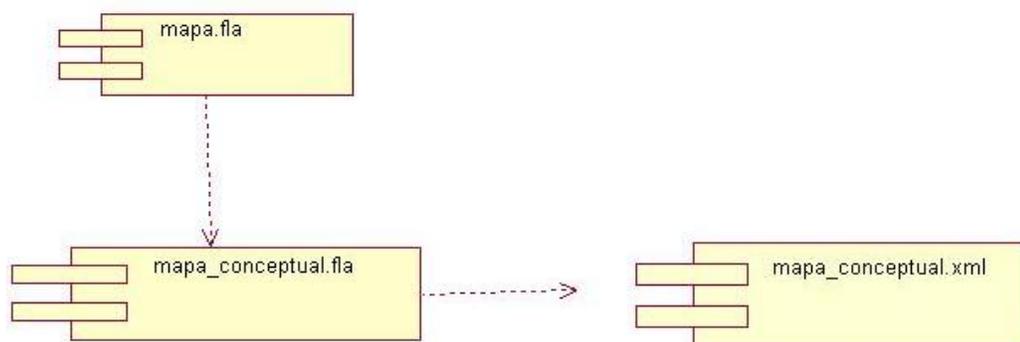
**IEEE, ACM. 2004.** *SEEK. Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering. 2004.*

**PRESSMAN, Roger. 2002.** *Ingeniería del Software: Un enfoque práctico. 2002.*

## Anexos



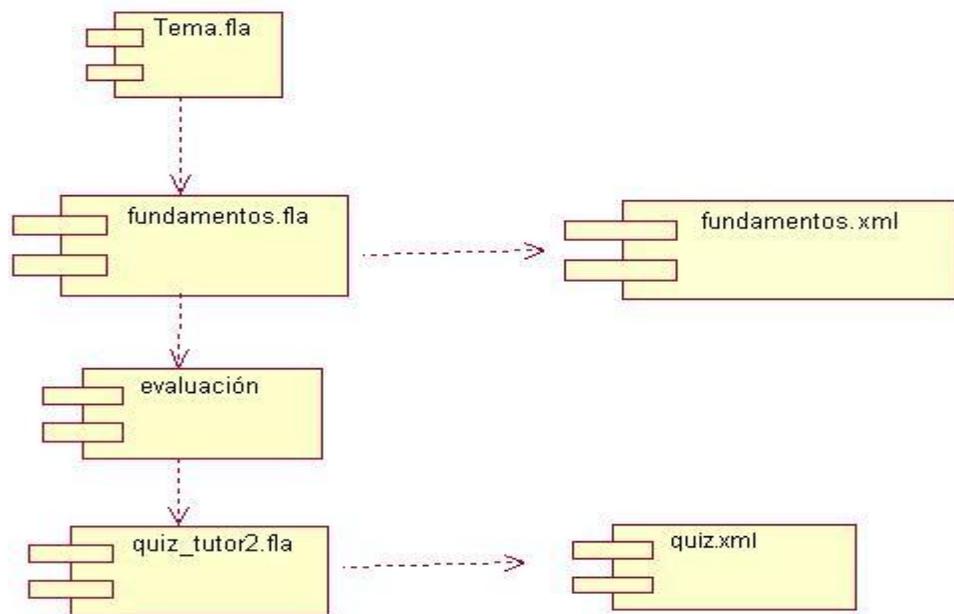
### Anexo 1: Paquete Juego



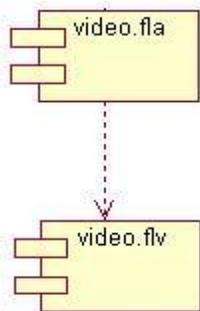
### Anexo 2: Paquete Mapa



### Anexo 3: Paquete Presentación



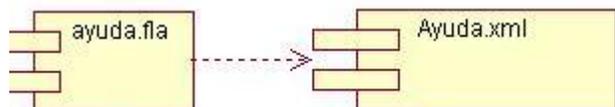
**Anexo 4: Paquete Tema**



**Anexo 5: Paquete Video**



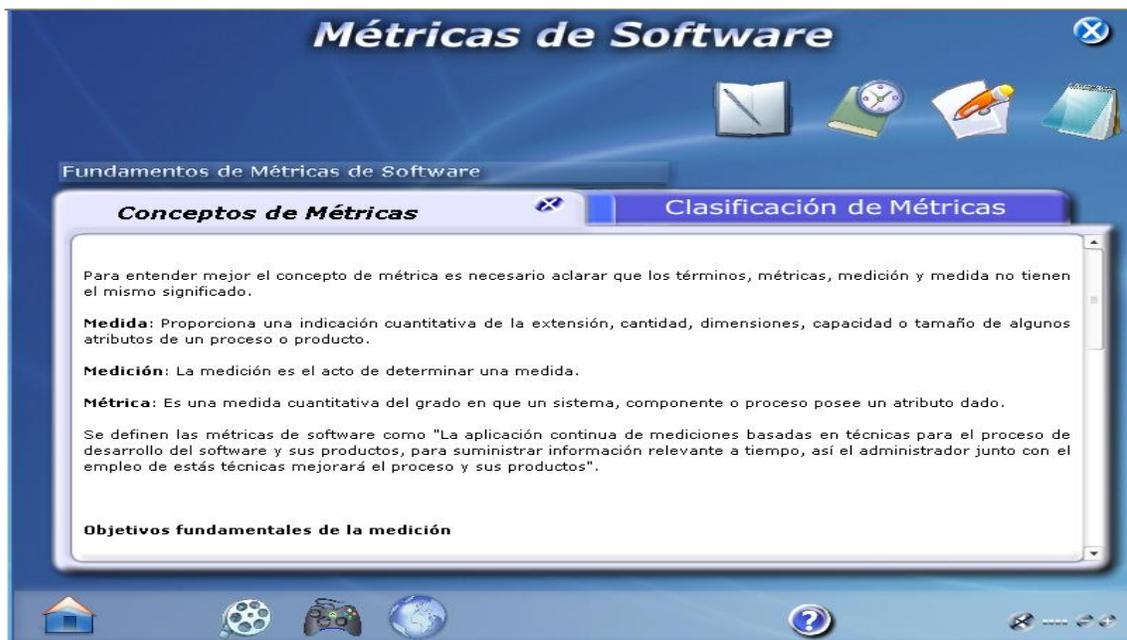
**Anexo 6: Paquete Glosario**



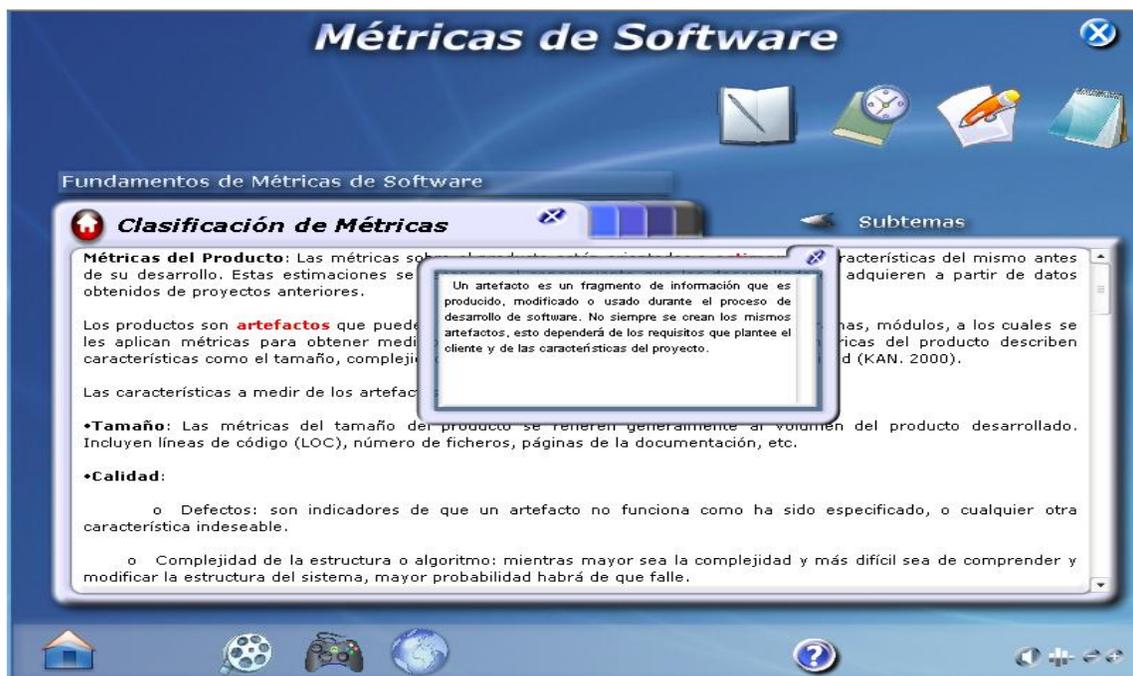
**Anexo 7: Paquete Ayuda**



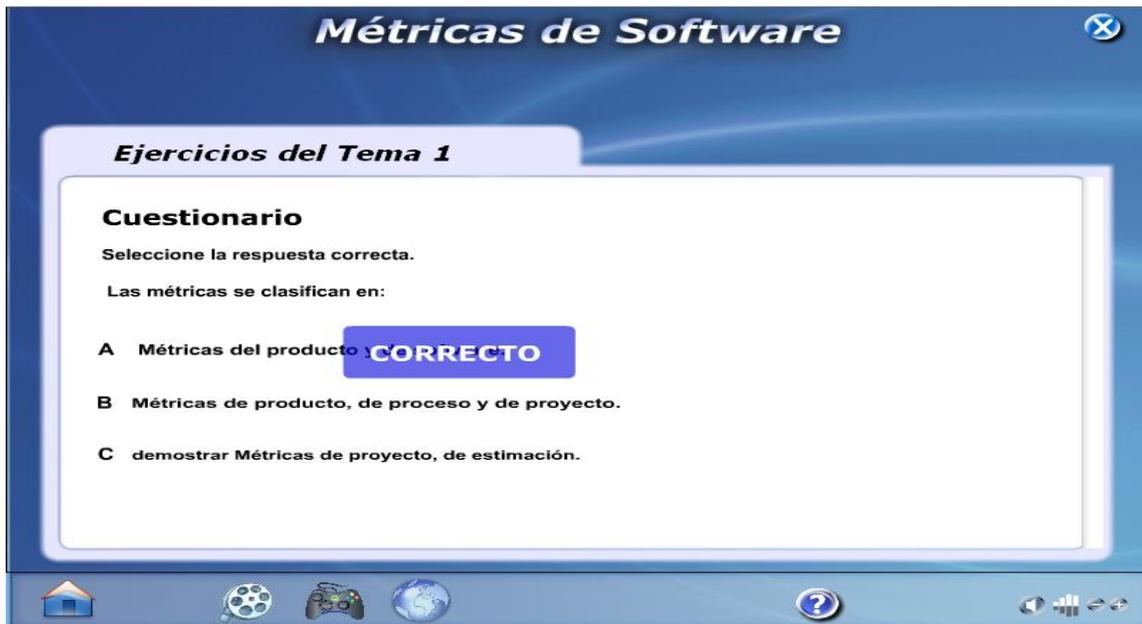
**Anexo 8: Pantalla principal de la multimedia**



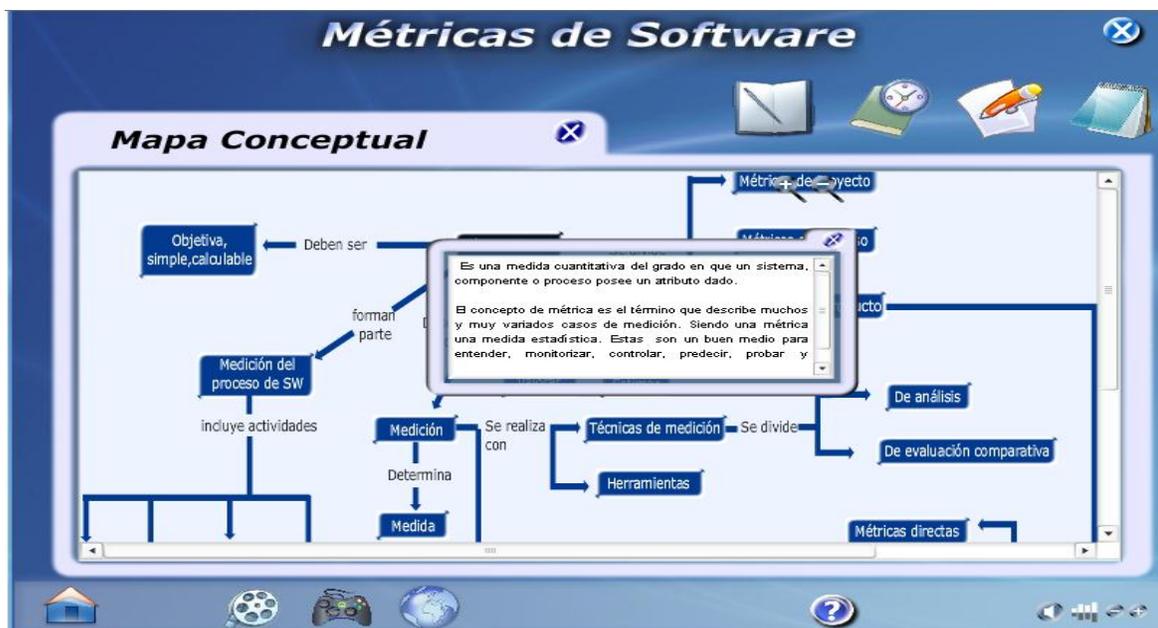
Anexo 9: Pantalla tema



Anexo 10: Pantalla mostrar palabras calientes



Anexo 7: Pantalla ejercicios



Anexo 8: Pantalla mapa conceptual con ventana flotante



Anexo 9: Pantalla juego



Anexo 10 : Pantalla Video

## **Glosario de Términos**

**Multimedia:** Es un sistema que utiliza más de un medio de comunicación al mismo tiempo en la presentación de la información, como texto, imagen, animación, vídeo y sonido

**Hipertexto:** Formato que se le aplica a un texto, en el cual se representan palabras claves (en la mayoría de los casos subrayadas o con otros colores) las cuales dan acceso a una información determinada.

**Pantalla:** Es la agrupación visual de elementos de medias contenidas en una vista determinada.

**Navegación:** Movimiento del usuario entre los objetos o segmentos de presentación de los medios, así como por su interior, para encontrar un objeto, un asunto determinado o un elemento de información específico.

**Diagrama:** Representación gráfica en la que se muestran las relaciones entre las diferentes partes de un conjunto o sistema.

**Scroll:** Barra de desplazamiento que se utiliza para movernos a través de una misma página.

**Usuario:** Un individuo u organización que utiliza el sistema operativo para ejecutar una función específica. (NC ISO/IEC 12207: 2005)

**Sistema:** Una combinación integrada, que consiste en uno o más procesos, hardware, software, facilidades y personas, que proporciona una capacidad de satisfacer una necesidad o un objetivo declarado. (NC ISO/IEC 12207: 2005)

**Proceso:** Un conjunto de actividades interrelacionadas que transforma las entradas en salidas. (NC ISO/IEC 12207: 2005)

**Producto de software:** El conjunto de programas de computación, procedimientos, y posiblemente documentación y datos asociados. (NC ISO/IEC 12207: 2005)

**Calidad de software:** “La calidad es la suma de todos aquellos aspectos o características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades, expresadas o implícitas” (ISO 8402)

**Métricas de calidad.** El concepto de métrica es el término que describe muchos y muy variados casos de medición. Siendo una métrica una medida estadística (no cuantitativa como en otras disciplinas ejemplo física) que se aplica a todos los aspectos de calidad de software, los cuales deben ser medidos desde diferentes puntos de vista como el análisis, construcción, funcional, documentación, métodos, proceso, usuario, entre otros. (NTP-ISO/IEC 12207:2003)

**Atributo:** (1) Una característica de un elemento, por ejemplo, su color, tamaño o tipo. (IEEE.Std610.12-1990) (2) Una propiedad física o abstracta mensurable de una entidad. . (IEEE Std 1061-1998)

**Eficiencia:** El grado en que un sistema o componente designado desempeña sus funciones con un mínimo consumo de recursos. (IEEE Std 610.12-1990)

**Etiqueta:** (1) Un nombre o identificador asignado a una instrucción de programa para permitir que otras instrucciones hagan referencia a esa. (2) Uno o más caracteres, dentro o adjunto a un conjunto de datos, que permiten identificar o describir los datos. (IEEE Std 610.12-1990)

**Indicador:** Un dispositivo o variable que puede ajustarse a un estado basado en los resultados de un proceso o la aparición de una determinada condición. (IEEE Std 610.12-1990)

**Interactivo:** Relativo a un sistema o modo de operación en el que a cada entrada dada por el usuario provoca una respuesta ó acción por el sistema. Contrasta con: procesamiento en lote. (IEEE Std 610.12-1990).

**Ingeniería:** La aplicación de una sistemática, disciplinada y cuantificables metodología para estructuras, máquinas, productos, sistemas, ó procesos. (IEEE Std 610.12-1990)

**Interfaz:** (1) Un límite compartido a través del cual la información se transmite. (2) Un componente de hardware o software que conecta dos o más componentes con el propósito de transmitir información de uno a otro. (3) Para conectar dos o más componentes con el propósito de transmitir información de uno a otro. (IEEE Std 610.12-1990)

**Métrica:** Una medida cuantitativa del grado en que un sistema, componente o proceso posee un determinado atributo. (IEEE Std 610.12-1990)

**Medida:** Una manera de evaluar o determinar el valor por comparación a una norma. . (IEEE Std 1061-1998)

**Medición:** El acto o proceso de asignación de un número o una categoría a la entidad para describir un atributo de esa entidad. (IEEE Std 1061-1998)

**Métricas predictiva:** Una métricas aplicadas durante el desarrollo que se utiliza para predecir los valores de un factor de calidad de software. . (IEEE Std 1061-1998)

**Métrica de proceso:** Una métrica utilizada para medir las características de los métodos, técnicas y herramientas empleadas en el desarrollo, implementación y mantenimiento del sistema de software. (IEEE Std 1061-1998)

**Métrica de producto:** Un métrica utilizada para medir las características de cualquier producto intermedio o final del proceso de desarrollo de software. . (IEEE Std 1061-1998)

**Métrica de calidad de software:** Una función cuyas entradas son los datos y cuya única salida es un valor numérico que puede interpretarse como el grado en que el software posee un atributo que afecta a su calidad. (IEEE Std 1061-1998)

**Norma de medición:** Una norma que describe las características de la evaluación de un proceso o producto. (IEEE Std 610.12-1990).

**Producto de software:** (1) El conjunto de programas de computación, procedimientos, y posiblemente documentación y datos asociados. (NC ISO/IEC 12207: 2005) (2) Un conjunto completo de programas de ordenador, procedimientos, y la documentación y los datos. (IEEE Std 1028-1997)(3) Uno o más programas de ordenador, junto con cualquier elemento que lo acompañe que no sea electrónico, ni mecánico tales como la documentación y las hojas de trabajo, entregados en virtud de una única denominación para el uso de otros (IEEE Std 1063-2001)

**Procedimiento:** serie de pasos ordenada que sigue un usuario para hacer una o más tareas (IEEE Std 1063-2001)

**Requisito funcional:** Un requisito que especifique una función que un sistema o componente del sistema debe ser capaz de realizar. Contrasta con: requisito de diseño, requisito de aplicación, la interfaz requisito; requisito de desempeño; exigencia física. (IEEE Std 610.12-1990)