

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



**Título: Multimedia Interactiva Gestión de la Calidad de Software**

**Autora: Anabel González Rivera**

**Tutores: MSc. Yamilis Fernández Pérez**

**Ing. Miguel Ángel del Pino Zincke**

## 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. Se gradúa de MSc en Informática Aplicada en 1995. Imparte docencia en universidades desde 1992. Ha desarrollado trabajos con Universidades extranjeras en Brasil, Bolivia y Canadá. Es la jefa de Departamento Docente Central de Ingeniería y Gestión de Software de la UCI desde su fundación.

Tutor: Ing. Miguel Ángel del Pino Zincke

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

## PENSAMIENTO

“La calidad empieza en la gente no en las cosas”

Philip Crosby

## AGRADECIMIENTOS

A **Fidel** y la **Revolución**: Por ser el eslabón principal en esta cadena de logros alcanzados.

A mi madre **Rosa María**: Por apoyarme en todo momento y estar ahí, cada vez que la necesite.

A mi padre **Hermes Víctor**: Por enseñarme desde muy pequeña como hacer las cosas yo sola.

A mi hermana **Sandra**: Por estar siempre pendiente, aún sin saber lo importante de esto para mí.

A mi novio **Reinel**: Por su comprensión, dedicación, cariño y por ser mi guía en todo momento.

A mis Tutores **Yamilis Y Miguel Angel**: Por apoyarme tanto y contribuir a realizar mis metas.

A mi **Tata, Mima, Mamita, Mami y Migyel**, por quererme tanto y estar tan pendientes de mí.

A mi suegro **Leonel**: Por ser como un padre para mí, y hacerme entender, que siempre se puede dar algo mejor.

A **Yany**: Gracias por ser esa persona que siempre necesite, al estar lejos de mi casa.

A **Yadith**: Gracias por dedicarme tanto en tan poco tiempo.

A **Yadira, Freeman, Carlos y Loco**: Gracias por ser tan buenos amigos y ayudarme con una lágrima, más que con una sonrisa.

A **Aritol, Dianilla, Daili y Yeni**: Gracias por ser tan buenas compañeras y lograr que mi último año se haga inolvidable.

A los que estuvieron pendientes y los que me dedicaron un minuto de su tiempo cada vez que los necesite, muchas gracias.

## DEDICATORIA

A **Marisol González Hernández** y **Anastasio Rivera Lara**, dos personas que hoy no están presentes, pero desearon mucho verme realizar este sueño.

## RESUMEN

En la actualidad, la humanidad se encuentra viviendo la era del conocimiento, la cual nos ha brindado un conjunto de facilidades que tienen como principal objetivo mejorar la calidad de vida de las personas en su espacio. Estas facilidades son las llamadas “Tecnologías de la Información y las comunicaciones” (TICs) a través de las cuales, gracias a su surgimiento y evolución, es que podemos hoy hablar de Internet e Informática.

El proceso de enseñanza viene avanzando paralelamente a estas tecnologías, y es por ésto que las metodologías y los medios de enseñanza son cada vez más modernos. El aprendizaje a través de técnicas multimedia ha impactado en el gran mundo de la red de redes, por ser éste una vía rápida, atractiva e interactiva. La universidad de las Ciencias Informáticas (UCI) ha mostrado interés en incorporar estas técnicas multimedia a su proceso de enseñanza, logrando así una capacitación más atractiva en materias opcionales de la carrera. Este trabajo de diploma propone una multimedia interactiva que apoye la capacitación de los estudiantes vinculados al proceso de enseñanza-aprendizaje de Gestión de Calidad de Software en la Universidad. Para la realización de la multimedia, primeramente se hizo un estudio detallado de las diferentes herramientas existentes para la creación de aplicaciones multimedia y se seleccionó la herramienta Macromedia Flash 8 con su lenguaje ActionScript para la implementación. También se utilizó la metodología RUP y el lenguaje de modelado UML, con su extensión OMMMA-L para aplicaciones multimedia y XML como lenguaje para gestionar y agrupar los datos en volúmenes compactos de información. La multimedia interactiva Gestión de la Calidad de Software se puede convertir en un interesante material de apoyo a la enseñanza de Gestión de la Calidad de Software en la UCI, ya que esta materia solo se imparte como un curso optativo; y no es suficiente con esto para que esté al alcance del conocimiento de todos los estudiantes.

Palabras claves: Calidad del software, gestión, multimedia, multimedia interactiva, metodología, herramientas.

## INDICE

Resumen.....	6
Introducción.....	11
Capítulo 1: Fundamentación Teórica.....	17
Introducción.....	17
1.1 Fundamentos de SWEBOK.....	17
1.1.2 Fundamentos de Calidad de Software según SWEEBOK.....	19
1.1.2.1 Valores y Costos de Calidad.....	20
1.1.2.2 Modelos y Características de Calidad.....	21
1.1.2.3 Mejora de Calidad.....	21
1.1.3 Proceso de Gestión de la Calidad de Software.....	22
1.1.3.1 Aseguramiento de la Calidad de Software (SQA).....	22
1.1.4 Consideraciones Prácticas.....	23
1.1.4.1 Técnicas de Gestión de la Calidad de Software.....	24
1.1.4.2 Caracterización de Defectos.....	24
1.1.4.3 Medición de calidad del software.....	24
1.2 Enseñanza de Gestión de la Calidad en la actualidad.....	25
1.3 Software Educativo.....	26
1.4 Multimedia. Definiciones Generales.....	27
1.4.1 Definición.....	27
1.4.2 Clasificación de Multimedia.....	29
1.4.3 Aportes al proceso de enseñanza aprendizaje.....	29
1.5 Tendencias tecnológicas actuales.....	30
1.5.1 Herramientas para la creación de Aplicaciones Multimedia.....	31
1.5.2 Herramienta a utilizar.....	32
1.5.3 Metodologías a considerar para el desarrollo del software.....	33

1.5.5 Fundamentación de la metodología a usar.....	34
1.5.4 Lenguaje de modelado.....	34
1.5.6 Lenguajes usados en la multimedia.....	36
Conclusiones.....	36
Capítulo 2: Descripción de la solución.....	37
Introducción.....	37
2.1 Identificación de la audiencia.....	37
2.2 Especificación del contenido.....	37
2.2.1 Mapa Conceptual.....	38
2.3 Descripción de Modelo de Dominio.....	40
2.3.1 Diagrama de clases de Modelo de dominio.....	40
2.3.2 Análisis de los conceptos de dominio.....	40
2.6 Diagrama de Navegación de OMMMA-L.....	41
2.6.1 Modelo de Navegación general.....	41
2.6.1 Sub Modelos de navegación por temas.....	42
2.7 Descripción de la funcionalidad.....	43
2.7.1 Requisitos Funcionales.....	43
2.7.2 Requisitos no Funcionales.....	44
2.7.2.1 Requisitos de apariencia.....	44
2.7.2.2 Requisitos de usabilidad.....	45
2.7.2.3 Requisitos de software.....	45
2.7.2.4 Requisitos de hardware.....	46
2.7.2.5 Requisitos de diseño e implementación.....	46
2.7.2.6 Requisitos de soporte.....	46
2.8 Modelos de casos de uso del sistema.....	46



2.8.1 Determinación y justificación de los actores del sistema .....	47
2.8.2 Diagrama de caso de uso.....	47
2.8.3 Breve explicación de los casos de uso del sistema .....	48
Conclusiones .....	51
Capítulo 3: Solución de la descripción propuesta .....	52
Introducción .....	52
3.1 Análisis de la arquitectura de la información utilizada .....	52
3.1.1 Principios y normas de diseño.....	52
3.1.2 Estándares para el diseño de la interfaz de usuario .....	53
3.1.3 Estándares de codificación.....	53
3.2 Descripción de archivos XML.....	55
3.3 Diagramas de Presentación del modelo de Análisis.....	58
3.4 Diagrama de Componentes .....	61
Conclusiones .....	61
Conclusiones Generales.....	62
Recomendaciones.....	63
Referencias .....	64
BIBLIOGRAFÍA .....	67
Anexos .....	68
Glosario de términos .....	73
Glosario de términos de la multimedia.....	75

## ÍNDICE DE FIGURAS

Figura 1: Indicadores a mejorar a nivel UCI.....	13
Figura 2: Temas de Calidad de Software según SWEBOK.....	18
Figura 3: Puntos de contacto de SWEBOK y SEEK .....	19
Figura 3: Países donde se consultaron las universidades .....	25
Figura 4: Diagrama de clase de dominio .....	40
Figura 5: Diagrama de navegación general .....	42
Figura 6: Diagrama de navegación temas .....	42
Figura 7: Diagrama de casos de uso .....	47
Figura 8: Diagrama de presentación Pantalla Presentación.....	58
Figura 9 Diagrama de presentación Pantalla Temas .....	59
Figura 10: Diagrama de presentación Pantalla Ejercicios .....	59
Figura 11: Diagrama de presentación Pantalla Glosario .....	60
Figura 12: Diagrama de presentación Pantalla Temas .....	60
Figura 13: Diagrama de componentes ajustado por paquetes .....	61
Anexo 1: Paquete Temas.....	68
Anexo 2: Paquete Mapa.....	68
Anexo 3: Paquete Juego.....	68
Anexo 4: Paquete Video.....	69
Anexo 5: Paquete Glosario.....	69
Anexo 6: Paquete Ayuda.....	69
Anexo 7: Paquete Presentación.....	69
Anexo 8: Pantalla principal de la multimedia.....	70
Anexo 9: Pantalla Glosario de la multimedia.....	70
Anexo 10: Pantalla Juego de la multimedia.....	71
Anexo 11: Pantalla Mapa de la multimedia.....	71
Anexo 12: Pantalla Video de la multimedia.....	72
Anexo 13: Pantalla Ayuda de la multimedia.....	72

## INTRODUCCIÓN

Durante los años noventa, cuando aparece el Internet, es que comienza ese impulso por la industria del software en el mundo. Todo parece girar en torno a una nueva idea, a una nueva forma de desarrollo. Un nuevo intento que trajo consigo nuevas necesidades, nuevas oportunidades y nuevos retos.

Al abrirse paso la tecnología y con el surgimiento de las tecnologías de la información y las comunicaciones (TIC), se ha comenzado una nueva era de desarrollo tecnológico, y la evolución ahora avanza a pasos agigantados. Las TICs han dado una configuración moderna a la sociedad actual, un cambio no solo a la informática y su tecnología adjunta, sino también a todos los medios de comunicación en general.

Estas tecnologías tienen como objetivo mejorar la calidad de vida de las personas y satisfacer las necesidades que tiene el hombre consigo mismo y con su entorno. Vienen provocando nuevas transformaciones en la cultura, incidiendo en todos los aspectos de la vida y ya se han hecho casi imprescindibles para el continuo avance y desarrollo de la humanidad.

Gracias a la existencia de los software, hoy en día es mucho más fácil el desempeño de cualquier rol en el área de trabajo. La producción de software se ha convertido en algo incontrolable. Todos necesitan software potentes, fiables y cuanto antes. ¿Qué sucede con esto? Se ha trabajado más en entregar primero, que en entregar completo y es por eso que hoy en el mundo la producción de software está pasando por un mal momento.

En el año 2008, según el estudio de la compañía norteamericana Standish Group<sup>1</sup> conocido como “The Chaos Report”, se obtuvieron los siguientes resultados. (Rubinstein, 2007)

Del total de proyectos evaluados: El 16% son completados con el alcance esperado, en el tiempo planificado y dentro del presupuesto asignado y el 53% de los proyectos son completados con menor alcance, y/o sobrecosto y/o fuera de término, mientras que el 31% de los proyectos 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% de los proyectos son cancelados antes de terminar.

Dentro de los principales factores que provocan el fracaso de los proyectos se encuentran:

- Requerimientos incompletos 13.1 %
- Falta de involucramiento del usuario 12.4 %

---

<sup>1</sup> Standish Group se formó en 1985, con una visión de futuro. Formada por un grupo de profesionales con años de experiencia práctica en la evaluación de riesgos, costos, el rendimiento y el valor de Tecnología de la Información (IT).

- Falta de recursos 10.6 %
- Expectativas no realistas 9.9 %
- Expectativas realistas 9.3 %
- Falta de apoyo de la Gerencia 8.7 %
- Requerimientos cambiantes 8.1 %

Estos datos fundamentan los problemas por los que pasa la industria del software con respecto al completamiento de los proyectos, significando ésto un atraso en la economía de cada país.

Cuba también se ve afectada por problemas en la industria del software. Según (Febles, 2003), se destacan como problemas más importantes de la industria cubana del software (ICSW) los siguientes:

- Pobre planificación y organización del trabajo.
- Escaso desarrollo de proyectos de investigación y desarrollo.
- No se aplican estándares de calidad.
- Pobre formación y superación continua de los recursos humanos.
- Poca aplicación de técnicas de comercialización.

Cuba se ha puesto como meta la creación de centros que produzcan software con el objetivo de ser usados en beneficio del país ,ya sea para uso interno o para exportarlos a otros países y fomentar la economía. Debido a todos los factores anteriormente mencionados que provocan el fracaso de los proyectos, Cuba también se ve afectada y los suyos se retrasan al igual que en otros países.

La Universidad de las Ciencias Informáticas (UCI) es una de las instituciones más grandes y con más condiciones tecnológicas que existe en el país para suplementar y enriquecer nuestros conocimientos, ya sea en la rama de la Informática o en otras. La UCI además se ha convertido en un centro de producción de profesionales calificados y dispuestos a la realización de software que aporten tanto a su formación personal como al país. (Dirección de Calidad, 2008)

La UCI no está exenta de estos problemas relacionados con la industria del software. La mayoría de los proyectos productivos está haciendo sus software con mucha prisa, se trabaja más para obtener un producto rápido, que un producto fiable. Esto hace que el software obtenido no tenga la calidad requerida y por tanto no satisfaga las necesidades del cliente.

Según un diagnóstico realizado por los integrantes la Dirección de Calidad en el año 2008 a 14 áreas planificadas y a 151 proyectos de 153 planificados, se obtuvieron los siguientes resultados (Dirección de Calidad, 2008):

La UCI cuenta actualmente con un total de 60 proyectos que presentan inconsistencias. Un 49% de los proyectos no tienen Plan de Desarrollo de Software, un 21% no tiene un cronograma que indique el avance de su proyecto, o sea, si han estado cumpliendo las tareas que se trazaron. Aquí se demuestra la poca evaluación de adherencia que existe. Un 30 % no tiene un Documento Visión y un 63% de los proyectos no hace una entrega total del expediente del proyecto. La evaluación demostró que se debe mejorar en la calidad de los proyectos.

Según el “Chaos Report”, existen 10 indicadores que inciden en el éxito o el fracaso de los proyectos. La Dirección de Calidad propone 5 de estos indicadores a mejorar a nivel UCI.

A continuación los resultados de la evaluación de los proyectos con los 5 factores que más influyen en la Universidad. (Ver Fig. #1)

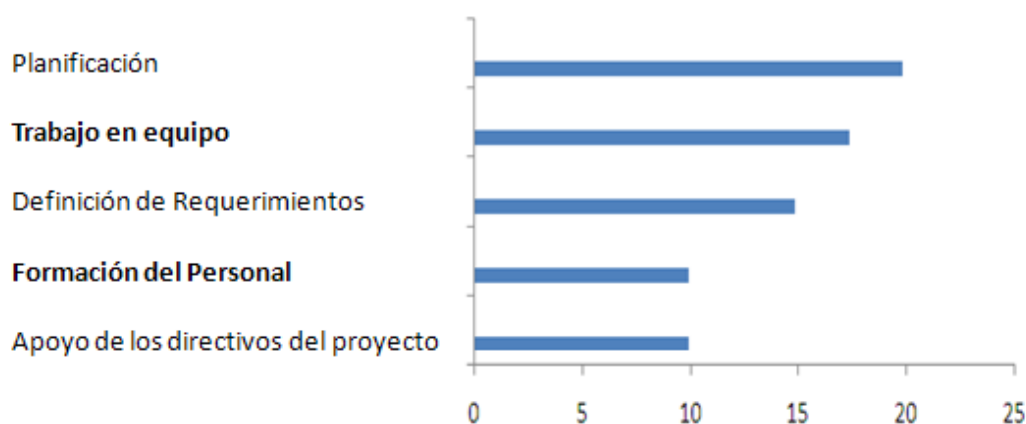


Figura 1: Indicadores a mejorar a nivel UCI

Teniendo en cuenta los resultados de la evaluación, se observa que la formación del personal y el trabajo en equipo, son dos de los indicadores a mejorar para lograr el éxito de los proyectos.

La Universidad se encuentra actualmente bajo un proceso de mejora hasta el nivel de madurez 2 del CMMI, donde se necesita personal capacitado en el tema de gestión de la calidad.

A raíz de una encuesta realizada a 6 proyectos de 4 de los polos productivos de la Universidad por los consultores del SIE Center<sup>2</sup> de Monterrey de México en Noviembre de 2008, con el objetivo de brindar el proceso de mejora que se está llevando a cabo, se dieron a conocer los siguientes resultados:

En el área de proceso de Aseguramiento de la Calidad de Proceso y Producto (PPQA), se detectaron como problemas parciales la evaluación objetiva de Procesos, la evaluación objetiva de Productos y servicios de trabajo, el establecimiento de los registros, la planeación del proceso, la supervisión y control del proceso y por último la revisión del estado con la alta gerencia. Además se detectó como problema principal que no se realiza en ningún proyecto el entrenamiento del personal y no se evalúa

<sup>2</sup> Centro de excelencia de la industria del software.

objetivamente adherencia. Esto quiere decir que aunque los proyectos tengan bien definidos sus procesos, no se evalúa si en verdad están cumpliendo los pasos que establecieron para esa definición.

Además de los resultados obvios de esta evaluación, el autor desea agregar las tres dimensiones críticas por las que pasa el proceso de software, ellas son:

- Procedimientos y métodos definiendo la relación de tareas
- Herramientas y Equipamiento
- **Personas con habilidades, entrenamiento y motivación.**

Finalmente se plantea la necesidad de un personal capacitado en el tema de Gestión de Calidad de Software para lograr que:

- Existan menos inconsistencias en los proyectos
- La UCI pueda llegar al nivel 2 en el proceso de mejora que se está llevando a cabo
- El software realizado funcione como establecen las dimensiones de su proceso.

Como la formación de Pregrado de la Universidad no prepara al personal en el tema Gestión de la Calidad de Software, debido a que solo se toman algunos conceptos generales en asignaturas que se comienzan a impartir no desde el comienzo de la carrera, y además como la temática no forma parte de una asignatura obligatoria del plan de estudios de la carrera; se plantea como **problema científico**: ¿Cómo sustentar la inclusión de buenas prácticas en el proceso de enseñanza de Gestión de la Calidad de Software, a través de las TICs?

A raíz de esto, el **objeto de estudio** será los medios para el aprendizaje de Gestión de la Calidad de Software en la UCI y como **campo de acción**, el proceso de desarrollo de una multimedia que apoye la enseñanza de Gestión de la Calidad de Software.

A partir del problema científico planteado, se ha determinado como **Objetivo de la investigación**: Implementar una multimedia interactiva como medio de enseñanza en el proceso de Gestión de la Calidad en la Universidad de Ciencias Informáticas.

Para cumplir el objetivo anteriormente planteado, se dará cumplimiento a las siguientes **Tareas Investigativas**:

- Recopilar y analizar de toda la información necesaria sobre gestión de calidad de software para elaborar el contenido del producto a desarrollar.
- Confeccionar un mapa conceptual que contenga los conceptos necesarios para la comprensión del tema.

- 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, a partir de las metodologías existentes para el desarrollo de aplicaciones con tecnología multimedia.
- Realizar el análisis y diseño de 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.

Para lograr el objetivo propuesto y dar respuesta al problema científico planteado anteriormente, se aplicaron los siguientes **métodos teóricos**: El método analítico-sintético, el cual se utiliza para desglosar todo el contenido vinculado al tema de Gestión de Calidad de Software, que es muy amplio. Revisando ordenadamente cada uno de los conceptos y fundamentaciones acerca del tema, descubrir todos los elementos que tienen relación de alguna forma con éste, e integrar todo esto en una unidad nueva, a partir del análisis previo realizado. El método histórico lógico para estudiar toda la evolución histórica de la situación problemática planteada y darle solución, y por último el método de modelación para crear una abstracción de la realidad y realizar el análisis y diseño de una solución técnica para el problema científico planteado. Como **método empírico** se utilizó el método entrevista, ya que se realizaron consultas a expertos en el tema de Gestión de Calidad de Software; de forma no estructurada ya que es más flexible y puede variar de una persona a otra.

El presente trabajo está constituido por tres capítulos que conforman el desarrollo de la investigación.

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

Se realiza la fundamentación teórica basada en el análisis de dos libros (SWEBOK Y SEEK) que definen el QUÉ y el CÓMO del tema a desarrollar. Además se hace una explicación del estado del arte y de las herramientas y metodologías a usar para el desarrollo del producto.

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

Este capítulo está vinculado al análisis de la multimedia interactiva Gestión de Calidad de Software. Se construye un modelo de dominio; se definen los requisitos funcionales y no funcionales que debe cumplir y se modela el sistema para la realización de los casos de uso correspondientes. Además se define el guión de la aplicación, donde va a estar implícito el contenido de la misma.

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

Este capítulo va dirigido a la definición del modelo de análisis del sistema y modelos de implementación, incluyendo todos los diagramas necesarios. Se hace una descripción de los archivos XML y también se describen las normas de diseño que va a tener la aplicación.



## CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

### Introducción

El conocimiento de la calidad de software se hace cada vez más necesario para todo Ingeniero Informático, ya que el término es usado cada vez más de forma trascendental en compañías y empresas asociadas al mundo de la competencia. En este primer capítulo se hace un estudio de un libro que describe todo el contenido necesario que debe conocer, comprender y analizar cada Ingeniero Informático: La guía del cuerpo del conocimiento de la Ingeniería de Software, (SWEBOK), la cual surgió en el año 2004 por IEEE Computer Society<sup>3</sup>. Después de haber realizado el estudio, se haya puntos clave entre este libro y la currícula de la disciplina de Ingeniería y Gestión de Software (SEEK) propuesta de ACM y IEEE, que debe regir el plan de estudios de la carrera de Ingeniería de Software.

En este capítulo se plasman además los resultados de una investigación realizada en los planes de estudio de la carrera Ingeniería Informática, en varias universidades de América del sur, América del norte y Europa, de cómo planifican la Gestión de la Calidad de Software como asignatura, así como también se hace una búsqueda de la existencia de software que enseñen Gestión de la Calidad de Software y qué características tienen los mismos.

Se analizan las herramientas, metodologías y lenguajes existentes para la realización de multimedia, eligiéndose finalmente lo que se va a usar en todo el proceso de desarrollo de la misma, justificando dicha elección, con las ventajas que puede brindar a la aplicación.

### 1.1 Fundamentos de SWEBOK

La descripción del área del conocimiento Calidad de Software según el SWEBOK, cubre tres subáreas. La primera describe los fundamentos de la calidad de software, tales como: cultura y éticas de Ingeniería de Software, el valor y los costos de calidad, los modelos y características de calidad, y la mejora de la calidad. La segunda cubre los procesos de gestión y aseguramiento de la calidad del software. Los asuntos aquí son verificación y validación, auditorías y revisiones y garantías de la calidad de software. La tercera y final subárea describe las consideraciones prácticas relacionadas con la calidad de software: requisitos, caracterización de defectos, técnicas de administración, y medidas de calidad de software. Finalmente contiene un listado de todos los estándares relacionados con el área del conocimiento en general. (Ver Fig. #2)

---

<sup>3</sup> Institute of Electrical and Electronics

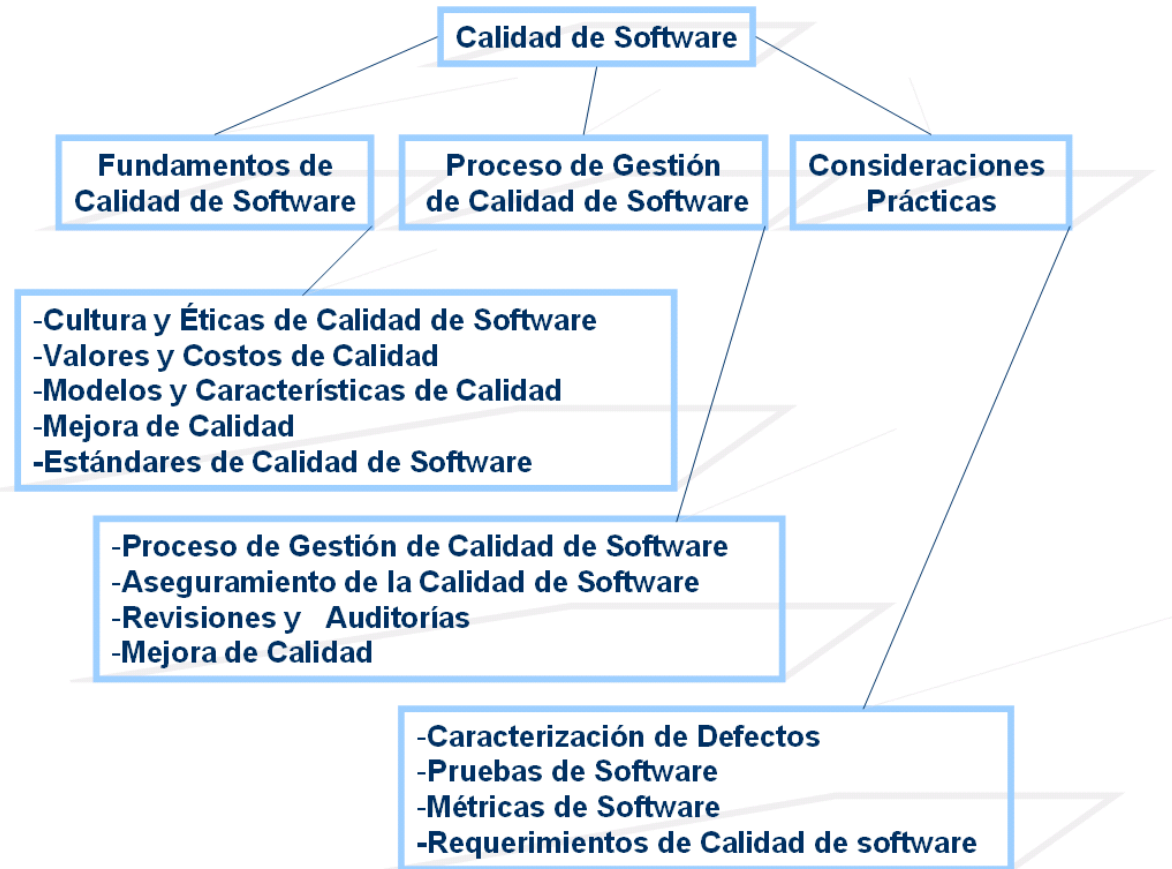


Figura 2: Temas de Calidad de Software según SWEBOOK

Después de analizar los fundamentos del SWEBOOK en el área del conocimiento de Gestión de la Calidad y los puntos clave que plantea la Currícula de la disciplina de Ingeniería y Gestión de Software, se hallaron los puntos de contacto entre ambos libros para definir el contenido a estudiar en este primer capítulo. Para analizar estos puntos se tuvo en cuenta el nivel de conocimiento que se le da a cada temática, según la *Taxonomía de Bloom*. Los puntos que coincidieron en ambos libros se muestran en la siguiente figura.

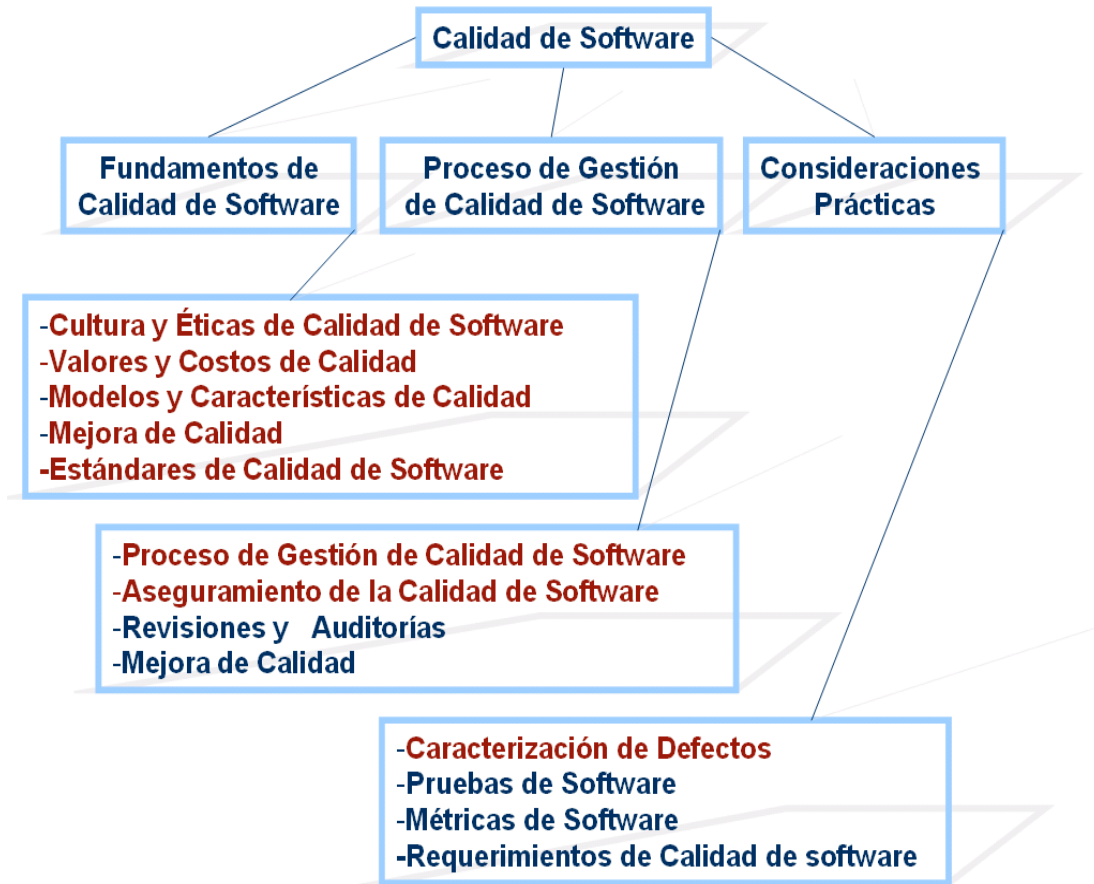


Figura 3: Puntos de contacto de SWEBOOK y SEEK

### 1.1.2 Fundamentos de Calidad de Software según SWEEBOK

Durante muchos años varias organizaciones y autores han definido el término **calidad** de maneras diferentes. Calidad fue definido como *la conformidad a los requisitos del usuario* según Phil Crosby. IBM acuña la frase “calidad impulsada por el mercado”, que se basa en lograr la satisfacción total del cliente. Además la calidad ha sido definida por la ISO9001-00, como el grado en el cual un grupo de características inherentes que satisfacen los requisitos.

Es importante la comprensión de estos conceptos por el ingeniero de software, así como las características de la calidad para el software que se desarrolla o se mantiene. Los ingenieros de software deben tener un compromiso con la calidad de software como parte de su cultura.

La IEEE y ACM han desarrollado un código de éticas y de prácticas profesionales, basado en 8 principios para ayudar a los ingenieros de Software a reforzar actitudes relacionadas a la calidad y a la independencia de su trabajo. Estos principios son:

- **Público** - los ingenieros de software actuarán constantemente con interés público.

- **Cliente y Empleado** - los ingenieros de software actuarán de forma que en los intereses de su cliente y empleado esté el constante interés público.
- **Producto** – los ingenieros de software se asegurarán de que sus productos y modificaciones relacionadas, cumplan con los más altos estándares profesionales posibles.
- **Juicio** - los ingenieros de software mantendrán integridad e independencia en su juicio profesional.
- **Administración** - los administradores y los líderes de la ingeniería de software suscribirán y promoverán un acercamiento ético a la administración del desarrollo y del mantenimiento del software.
- **Profesión** - los ingenieros de software avanzarán la integridad y la reputación de la constante profesión con el interés público.
- **Colegas** - los ingenieros de software serán justos y apoyarán a sus colegas.
- **Uno mismo** - los ingenieros de software participarán en la formación continua con respecto a la práctica de su profesión y promoverán un acercamiento ético a la práctica de la profesión.

### 1.1.2.1 Valores y Costos de Calidad

Las características de la calidad pueden requerirse o no pueden ser requeridas, a un mayor o menor grado, y pueden hacerse compensaciones entre ellas. Se entiende por **Costos de Calidad** al dinero destinado para obtener la calidad requerida. La calidad requerida no se consigue por casualidad ni accidentalmente, sino que todo debe ser planeado en actividades, medido y garantizado. El costo de calidad puede segmentarse en el costo de prevención, el costo de evaluación, el costo de falla interno, y el costo de falla externo.

**Costo de prevención:** Es el costo de todas las actividades y medidas que tienden a prevenir fallas o problemas.

**Costo de evaluación:** Es el costo de los ensayos, inspecciones o auditorías para evaluar si la calidad especificada es lograda y mantenida.

**Fallas internas:** Son los costos resultantes de ineficiencias o de un producto que no logra cumplir con los requisitos de calidad, previo a su entrega al cliente.

**Fallas externas:** Son los costos resultantes de un producto o servicio que logra cumplir con los requisitos de calidad, luego de ser adquirido por el cliente.

Una motivación latente tras un proyecto de software es el deseo de crear un software que tiene valor, y este valor puede o no ser cuantificado como un costo.

El cliente no tiene en mente el problema de calidad y de su costo, él solamente piensa en el costo del software y las funcionalidades de éste. Además puede tener expectativas respecto a la calidad.

Es esencial comprometer al cliente con el proceso de toma de decisión y de hacerlo consciente, tanto de los costos de calidad como de sus beneficios.

### 1.1.2.2 Modelos y Características de Calidad

La terminología para las características de calidad del software difiere de una taxonomía (o modelo de calidad de software) a otra, cada modelo quizás tenga un número diferente de niveles jerárquicos y un número total diferente de características.

La gestión de la calidad de software y la calidad de proceso de la ingeniería de software guardan relación directa con la calidad del producto software. Los modelos y los criterios que evalúan las capacidades de las organizaciones de software son esencialmente la organización del proyecto y las consideraciones de gestión. No es posible distinguir completamente la calidad del proceso de la calidad del producto. La calidad de proceso afecta a las características de calidad de los productos software, que a su vez repercuten en la calidad-en-el-uso tal y como es percibido por el cliente.

Entre los estándares más importantes, se destaca la ISO 9001-00 que es la más general, así como sus directrices para su aplicación al software [ISO90003-04]. Se destaca también la ISO 9126, la cual define características y medidas útiles para determinar la calidad de producto de software y como estándar industrial CMMI, el cual propone una mejora para los procesos, TickIT y otros.

Existió un debate acerca de: ¿Qué estándar usar para asegurar la calidad? ¿ISO o CMMI? El resultado es que los dos se complementan; y que tener una certificación ISO 9001 puede ser de gran ayuda para los logros de los más altos niveles de madurez de CMMI.

### 1.1.2.3 Mejora de Calidad

La calidad de los productos software puede ser mejorada mediante un proceso iterativo de mejora continua que requiere control de dirección, coordinación, y retroalimentación de varios procesos concurrentes:

- Los procesos de ciclo de vida de software
- El proceso de detección, eliminación y prevención de errores/defectos
- El proceso de mejora de calidad.

La construcción en calidad, mediante la prevención y detección temprana de errores, mejora continua y enfoque en el cliente, son conceptos basados en el trabajo de expertos en calidad, los cuales han afirmado que la calidad de un producto está directamente conectada con la calidad del proceso empleado para crearlo.

Apoyando la gestión de calidad se sustenta el proceso y la evaluación del producto, entonces se desarrolla un programa de mejora identificando acciones detalladas y proyectos de mejora para ser gestionados en un plazo de tiempo factible.

### 1.1.3 Proceso de Gestión de la Calidad de Software

La gestión de la calidad del software se aplica a todas las perspectivas de los procesos del software, productos y recursos.

La calidad de software ha de ser planificada. La planificación para la calidad de software comprende: definición del producto en términos de sus características de calidad y planificación de los procesos para alcanzar el producto requerido.

Calidad del producto de software: Lo primero es determinar el objetivo real del software. Para ello es importante tener en mente que los requisitos del cliente son lo primero, e incluyen los requisitos de calidad, no solo los requisitos funcionales. El estándar ISO9126 define, para 2 de sus 3 modelos de calidad, las características y subcaracterísticas de calidad relacionadas, y las medidas útiles para la evaluación de la calidad del producto del software.

Los procesos de gestión de la calidad del software deben abordar cuán bien los productos de software van a satisfacer o satisfacen los requisitos de los involucrados, y los clientes ofrecen la calidad de software que se necesita para satisfacer los requisitos del software. Los proyectos de Aseguramiento de la Calidad de Software (SQA) consisten en tareas y técnicas para indicar cómo los planes del software son implementados y cómo los productos intermedios y finales definen los requisitos del software. Los resultados de estas tareas se llevan a reportes para gestionarlos antes de empezar la acción correctiva.

El SQM puede usarse tanto para los productos intermedios, como para el producto final. Algunos de los procesos SQM específicos se definen en el estándar IEEE 12207:

- Proceso de Aseguramiento de la calidad
- Verificación y Validación
- Revisión
- Auditorias

#### 1.1.3.1 Aseguramiento de la Calidad de Software (SQA)

El proceso de SQA procura mantener la calidad a lo largo de todo el desarrollo y mantenimiento del producto mediante la ejecución de una variedad de actividades en cada etapa, que pueden permitir identificación temprana de problemas, un rasgo casi inevitable de cualquier actividad compleja. El papel

del proceso SQA es asegurar que procesos planeados, sean apropiados y posteriormente implementados de acuerdo con la planificación.

Estos procesos ofrecen la garantía de que el software, los productos y los demás procesos en todo el ciclo de vida del proyecto, se ajusten a los requisitos planeados y además realizan un sistema de actividades para proporcionar una confianza de que se está incorporando la calidad al desarrollo del software. El papel del aseguramiento de la calidad del software con respecto a los procesos es asegurarse de que sean apropiados y ejecutados según el plan.

El plan de aseguramiento de la calidad del software define los medios que serán utilizados para asegurar que el software desarrollado para un producto específico satisfaga los requisitos del usuario y que sea de la mejor calidad posible dentro de los apremios del proyecto. El plan de SQA debe ajustarse al plan de gestión de la configuración e identificar documentos, estándares, prácticas y las convenciones que gobiernan el proyecto y cómo serán ellos comprobados. Debe a la vez identificar los documentos, estándares, prácticas y convenciones que gobiernan un proyecto y cómo serán comprobados y supervisados para asegurar eficiencia y conformidad. El Plan SQA trata las actividades de aseguramiento de calidad de software de cualquier actividad descrita en los planes del software y puede contener criterios de aceptación, así como actividades de gestión y de reporte que son críticas en la calidad de software.

#### 1.1.4 Consideraciones Prácticas

Varios factores influyen en la planificación, gestión y selección de las actividades y técnicas de la gestión de calidad del software. La información de estos factores, influye en cómo los procesos de gestión de calidad del software son organizados y documentados; cómo las actividades específicas de SQM son seleccionadas, qué recursos son necesitados y cuáles impondrán límites en los esfuerzos. Estos factores son:

- El dominio del sistema en el cual se instalará el software
- Requisitos de software y sistema
- Los componentes comerciales y estándares que serán utilizados por el sistema
- Los estándares de Ingeniería de Software aplicados.
- Los métodos y herramientas de software a ser utilizados para el desarrollo y mantenimiento y para la evaluación y mejora de la calidad
- El presupuesto, personal, organización del proyecto, planes y cronogramas de todos los procesos
- Los usuarios y el uso del software
- Nivel de Integridad del sistema.

#### 1.1.4.1 Técnicas de Gestión de la Calidad de Software.

Las técnicas de gestión de la calidad de software pueden ser categorizadas en varias formas: estáticas, intensivo a personas, analíticas y dinámicas.

**Estáticas:** Incluyen el revisar la documentación del proyecto y software y otras informaciones acerca de los productos de software, sin ejecutarlos.

**Intensivo a personas:** Incluye revisiones y auditorías, una reunión formal o informal, pero frecuentemente 2 o más personas están relacionadas. Deben prepararse con tiempo.

**Analíticas:** Ejemplos de esta técnica son los análisis de complejidad, de flujo de control y algorítmicos. Cada análisis tiene un objetivo específico y no todos los tipos se aplican a todo proyecto.

**Dinámicas:** Estas son técnicas de testeo, pero además tenemos simulación, el chequeo de módulos y la ejecución simbólica.

#### 1.1.4.2 Caracterización de Defectos

Los procesos SQM encuentran defectos. Caracterizar estos defectos conduce a un entendimiento del producto, facilita correcciones al proceso o al producto, e informa al gestor del proyecto o al cliente del estado del proceso o el producto. Existen muchas taxonomías defectuosas y, mientras se ha intentado obtener un consenso en una taxonomía de defectos o fallos, la literatura indica que hay bastantes en uso. Una probable acción resultante de las conclusiones SQM es eliminar los defectos del producto durante su inspección.

#### 1.1.4.3 Medición de calidad del software

Los modelos de calidad del software incluyen métricas para determinar el grado de calidad de cada característica alcanzada por el producto. Si estos modelos son usados correctamente, las métricas pueden ser usadas en procesos de toma de decisiones de gestión. Pueden encontrar áreas problemáticas y cuellos de botella en procesos de software, y pueden ayudar a los ingenieros de software a evaluar la calidad de su trabajo respecto de objetivos SQA y respecto a los procesos a largo plazo de mejora de calidad.

Mientras que las métricas para determinar las características de calidad y rasgos de producto pueden resultar útiles en sí mismas (por ejemplo, el número de requerimientos defectuosos o la proporción de requerimiento defectuosos), pueden aplicarse técnicas matemáticas y gráficas para ayudar en la interpretación de esas métricas.



## 1.2 Enseñanza de Gestión de la Calidad en la actualidad

En la actualidad el término calidad es muy usado en empresas maduras o inmaduras, porque es la base de la competencia. El problema es que no todos siguen los pasos para lograr una mejora del proceso. Cada rol desempeñado en los procesos del ciclo de vida del software juega un papel fundamental para que éste funcione como un todo integrado. He aquí la importancia de los conocimientos que debe tener el personal dedicado al aseguramiento de la calidad, además de conocer modelos y estándares que son los que logran que cada producto cumpla con los requisitos del cliente.

Después de analizar a varios países de América del Norte, América del Sur y Europa, los cuales se señalan a continuación, (Ver Fig. # 4), se plantea:



Figura 4: Países donde se consultaron las universidades

Muchos países se han desarrollado en cuestiones de calidad del software. Colombia por ejemplo tiene el Instituto Colombiano de Normas técnicas y certificación, donde se imparten cursos a distancia en distintas ciudades del país relacionados con esta temática, ninguno de estos cursos es gratis. Además, en términos de convenios educativos, este instituto es uno de los centros que más maestrías y cursos de Postgrados (relacionados con la calidad, específicamente el aseguramiento de la Calidad), ofrece a profesionales que se quieran superar.

En muchas Instituciones como La Universidad Politécnica de Madrid en España, la Universidad de Buenos Aires en Argentina y la Universidad Autónoma de México, la calidad se encuentra en el plan de estudios de la carrera Ingeniería Informática como una asignatura de libre elección, o adjunta a alguna otra asignatura, y no como un curso obligatorio del plan. Debido a esto, se evidencian algunos de los

problemas actuales con los proyectos y la poca capacitación que tiene el personal vinculado al trabajo. Sin embargo, en otras se comporta de manera diferente. En la Universidad de Alagappa de India, la Universidad Tecnológica de México y la Universidad de Waterloo en Canadá, la calidad forma parte del plan de estudios como materia obligatoria, además se dan cursos y postgrados de calidad de software, principalmente de modelos de calidad, pero no son gratuitos. En la Universidad Iberoamericana de México existe la maestría en la especialidad de calidad, ya que según la institución los problemas existentes en la actualidad con la industria del software son debidos al poco conocimiento del personal. La Universidad de Carnegie Mellon cuenta con una facultad de Ciencias en la Computación, la cual es la más destacada a nivel mundial, y un Instituto de Ingeniería de Software (SEI), un auténtico líder en la normalización de los procesos del software, ya que desarrolló su línea de trabajo sobre el concepto de “madurez” de las organizaciones para producir software. Publicó el modelo de madurez de la capacidad para el desarrollo de software (CMM), que tras más de una década de existencia, ha mostrado que en muchas organizaciones ha resultado eficaz. Varios trabajadores de la Universidad han desarrollado trabajos importantes relacionados con la mejora de la calidad del software en los proyectos y la industria del software.

En la Universidad de la Ciencias Informáticas la materia de calidad de software se imparte como un curso de perfil de la formación de pregrado. En algunas asignaturas como la disciplina Ingeniería y Gestión de Software, se toman algunos conceptos y se da alguna idea de algunos modelos y estándares existentes. Estas asignaturas son impartidas por métodos tradicionales centrados en el profesor, y además se pueden encontrar en el entorno virtual de aprendizaje (EVA), en el cual está incluido todo el plan de estudios de la carrera en la Universidad. Está incluida en la preparación de Pregrado como curso opcional del perfil de Calidad de Software. En el caso del del Postgrado existe un diplomado, y se trabaja en el diseño de una maestría.

### 1.3 Software Educativo

El software educativo es empleado cada vez más en el proceso enseñanza-aprendizaje de todos los niveles de enseñanza, ya que está destinado a la enseñanza y al auto aprendizaje de cualquier materia. Con éstos se puede instruir o ilustrar alguna temática de una forma más interactiva y atractiva al usuario. Se define como software educativo cualquier programa computacional cuyas características estructurales y funciones sirvan para apoyar el proceso de enseñar, aprender y administrar, es decir, un material de aprendizaje especialmente diseñado para ser utilizado en una computadora en los procesos de enseñar y aprender. (Ramos, 2008)

Actualmente existen empresas que se han dado cuenta de la necesidad que hay de tener un personal capacitado cuando se habla de calidad de software, y de tener contar con un proceso maduro que forme

parte de un ciclo de mejoras constantemente. La compañía española SEIS Informática desarrolló un software para la Gestión de la calidad (QSEIS) orientado a la mejora continua. Este software tiene como uno de sus componentes la mejora de la formación del personal. Como principales características este software tiene:

- Conformidad con la normas internacionales sobre calidad
- Interfaz amigable, facilidad de uso
- Estructura modular integrada y escalamiento
- Multiempresa

Dicho software es una aplicación Cliente/Servidor, y de forma gratuita solo se obtiene un pequeño DEMO

Finalmente se plantea que no se encontró ningún software que cumpla las características de la aplicación a desarrollar, y que además esta al alcance de toda la docencia, de manera gratuita.

## 1.4 Multimedia. Definiciones Generales

Con el surgimiento de las TICs el proceso enseñanza-aprendizaje ha dado un giro impresionante, capaz de llevar la enseñanza contemporánea a una enseñanza actual llena de nuevas posibilidades y nuevas técnicas. La computadora es la base de este proceso, porque sin ella, las posibilidades de avanzar aplicando estas nuevas técnicas serían prácticamente nulas. Las tecnologías multimedia surgen con el objetivo de fomentar el proceso de enseñanza-aprendizaje, y por sus ventajas hoy en día son muy usadas para apoyar la comprensión de alguna temática que acompañada de un conjunto de imágenes audio y video hace que el aprendizaje sea más atractivo e interactivo entre un usuario y la máquina.

### 1.4.1 Definición

La **multimedia** se inicia al final de la década de los 90, cuando la gran compañía Apple lanzó al mercado su nuevo producto: la computadora Macintosh, debido a que ésta, tenía una gran capacidad de reproducción de sonidos y diseño gráfico. Se define un concepto como la combinación de varios medios en una computadora, sean éstos textos, imágenes, audio, videos y animaciones.

En el campo de las Nuevas Tecnologías se puede acotar el concepto de multimedia al sistema que integra o combina diferentes medios: texto, imagen fija (dibujos, fotografías) sonidos (voz, música, efectos especiales) imagen en movimiento (animaciones, vídeos), a través de un único programa software. (Díaz, 2004)

Con el paso de los años las multimedia han evolucionado y han dejado de ser un conjunto de textos, adornado con imágenes y gráficos, para convertirse en toda una atracción. Es como surge la interactividad.

Desde el punto de vista del usuario, **interactividad** es la cantidad de control que éste tiene sobre los contenidos. Esta definición alude a los grados de interactividad que puede tener el producto. El más bajo de estos niveles de interactividad es el agotado y agotador recurso de teclear una y otra vez enter –lo que algunos, con indudable gracia, han dado en llamarle interactividad. La verdadera interactividad, en cambio, implica una modificación en la propia estructura del "relato" multimedia: se pasa de una presentación linear donde todo el control es del autor, a la multilinearidad donde el control es del usuario. El usuario es así él dueño de la secuencia y del tiempo dedicado al contenido.

Resumiendo un poco, la interactividad será el diálogo entre usuario-máquina, en el cual el grado de interactividad de la multimedia será el soporte para determinar su complejidad. (Multimediam, 2006)

Cuando se habla de multimedia, se hace necesario también definir el concepto de **hipertexto**, el cual ha sido definido como un enfoque para manejar y organizar la información, en el cual los datos se almacenan en una red de nodos conectados por enlaces. Los nodos contienen textos y si contienen además gráficos, imágenes, audio, animaciones y video, así como código ejecutable u otra forma de datos, se les da el nombre de **hipermedio**, es decir, una generalización de hipertexto. (Bianchini, 2000)

Los usuarios suelen confundir los términos **hipermedia** e **hipertexto**. Se debe hacer una aclaración de estos conceptos para que no surjan estas dudas. El término **hipermedia** surge de la fusión de los conceptos de **hipertexto** y **multimedia**. A diferencia del **hipertexto**, la **hipermedia** trabaja no solo con palabras, sino también con imágenes, sonidos y videos. A continuación se amplían un poco más estos conceptos por separado.

La **hipertextualidad** es una característica de cierta producción cultural contemporánea, una tendencia que Internet no crea, sino que solamente incrementa. (Borges, 2008)

Finalmente, dando una definición más sencilla, es el documento que reúne imágenes, textos, audio y videos relacionados entre sí mediante vínculos. Debe estar siempre bien estructurado, sino provocaría una desorientación al usuario y ésto resultaría fatal a los resultados del impacto de la multimedia.

La **hipermedia** puede considerarse como una forma especial de multimedia interactiva que emplea estructuras de navegación más complejas que aumentan el control del usuario sobre el flujo de la información. (Sabatini, 2008)

## 1.4.2 Clasificación de Multimedia

Atendiendo a su estructura, los materiales didácticos multimedia se pueden clasificar en programas tutoriales, de ejercitación, simuladores, bases de datos, constructores, programas herramienta..., presentando diversas concepciones sobre el aprendizaje y permitiendo en algunos casos la modificación de sus contenidos y la creación de nuevas actividades de aprendizaje por parte de los profesores y los estudiantes. (Graells, 2009)

**Programas de ejercitación.** Se limitan a proponer ejercicios autocorrectivos de refuerzo sin proporcionar explicaciones conceptuales previas.

Su estructura puede ser: lineal: la secuencia en la que se presentan las actividades es única o totalmente aleatoria. Ramificada: la secuencia depende de los aciertos de los usuarios, o tipo entorno: proporciona a los alumnos herramientas de búsqueda y de proceso de la información para que construyan la respuesta a las preguntas del programa.

**Programas tutoriales.** Presentan unos contenidos y proponen ejercicios autocorrectivos al respecto. Si utilizan técnicas de Inteligencia Artificial para personalizar la tutorización según las características de cada estudiante, se denominan tutoriales expertos.

**Bases de datos.** Presentan datos organizados en un entorno estático mediante unos criterios que facilitan su exploración y consulta selectiva para resolver problemas, analizar y relacionar datos, comprobar hipótesis, extraer conclusiones, etc.

## 1.4.3 Aportes al proceso de enseñanza aprendizaje

Con el transcurso de los años, un tema de preocupación para algunos centros educacionales es que los estudiantes adquieran los conocimientos y desarrollen habilidades. Para ésto es necesario tratar de brindarle al estudiante servicios de apoyo que le faciliten su proceso de formación. El software educacional surge por estas razones; y ha sido de gran utilidad en la formación de cada estudiante.

Las multimedia interactivas educativas son un eslabón básico en esta cadena de software si se definen sus ventajas de una manera clara, sobre todo porque ha quedado claro con el transcurso del tiempo que el ser humano es capaz de adquirir mejor unas ideas que estén plasmadas en una imagen, o darle

continuidad a una trayectoria de ideas, que estar escuchando una grabación o estar viendo un video , donde no haya ningún tipo de interacción.

Las ventajas de usar las multimedia es que el estudiante proporciona nuevos estímulos y propicia un aprendizaje a través del ensayo y error. La multimedia no solo colabora con el estudiante, también colabora con el docente para que por medio de ésta pueda facilitar su trabajo creando software educativos, videos, entre otros, para así fortalecer el Sistema Educativo. (Jiménez, 2008)

## 1.5 Tendencias tecnológicas actuales

Debido al gran auge que ha tomado la formación de profesionales utilizando los cursos distribuidos a través de Internet, hoy la tarea de crear herramientas que fomenten este proceso se hace cada vez más imprescindible. Los Entornos Ambientales Virtuales de Aprendizaje (EVA) no surgen con el objetivo de trasladar un aula física a un aula virtual, sino de lograr un ambiente donde prevalezca una interacción mutua entre estudiantes-profesores y estudiantes-estudiantes, y que este proceso esté apoyado por un conjunto de herramientas que ayuden a crecerlo. Dichas herramientas pueden estar destinadas tanto a la creación de materiales multimedia, como a editores de páginas web o software de comunicación.

Actualmente existen varias herramientas, algunas de forma gratuita y otras no, a la disposición de profesores y educadores para la creación de entornos de enseñanza-aprendizaje a través de Internet. A veces se hace un poco complejo determinar la herramienta más adecuada para lograr los objetivos específicos, es por ésto que haciendo un estudio detallado de cada una de ellas, esta actividad se haría menos engorrosa. A continuación se presenta algunas de las clasificaciones de estas herramientas: (Crosetti, 2000) (Díaz, 2005)

- Herramientas para la gestión y administración académica: gestionan la matrícula e inscripción de los alumnos en los cursos, proporcionan información académica como horarios, fechas de exámenes, notas, planes de estudios, expedición de certificados, concretar reuniones, tutorías, etc.
- Herramientas para la creación de materiales de aprendizaje multimedia. En este grupo se engloban todos aquellos programas que son utilizados para la creación de los contenidos de aprendizaje como: los editores de páginas web, las herramientas de autor (posibilitan la realización de aplicaciones multimedia interactivas las cuales pueden ejecutarse en Internet a través de 'plugins') o las que facilitan la creación de ejercicios de autoevaluación, simulaciones, etc.
- Herramientas integradas para la creación y distribución de cursos a través de la WWW. Desarrolladas específicamente para propósitos educativos. Se refiere a aplicaciones de Internet que facilitan la creación de entornos de enseñanza-aprendizaje integrados, dependiendo del grado de

desarrollo de la propia herramienta, los materiales de aprendizaje, las herramientas de comunicación y colaboración y la gestión.

La Universidad de las Ciencias Informáticas cuenta con un Entorno Virtual de Aprendizaje (EVA), el cual es un espacio de apoyo al proceso de formación de la Carrera de Ingeniería en Ciencias Informáticas y a los programas de postgrados que tributan a la formación del claustro de profesores de la universidad. EVA presenta dos categorías principales: Formación de Pregrado y Formación de Postgrado, las cuales no están muy enriquecidas de materiales interactivos que apoyen la capacitación de los estudiantes, sobre todo pregrado que es donde más interacción estudiantes-profesor debe existir.

### 1.5.1 Herramientas para la creación de Aplicaciones Multimedia

Hoy en día muchas instituciones utilizan métodos de enseñanza continua y cada vez son más las empresas que utilizan las nuevas tecnologías para conseguir mejores resultados en la enseñanza. Los programas de autor se han utilizado tradicionalmente para el diseño y creación de aplicaciones multimedia. Técnicamente todas las aplicaciones multimedia son muy parecidas, pues todas utilizan diferentes recursos como texto, imágenes, sonido, videos etc. Sin embargo, a nivel funcional se clasifican en 3 tipos:

- Kioscos: Son aplicaciones jerárquicas que conducen al usuario de forma interactiva por todo el contenido existente.
- Computer Based Training (CBT): Son cursos que hacen uso de multimedia para reforzar conceptos importantes, además proporcionan interactividad para permitir un aprendizaje autónomo. A diferencia de los Kioscos, en ellas existe una parte de autoevaluación.
- Presentaciones: Tienen un componente de interactividad muy bajo, su objetivo principal es dar a conocer algún tipo de producto. (Díaz, 2005)

Después de un análisis previo a las diferentes tipos de aplicaciones multimedia, se decide optar por un CBT, ya que se necesita de un alto grado de interactividad y además que exista una parte de evaluación para comprobar los conocimientos de los estudiantes.

Luego de analizar varias herramientas, que existen en la actualidad para la creación de multimedia, como son: Authorware, Director, Scala multimedia y ToolBook, descartamos a las herramientas Director y Scala Multimedia, ya que no permiten realizar todo tipo de contenidos y se usan más bien para presentaciones y resaltar efectos visuales.

Después de un análisis a las herramientas ToolBook y Authorware, se valoró que tampoco cumplen nuestras expectativas por las características que presentan:

*ToolBook*: Es una herramienta con un lenguaje de programación propio: Open Script, muy flexible, las aplicaciones creadas en ToolBook se asemejan mucho a la estructura de un libro. La herramienta tiene una dependencia total de la plataforma Windows, pero tiene como desventaja que posee un costo elevado, y que tiene un componente de interactividad muy bajo.

*Authorware*: Es una herramienta diseñada para la creación de libros, enciclopedias interactivas y otros materiales que permite la combinación de varios medios y es multiplataforma. Como desventaja tiene la dependencia de la paginación, su poca flexibilidad, y el ajuste a patrones preestablecidos (muy limitado).

### 1.5.2 Herramienta a utilizar

Para el desarrollo del producto multimedia Interactiva Gestión de la calidad, se decide optar por la herramienta Macromedia Flash, en su versión 8, que es el entorno más avanzado en el mercado. Flash 8 supera las expectativas debido a sus características y las ventajas que ofrece, ya que Flash ofrece de forma explícita una calidad final del trabajo y es multiplataforma. A continuación una serie de características que fomentan la elección de la herramienta. (AulaClic, 2006)

- Carga dinámica de imágenes (JPEG, PNG, GIF, JPG)
- Ayuda tanto para la programación (ActionScript) como para el diseño de animaciones.
- Incluye componentes ya creados que ayudan a la hora de crear animaciones.
- Interfaz gráfica amigable, sencilla de usar y con muchas herramientas.
- Biblioteca de símbolos.
- Soporte de audio MP3.
- Soporta video.
- Posee un lenguaje de programación propio, muy conocido (ActionScript).

**ActionScript**: Es el lenguaje de programación para crear scripts en flash. A través de él, se puede controlar la línea de tiempo, (grupo de fotogramas secuenciales que presenta Flash) y las propiedades de los objetos. Hace posible generar el contenido programado y la comunicación con el servidor.

ActionScript es un lenguaje de programación orientado a objetos. Tiene similitudes, por tanto, con lenguajes tales como los usados en el Microsoft Visual Basic, en el Borland Delphi, etc y, aunque evidentemente no tiene la potencia de un lenguaje puramente orientado a objetos derivado del C o del Pascal como los anteriores, cada versión se acerca más a un lenguaje de este tipo. Así, la versión 2.0 recientemente estrenada en el Flash MX 2004 es mucho más potente y mucho más "orientado a objetos" que su anterior versión 1.0. (AulaClic, 2004)



### 1.5.3 Metodologías a considerar para el desarrollo del software

Las metodologías para el desarrollo del software sirven para subir la calidad de éste, ya que controlan de forma transparente todo el proceso. En el presente trabajo se hizo un análisis de varias metodologías existentes, para elegir la más adecuada.

Para la realización de este análisis, se definieron varios criterios

1. Tamaños de los equipos.
2. Nivel de programación y flexibilidad
3. Obtención de requisitos
4. Carga de trabajo
5. Conocimiento del autor
6. Antecedentes en la colección de multimedia

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

	<b>RUP</b>	<b>XP</b>	<b>MFS</b>	<b>FDD</b>
<b>1</b>	proyectos grandes con roles designados y duración extendida.	Proyectos cortos, rotables.	Proyectos cortos, corto plazo.	Proyectos medianos, A mayor necesidad de código, mayor organización
<b>2</b>	Tiene 4 fases que se desarrollan mediante un ciclo iteraciones, cada ciclo exige uso de artefactos.	Programación rápida o extrema, funcionalidades retroalimentación continua.	Un poco rígida	Es flexible, incluye un Monitoreo permanente
<b>3</b>	Se basa en casos de uso, donde se definen los requisitos.	Se basa en historias de casos de uso, solo define detalles técnicos.	Durante la fase de Planificación, se Preparan las Especificaciones funcionales	No hace énfasis, en la obtención de requerimientos.
<b>4</b>	Se basa en la documentación. Presenta un plan de desarrollo, donde se reconocen los errores y se pueden corregir	Proceso ligero, no se asignan roles organizativos al equipo	Tiene definido un modelo de roles, con varias personas ningún rol es omitido	Se genera una documentación, depende del jefe que es el más responsable
<b>5</b>	Muy conocido, es el usado en toda la trayectoria	Poco conocido	Poco conocido	Desconocido
	si	no	no	no

Después de haber realizado este análisis, no cabe duda en la elección de la metodología RUP, pero XP también tiene características que no debemos pasar por alto.

Decir de RUP que es un proceso pesado y de XP que es un proceso ligero, sin más calificativos no aclara la función de cada uno, pero XP no es un formato libre, sino sujeto a una disciplina: Se centra estrechamente en un aspecto determinado del desarrollo de software y en un modo de proporcionar valor, y es prescriptivo sobre la forma en que debe alcanzarse este objetivo. El ámbito de RUP es mucho más amplio y de la misma profundidad, lo que explica su "tamaño" aparente. No obstante, a un micronivel del proceso, RUP permite y ofrece ocasionalmente alternativas igualmente válidas, mientras que XP no. (IBM Corporation, 2006).

### 1.5.5 Fundamentación de la metodología a usar

Después de un análisis a las metodologías que existen actualmente, se hace uso de RUP. Pues es un proceso para el desarrollo de un proyecto de software que define claramente quien, cómo, cuándo y qué debe hacerse en el mismo. Gracias a su enfoque basado en modelos, utiliza un lenguaje bien definido para tal fin, el Lenguaje Unificado de Modelado (UML). Está preparado para desarrollar grandes y complejos proyectos, unifica los mejores elementos de metodologías anteriores y es orientado a objetos. RUP proporciona una visión disciplinada para la asignación de tareas y responsabilidades en las organizaciones de desarrollo de software, además integra un conjunto de "buenas prácticas" para el desarrollo de software en un marco de procesos válido para un rango amplio de tipos de proyectos y organizaciones. Las principales buenas prácticas cubiertas son:

- Desarrollo iterativo.
- Gestión de requisitos.
- Uso de arquitecturas basadas en componentes.
- Uso de técnicas de modelado visual.
- Verificación continua de la calidad.
- Gestión y control de cambios.

### 1.5.4 Lenguaje de modelado

Modelar es diseñar las aplicaciones de software antes de la codificación. Es una parte esencial de los grandes proyectos de software y ayuda también a los proyectos pequeños.

RUP es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado (UML), constituye la metodología estándar más utilizada para el análisis, implementación y documentación de

sistemas orientados a objetos. UML está compuesto por diversos elementos gráficos que se combinan para conformar diagramas, ya que es un lenguaje y cuenta con reglas para combinar tales elementos. UML ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de las piezas que constituirán el modelo. Sin embargo, desde el punto de vista puramente tecnológico, UML tiene una gran cantidad de propiedades que han sido las que, realmente, han contribuido a hacer de UML el estándar de facto de la industria que es en realidad. Algunas de las propiedades de UML como lenguaje de modelado estándar son:

- Concurrencia: Es un lenguaje distribuido y adecuado a las necesidades de conectividad actuales y futuras.
- Ampliamente utilizado por la industria desde su adopción por OMG.
- Reemplaza a decenas de notaciones empleadas con otros lenguajes.
- Modela estructuras complejas.
- Las estructuras más importantes que soporta tienen su fundamento en las tecnologías orientadas a objetos, tales como objetos, clase, componentes y nodos.
- Emplea operaciones abstractas como guía para variaciones futuras, añadiendo variables si es necesario.
- Comportamiento del sistema: casos de uso, diagramas de secuencia y de colaboraciones, que sirven para evaluar el estado de las máquinas.

UML es un lenguaje de modelado de propósito general. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática. No pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso. (Pérez, 2006)

Existen varios lenguajes de modelados que siguen los pasos de UML, y son usados directamente para el desarrollo de aplicaciones multimedia, ya que éste anteriormente mencionado no posee ciertas características; como por ejemplo modelar la interfaz de usuario.

En el año 2001 surge OMMMA-L para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario, el cual está sustentado en cuatro vistas fundamentales: Vista Lógica, Vista de Presentación espacial, Vista de Comportamiento temporal predefinido y Vista de Control Interactivo. Cada una de estas vistas se asocia a un tipo de diagrama en particular.

Años después, en el 2007, surge el lenguaje de modelado de aplicaciones educativas ApEM-L, el cual utiliza para su representación todas las herramientas CASE que existen actualmente para la modelación de UML. Utiliza el estándar internacional OCL y no modifica la semántica del lenguaje base UML, sino que trabaja en estereotipos restrictivos. (Ciudad, 2008)

Para el desarrollo de la aplicación se decide optar por OMMMA – L, ya que actualmente se evalúa en diferentes escenarios como proyectos industriales para la especificación de servicios de información multimedia, y se investigan características adicionales de sincronía para su especificación en el lenguaje y la formalización de un modelo para la composición dentro y entre los diferentes diagramas de comportamiento.

### 1.5.6 Lenguajes usados en la multimedia

**XML:** es un Lenguaje de Etiquetado Extensible muy simple, pero estricto, el cual juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML, pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información. (W3C 2008)

## Conclusiones

En este capítulo se analizaron todos los conceptos fundamentales dados por el SWEBOK, para dar un mayor entendimiento de los aspectos más importantes del tema y seguir con la solución del problema a resolver. Se hizo un análisis del conocimiento del tema en varias universidades del mundo. Se hizo un estudio de software educativos existentes para la enseñanza de la gestión de la calidad en el mundo. Se analizaron sus características para hacer una comparación con el producto multimedia “Gestión de Calidad de Software”. Finalmente, se hizo un estudio de los lenguajes y metodologías existentes para elegir la más aceptada para el desarrollo de la aplicación.

## CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN

### Introducción

En el capítulo anterior se planteó un problema de investigación, al cual se le va a dar solución en este segundo capítulo. Primeramente se identificará la audiencia a la cual va dirigida el producto, y se dará una breve explicación de los contenidos que se pondrán en la aplicación. Se modelará un diagrama de dominio y se detallarán los conceptos que lo conforman. Se seleccionarán los requisitos funcionales y no funcionales. Por último, se realizará el diagrama de casos de uso del sistema y se hará una breve descripción de cada uno de ellos para que la audiencia pueda comprender cada uno de éstos.

### 2.1 Identificación de la audiencia

La identificación de la audiencia es un aspecto de suma importancia, pues se hace muy necesaria una correcta identificación del usuario final, para lograr superar sus expectativas.

La aplicación va dirigida a la docencia en general, principalmente el rol asegurador de cada proyecto productivo de la Universidad, el cual necesita materiales que apoyen su formación personal. La multimedia “Gestión de Calidad de Software” será una aplicación que fomentará el proceso de enseñanza-aprendizaje y ayudará a que cada proyecto tenga una documentación con las exigencias y la calidad que requiere.

### 2.2 Especificación del contenido

El contenido de la multimedia interactiva Gestión de Calidad de Software, está dividido en 4 temas:

Tema1: Fundamentos de Calidad de Software

En este tema se tocan algunas definiciones del concepto calidad, así como la evolución de la misma. Se plantea el concepto de costo y los tipos de costos existentes, así como una breve explicación de su importancia. Se introducen los términos de calidad de proceso y producto y se explica todo el proceso que se lleva a cabo en la Mejora de Calidad.

Tema2: Modelos y Estándares de Calidad de Software

En este tema se mencionan todos los modelos y estándares dados por el SWEBOK, ya sea para proceso o para producto, y se explican los considerados más importantes.

### Tema3: Proceso de Gestión de Calidad de Software

Aquí se explicará el proceso de gestión de la calidad, basándose en las cuatro partes que la conforman, se mencionarán algunas técnicas de gestión de calidad y se profundizará en otras. Por último se explicará el Aseguramiento de la Calidad del Software, basandose en conceptos, normas y Plan SQA.

### Tema4: Evaluación de Calidad de Software

Este tema estará dividido en las evaluaciones del proceso y del producto. En la evaluación del producto se abordará el tema de las pruebas de software y modelos y estándares usados para el mismo fin. En el caso de la evaluación del proceso se abordarán las auditorías y las revisiones y normas y herramientas que realizan la misma función.

#### 2.2.1 Mapa Conceptual

En el epígrafe anterior se explicaba la especificación del contenido a tratar en la multimedia. Además de los 4 temas que se pondrán, el producto contiene un mapa de conceptos que resume de una manera más simplificada el contenido del curso. Un mapa conceptual no es más que una representación de varios conceptos en forma de red que se encuentran interrelacionados entre sí. Cada nodo será un concepto diferente y los enlaces serán las relaciones entre ellos.

El mapa conceptual es un instrumento útil para la organización y la representación visual del conocimiento, cuya elaboración puede ejecutar el docente para mostrar al estudiante cómo se relacionan determinados conceptos o puede ejecutar el alumno, individualmente o en grupo, con el objeto de alcanzar una mayor comprensión de los conceptos que estudia

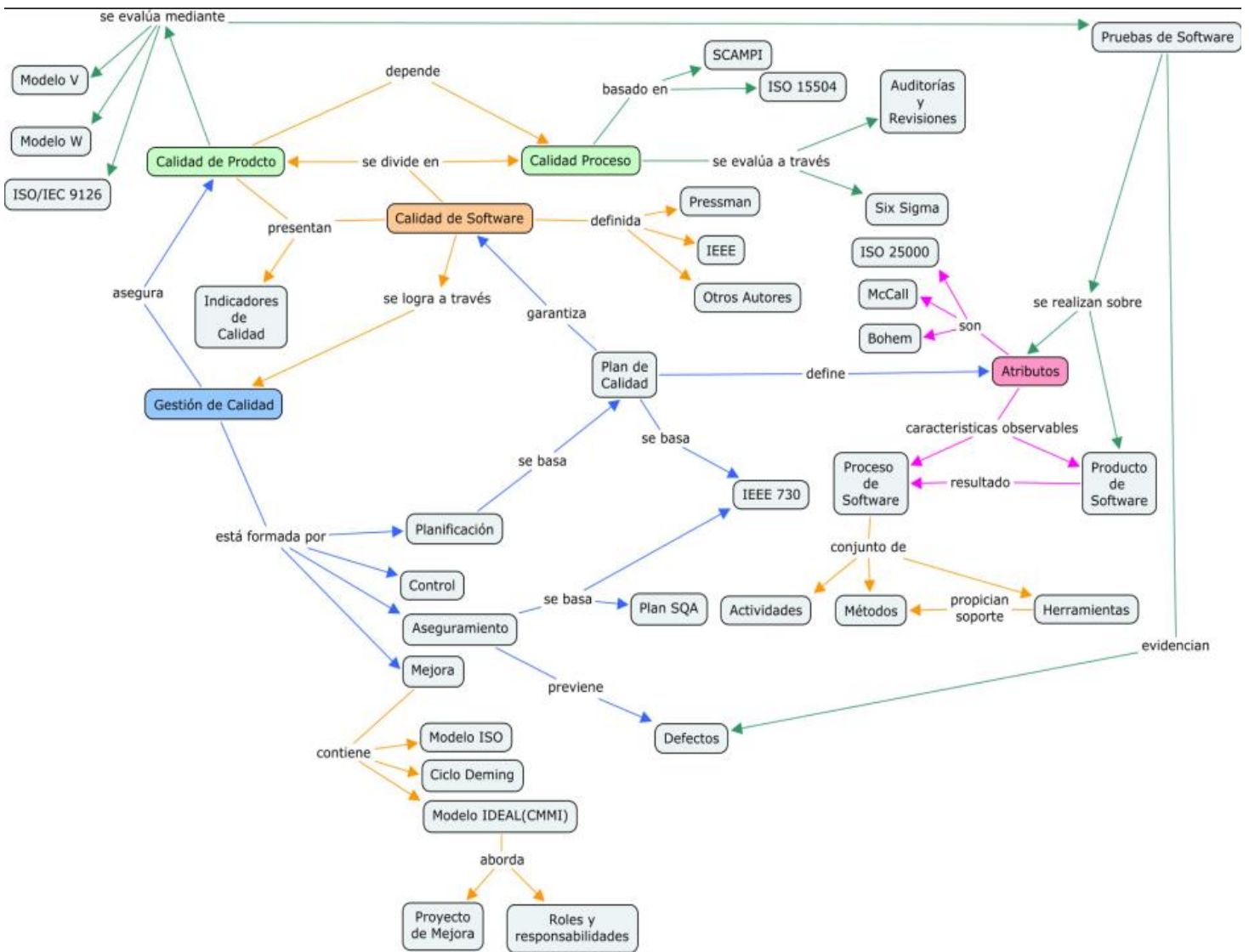


Figura 5: Mapa de Conceptos

Los mapas conceptuales se caracterizan por la jerarquización de los conceptos, ya que los conceptos más inclusivos ocupan los lugares superiores de la estructura gráfica; todo lo anterior ocurre debido a la selección de los términos que van a ser centro de atención y por el impacto visual, ya que permiten observar las relaciones entre las ideas principales de un modo sencillo y rápido. Dadas esas características, esta estrategia didáctica puede ser un instrumento eficaz para el desarrollo del pensamiento científico en los estudiantes, porque en ellos se ponen de manifiesto las características esenciales de este tipo de pensamiento, el carácter jerárquico, el carácter integrador y la multiplicidad de descripciones. (Díaz, 1997)

## 2.3 Descripción de Modelo de Dominio

Debido a la ausencia de una visión clara de los procesos de negocio, se plantea un modelo de dominio global, que dará una representación visual de los conceptos más significativos para el problema a través de las clases conceptuales del dominio del mismo. Este modelo se representa en UML con un diagrama de clases en el que se muestran los conceptos del dominio del problema y las asociaciones entre las clases conceptuales. Para la realización del mismo, se listaron todos los conceptos, se representaron en un diagrama y se agregaron las asociaciones para registrar las relaciones entre los mismos.

### 2.3.1 Diagrama de clases de Modelo de dominio

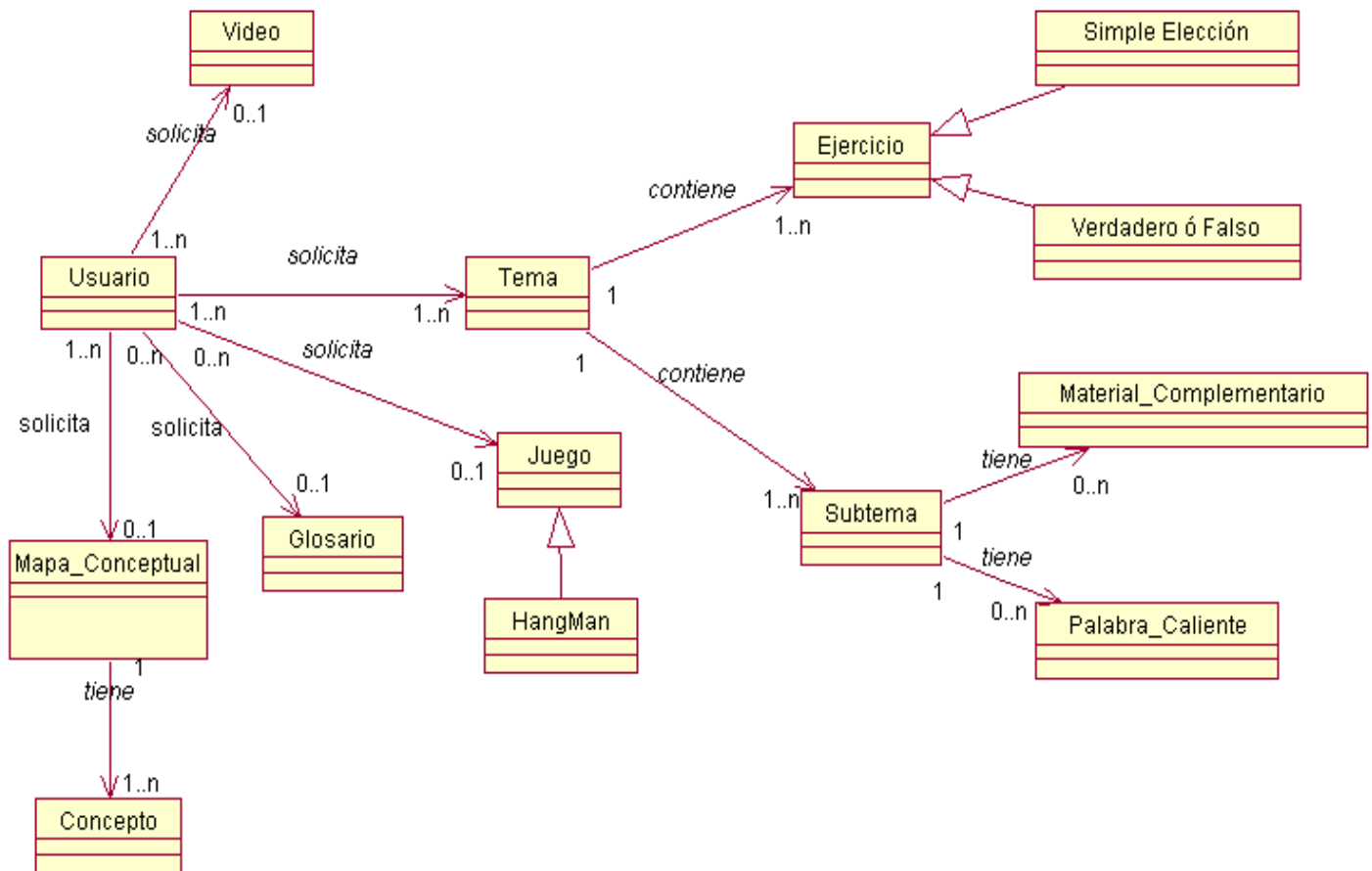


Figura 6: Diagrama de clase de dominio

### 2.3.2 Análisis de los conceptos de dominio

A continuación se realiza la descripción de los conceptos que son utilizados en el diagrama del modelo de dominio:

- ✓ **Usuario:** Toda persona que usa el sistema e interactúa con él.



- ✓ **Video:** representa una videoconferencia sobre Gestión de Calidad de Software.
- ✓ **Tema:** Partes en que se divide el contenido del curso, el cual está dividido en 4 temas fundamentales: Fundamentos Gestión de Calidad de Software, Modelos y Estándares de Calidad de Software, Proceso de Gestión de Calidad de Software y Evaluación de Calidad de Software.
- ✓ **Juego:** Representan objetos que ejercitan de una manera didáctica los conocimientos adquiridos como es el caso del juego del Ahorcado, donde el usuario intentará descubrir palabras claves pertenecientes al tema del curso.
- ✓ **Mapa Conceptual:** Mapa de conceptos sobre el contenido del curso relacionado entre sí.
- ✓ **Concepto:** Los conceptos son las estructuras mediante las cuales se identifican las características esenciales para definir las cosas, hechos o eventos que dan significado a nuestra realidad inherente y circundante.
- ✓ **Ejercicios:** Representan el conjunto de ejercicios divididos por temas para ejercitar los contenidos (3 o más ejercicios por tema).
- ✓ **Subtema:** Se le denomina subtemas a los diferentes contenidos que se tratan dentro de un tema general.
- ✓ **Material Complementario:** Representa conferencias, documentos o presentaciones; relacionadas con alguna de las temáticas que se abordará.
- ✓ **Palabra Caliente:** hipertexto que abre una ventana flotante que profundiza en el significado de alguna palabra o concepto tratado en un epígrafe.
- ✓ **Glosario:** Representa todo el conjunto de términos que pueden ser de poco conocimiento al usuario.

## 2.6 Diagrama de Navegación de OMMMA-L

El mapa de navegación es un diagrama especificado por OMMMA-L que expresa la forma en que el usuario se debe desplazar entre los diferentes pantallas de la aplicación, proporcionando información real sobre el desplazamiento dentro de la misma.

Para una mejor comprensión a continuación se muestra un diagrama de navegación general y un sub-diagrama de navegación.

### 2.6.1 Modelo de Navegación general

La aplicación da inicio con una pantalla de presentación del curso. La misma da paso a la página principal donde el usuario podrá visualizar los temas, el mapa conceptual, una videoconferencia, un juego de entretenimiento, el glosario de términos o consultar la ayuda. Solamente desde la pantalla de los temas, podrá acceder a los ejercicios de cada tema en específico. El usuario podrá abandonar la aplicación desde cualquier pantalla que desee, y una vez realizada esta operación, se visualizarán los créditos.

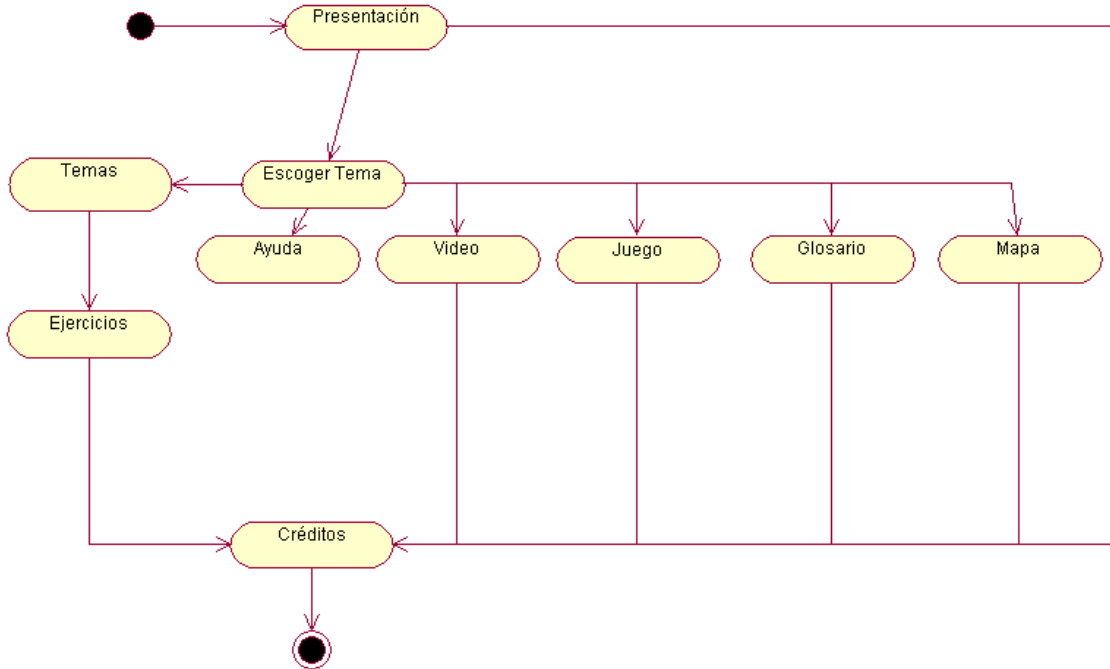


Figura 7: Diagrama de navegación general

### 2.6.1 Sub Modelos de navegación por temas

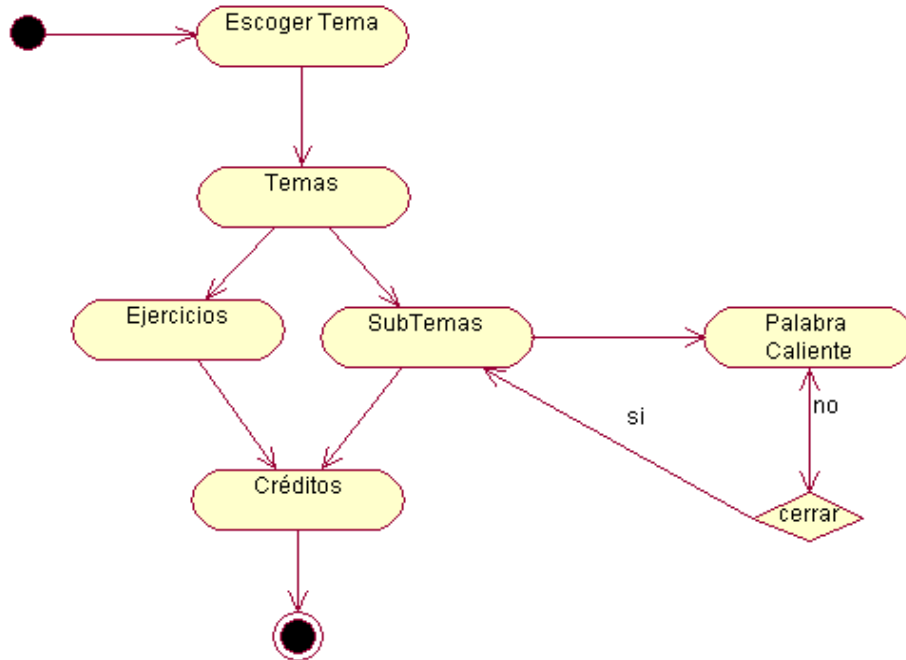


Figura 8: Diagrama de navegación Temas

## 2.7 Descripción de la funcionalidad

A continuación se muestra un listado de los requisitos funcionales y no funcionales que se han tenido en cuenta para el desarrollo de la multimedia.

### 2.7.1 Requisitos Funcionales

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

Requisitos	Función
R1	Mostrar una pantalla de presentación con el nombre del producto, que de inicio a la página principal del curso.
R2	Reproducir música de fondo automática de forma predeterminada
R3	Permitir al usuario (pausar y reanudar) la música de fondo
R4	Permitir la navegación por el contenido de los temas
R5	Mostrar palabras calientes (para los temas y para los conceptos del mapa)
R6	Permitir al usuario la navegación por los subtemas que componen los temas
R7	Mostrar información contenida dentro del subtema del tema seleccionado
R8	Permitir al usuario seleccionar los subtemas existentes dentro de un tema
R9	Permitir al usuario desplazar el Scroll de texto para una completa visualización de la información contenida dentro de un tema
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 didáctico que se encuentra disponible
R13	Mostrar información referente a las palabras calientes consultadas por el usuario
R14	Permitir al usuario visualizar materiales complementarios

R15	Visualizar videoconferencia
R16	Controlar el sonido del videoconferencia
R17	Permitir maximizar el videoconferencia
R18	Permitir visualizar el mapa conceptual
R19	Mostrar concepto asociado al nodo seleccionado en el mapa conceptual
R20	Permitir al usuario visualizar el glosario de términos
R21	Mostrar los términos del glosario ordenados alfabéticamente
R22	Permitir siempre ir a la pantalla principal desde cualquier otra.
R23	Mostrar una ayuda que contenga una breve explicación de todas las pantallas y botones que tiene la aplicación
R24	Permitir salir de la aplicación en cualquier momento que el usuario desee
R25	Mostrar créditos al salir de la aplicación

## 2.7.2 Requisitos no Funcionales

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

### 2.7.2.1 Requisitos de apariencia

- ✓ El sistema debe contener una interfaz sencilla e intuitiva, que brinde facilidades y sea agradable al usuario
- ✓ La aplicación debe utilizar como idioma principal el español, excepto aquellas palabras técnicas que no puedan ser traducidas.
- ✓ El sistema proporcionará una correcta organización de la información, permitiendo una buena interpretación de la misma.
- ✓ Utilizar botones que expresen su función, ya sea que se intuya o expresados con texto.

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

### 2.7.2.2 Requisitos de usabilidad

- ✓ Los usuarios que utilizarán el sistema deberán tener conocimiento básico del manejo de la computadora, así como del trabajo con sistemas operativos visuales.
- ✓ La aplicación deberá poseer una interfaz y navegación acorde y funcional, tanto para usuarios expertos, como para los que no tienen conocimientos profundos de Informática.

### 2.7.2.3 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 podrá ser visualizada..

Windows	
Plataforma	Navegador
Microsoft® Windows® Vista	Microsoft Internet Explorer 7, Firefox 2.0, AOL 9, Safari 3.x o superior
Microsoft Windows XP	Microsoft Internet Explorer 6.0 o superior, Firefox 1.x, Firefox 2.x, Mozilla 1.x o superior, Netscape 7.x o superior, AOL 9, Opera 7.11 o superior, Safari 3.x o superior.
Windows Server® 2003	Microsoft Internet Explorer 6.0 o superior, Firefox 1.x, Firefox 2.x
Windows 2000	Microsoft Internet Explorer 5.x, Firefox 1.x, Firefox 2.x, Mozilla 1.x, Netscape 7.x o superior, AOL 9, Opera 7.11 o superior.
Windows Me	Microsoft Internet Explorer 5.5, Firefox 1.x, Mozilla 1.x, Netscape 7.x o posterior, AOL 9, Opera 7.11 o superior
Windows 98	Microsoft Internet Explorer 6.0 o superior, Firefox 1.x, Mozilla 1.x, Netscape 7.x o superior, Opera 7.11 o superior
Linux	
Plataforma Navegador Red Hat® Enterprise Linux®	Firefox 1.5.0.7 o superior; Mozilla 1.7.x o superior; SeaMonkey 1.0.5 o superior.

(RHEL) 3 actualización 8, RHEL 4 actualización 4 (AS/ES/WS)	
Novell SUSE 9.x o 10.1	Firefox 1.5.0.7 o superior; Mozilla 1.7.x o superior; SeaMonkey 1.0. superior.

#### 2.7.2.4 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.

#### 2.7.2.5 Requisitos de diseño e implementación

- ✓ Las herramientas a usar en el diseño gráfico de la aplicación serán el Macromedia Flash 8.
- ✓ El contenido se cargará desde archivos XML; y el lenguaje de programación será ActionScript 2.0.

#### 2.7.2.6 Requisitos de soporte

- ✓ La aplicación es extensible a plataformas Web sin alterar de algún modo el contenido de sus datos.
- ✓ Para su correcto funcionamiento, la computadora donde se ejecute la multimedia deberá tener tarjeta de video, tarjeta de sonido y demás aditamentos para la reproducción de sonido.

### 2.8 Modelos de casos de uso del sistema

El modelado de casos de uso es una de las técnicas que se utilizan para modelar los requisitos del sistema. Los casos de uso son fragmentos de funcionalidad que el sistema ofrece para aportar un resultado de valor para sus actores, permitiendo modelar el funcionamiento del sistema de la forma que el cliente desee. Utilizando UML, se pueden representar las funcionalidades con las que debe cumplir el sistema, o sea, los requisitos funcionales y los actores que van a interactuar con el producto, mediante un diagrama de casos de uso.

### 2.8.1 Determinación y justificación de los actores del sistema

Actor	Justificación
usuario	Entidad (persona) que va a entrar a la aplicación a consultar información sobre Gestión de Calidad de Software.

### 2.8.2 Diagrama de caso de uso

En la siguiente figura se muestra el actor y los casos de uso del sistema identificados en este trabajo.

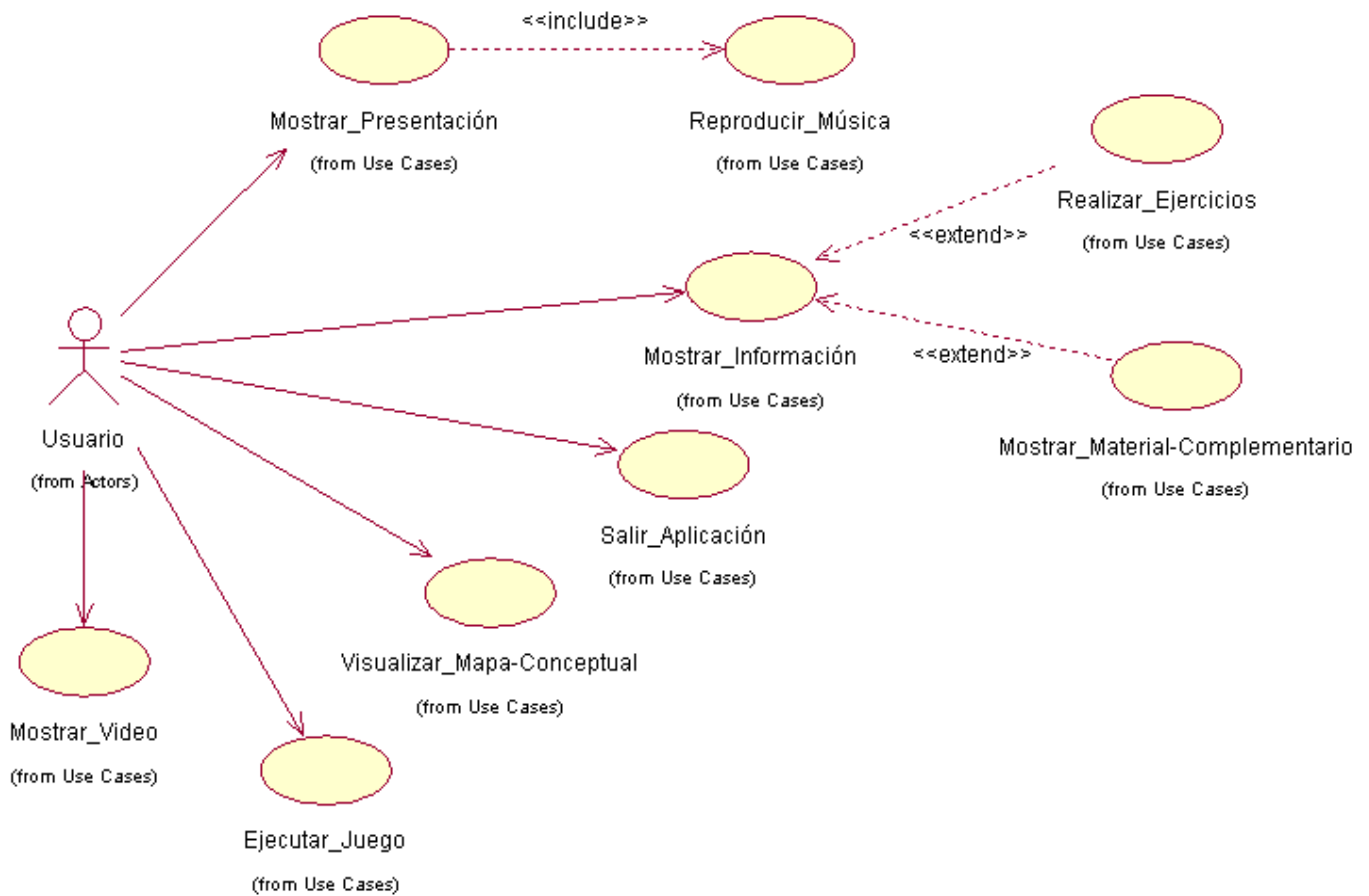


Figura 9: Diagrama de casos de uso

### 2.8.3 Breve explicación de los casos de uso del sistema

A continuación un pequeño resumen que describe la funcionalidad del caso de uso (CU) y el requisito al que da respuesta.

<b>Caso de Uso:</b>	<b>Mostrar Presentación</b>
Actor(es):	Usuario
Propósito:	Mostrar la presentación de la multimedia.
Resumen:	Al ejecutar la aplicación se muestra la presentación del producto, la cual incluye la reproducción de un sonido de fondo de manera automática. Al concluir la presentación de la aplicación se pasa automáticamente al menú principal de la misma.
Referencias:	R1,R2,R3

<b>Caso de Uso:</b>	<b>Mostrar Información</b>
Actor(es):	Usuario
Propósito:	Mostrar la información contenida en la multimedia.(Texto, imágenes, información contenida en palabras calientes, conceptos pertenecientes al mapa conceptual, ejercicios y materiales complementarios).
Resumen	El sistema muestra la información para que el usuario seleccione cual información desea ver. Hay varios subtemas dentro de cada unos de los temas, conceptos pertenecientes a palabras calientes o mapa conceptual, materiales complementarios, ejercicios, etc. Para poder visualizar la información que desea, tendrá que hacer clic sobre el tema y subtema específico contenido dentro del tema que contiene la información deseada y el sistema se encargará de obtener y mostrar la información.
Referencias:	R4,R5,R6,R7,R8,R9,R 13,R22



<b>Caso de Uso:</b>	<b>Mostrar Video</b>
Actor(es):	Usuario
Propósito:	Mostrar la videoconferencia contenida en la multimedia.
Resumen	El caso de uso se inicia cuando el usuario decide visualizar la videoconferencia contenida en la aplicación. El usuario se dirige al icono que da la opción de visualizar la misma y con sólo hacer clic, se muestra la pantalla en la cual se verá dicho video y desde donde podrá realizar diferentes actividades como parar, pausar, reproducir, subir o bajar el volumen del mismo.
Referencias	R15, R16, R17, R22

<b>Caso de Uso:</b>	<b>Realizar Ejercicio</b>
Actor(es):	Usuario
Propósito:	Mostrar y ejecutar los ejercicios de los temas estudiados en el curso.
Resumen	El caso de uso se inicia cuando el usuario, después de haber consultado un determinado tema, decide realizar ejercicios del tema estudiado haciendo clic sobre el último botón del menú desplegable que se llama Ejercicios. Seguidamente se muestra en pantalla con un cuestionario que el usuario deberá responder haciendo clic sobre la respuesta que considere correcta. Para este cuestionario se utilizarán preguntas de verdadero y falso o de selección simple. De manera automática, la aplicación irá pasando de una pregunta a otra hasta llegar al final del cuestionario. Si el usuario no responde correctamente, se le mostraran las respuestas correctas.
Referencias:	R10, R 11, R22

<b>Caso de Uso:</b>	<b>Ejecutar juego</b>
Actor(es):	Usuario
Propósito:	Ejecutar el juego disponible en la aplicación.
Resumen	El caso de uso comienza cuando el actor selecciona el icono de juego en la barra contenedora de todos los iconos en la parte inferior de la página principal. En ese momento aparecerá el juego del ahorcado, el cual consiste en descubrir una palabra relacionada con el tema que se trata en el curso. El juego irá guardando en la memoria de la máquina un historial que será con valores positivos o negativos, en dependencia de la cantidad de aciertos o fallas con respecto a las

	palabras a adivinar.
Referencias:	R12, R22

<b>Caso de Uso:</b>	<b>Visualizar Mapa</b>
Propósito:	Mostrar el mapa conceptual que contiene los principales conceptos del curso de estudio.
Actor(es):	Usuario
Resumen:	El caso de uso comienza cuando el actor hace clic sobre el icono Mapa. Automáticamente se carga una pantalla que contiene el mapa conceptual. El usuario se puede desplazar por el mismo, e ir viendo cada uno de los conceptos que en el mismo muestra a través de nodos interrelacionados. Haciendo clic sobre un nodo, podrá visualizar en una ventana flotante con el concepto del mismo.
Referencias:	R18, R19, R22

<b>Caso de Uso:</b>	<b>Mostrar materiales complementarios.</b>
Actor(es):	Usuario
Propósito:	Mostrar materiales complementarios que ayuden a una mejor comprensión de un determinado tema.
Resumen:	El caso de uso comienza cuando el actor, después de haber consultado información referente a un tema determinado, decide visualizar un documento que la aplicación le aconseja que visualice para una mejor comprensión del tema que se trata. Dando clic se abre el material complementario seleccionado.
Referencias:	R14, R22

<b>Caso de Uso:</b>	<b>Visualizar Ayuda</b>
Actor(es):	Usuario
Propósito:	Mostrar una guía al usuario que ayude la navegación por la aplicación
Resumen:	El caso de uso comienza cuando el actor decide consultar la ayuda, porque necesita, una mejor orientación por los contenidos de la aplicación. Dando clic sobre el botón de ayuda, aparece una guía de navegación al usuario.

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

<b>Caso de Uso:</b>	<b>Visualizar Glosario de términos</b>
Actor(es):	Usuario
Propósito:	Mostrar un glosario de términos ordenados alfabéticamente que ayuden a la comprensión de algunos de los conceptos de difícil entendimiento al usuario.
Resumen:	El caso de uso comienza cuando el actor, después de haber consultado información referente a un tema determinado, decide consultar el concepto de una determinada palabra que desconoce su significado. Dando clic sobre el botón glosario, aparece un glosario de todos los términos de comprensión más compleja que contiene la aplicación, ordenados alfabéticamente.
Referencias:	R20,R21, R22

<b>Caso de Uso:</b>	<b>Salir de la aplicación.</b>
Actor(es):	Usuario
Propósito:	Permitir salir de la aplicación en cualquier momento deseado y desde cualquier lugar de la multimedia.
Resumen:	El caso de uso inicia cuando el usuario desea salir de la aplicación. En cualquier instante que el usuario lo desea, al hacer clic en el botón de salir, la aplicación le mostrará un mensaje de confirmación y el usuario puede aceptar o cancelar la opción de salir. Una vez confirmada la salida, se mostrarán los créditos automáticamente.
Referencias:	R24, R25

## Conclusiones

En el capítulo se deja bien claro la audiencia a la cual va dirigida la aplicación y se especifican los contenidos que se van a mostrar en la misma. Quedan definidos los conceptos relacionados con el negocio estudiado, los cuales quedaron plasmados mediante un diagrama de dominio y los requisitos funcionales y no funcionales que debe tener el producto. Finalizado de esta forma el capítulo, se puede proceder al diseño e implementación de la aplicación.

## CAPÍTULO 3: SOLUCIÓN DE LA DESCRIPCIÓN PROPUESTA

### Introducción

En este capítulo se expondrán los principios y normas de diseño que se usarán en la implementación de la multimedia. Se modelarán los diagramas de presentación que exige el lenguaje de modelado seleccionado. Se realizará el diagrama de componentes correspondiente a la implementación. Se concluirá con una descripción de los XML que se usarán en la multimedia.

### 3.1 Análisis de la arquitectura de la información utilizada

La interfaz del software multimedia debe ser lo más navegable e intuitiva posible, ya que mediante ésta es que el usuario interactúa con la información que está presente en el producto. Un buen diseño del producto proporcionará facilidad de uso y a la vez facilidad de aprendizaje para el usuario. Por todo esto es que se hace necesario adoptar diferentes métodos y procedimientos de diseño para que el producto multimedia cumpla con las expectativas de los usuarios.

#### 3.1.1 Principios y normas de diseño

Con el objetivo de lograr una interfaz amigable, y que dicha interfaz posibilite un bienestar al usuario que interactúe con esta, o sea que el usuario cumpla sus expectativas; se analizaron varias multimedias de enseñanza para ver la manera en que trataban los temas de organización de la información. Además se tuvo en cuenta la audiencia a la cual va dirigida y la temática que se va a impartir en la aplicación.

Como normas de diseño se definen las siguientes:

- Las pantallas de contenido tienen la información necesaria para evitar la sobrecarga.
- Se mantienen todos los iconos de la aplicación de manera visible en todas las pantallas, para que el usuario visite lo que desee en cualquier momento, y así lograr una mejor navegabilidad.
- El sistema va a contener colores serios y poco llamativos, ya que es una aplicación para enseñar una temática.
- La aplicación debe tener una pantalla de ayuda donde brinde información al usuario acerca de cómo navegar por el contenido de la misma.

### 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. Esta muestra una explicación del curso y cómo navegar en la aplicación 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 a gusto del usuario.

#### **Color**

Los colores que predominan son el azul oscuro, por ser el color que representa a la UCI, y el amarillo por su combinación con el azul; además es un color que ayuda a tener ideas claras, y ambos se relacionan mucho con la parte más intelectual de la mente.

### 3.1.3 Estándares de codificación

Toda aplicación implementada debe tener un estándar de codificación que guíe al que esté leyendo para conocer qué elementos está tratando, sin tener que buscar en toda la aplicación, paso a paso.

#### **Elegir nombres representativos para las variables, funciones e instancias:**

Se utilizan nombres representativos para que a simple vista se den cuenta de cuál es la función de cada uno de los elementos. Son escritos con minúscula, juntos, o con ayuda de un separador si son dos o más palabras.

Ejemplos:

ventanacerrar\_btn

inicio\_btn

ayuda\_btn

### **Utilizar sufijos para nombres de instancia:**

Cuando se crean instancias de clases como MovieClip, o Button, además de elegir un nombre representativo para las mismas, se hace uso de los sufijos correspondientes para que, al utilizar la instancia, se pueda acceder fácilmente a la ayuda de referencia y a la lista de los métodos y las propiedades correspondientes a la clase de la cual se hace referencia.

Ejemplos:

Button: salir\_btn.

MovieClip: principal\_mc

Grafico: grafico\_gc

### **Comentar el código:**

Se comenta el código para permitir una interpretación o depuración más rápida si se necesita analizar éste posteriormente, ya sea por terceras personas o por el autor quien, después de cierto tiempo, no tiene por qué recordarlo fielmente.

```
var sonido_fondo = new Sound();
```

```
// Crea una variable para cargar el sonido.
```

```
var volumen=100;
```

```
// Pone la variable en 100 para alcanzar el máximo volumen
```

```
sonido_fondo.loadSound("recursos/yanni.mp3",true);
```

```
// Se carga el sonido de la dirección pasada por parámetros.
```

```
function Reproducir_Fondo_Musica() // Esta función comienza a reproducir el sonido cargado de forma automática.
```

```
{  
    sonido_fondo.start();  
};
```

### 3.2 Descripción de archivos XML

Para almacenar la información, dígame (textos, imágenes, gráficos) que representen el contenido de la multimedia se utilizó la tecnología XML, lo que permite almacenar y modificar la información sin necesidad de interactuar con la aplicación. Para cada tema se crea un archivo XML con la siguiente estructura:

#### Descripción de archivos XML Tema

```
<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>
<font color="#FF0000"><a href="asfunction:_root.Nombre_Función, definición de la palabra
caliente "><b>palabra caliente</b></a></font>
  <img src = "dirección de la imagen" height = "largo" width= "ancho"></img>
  <texto_ subtema_n> Texto que pertenece al subtema 1 del tema</texto_ subtema_n>
</tema>
```

## Descripción de los archivos XML ayuda

```
<ayuda>

    <bienvenida><p align='center'><font color='#000000' size='22'></font></p>

Bienvenido al curso Gestión de Calidad 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 cómo
puede navegar la misma.

<img src = "dirección de la imagen" height = "largo" width= "ancho"></img>

</descripción>

</ayuda>
```

## Descripción de archivos XML ejercicios

```
<quiz>

    <title> Nombre (OMG, 2007)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>

    <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>

.

.

<item n>.....</item 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.

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

</items> cierre de nodo.

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

</item> 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.

### 3.3 Diagramas de Presentación del modelo de Análisis

A través de los diagramas de presentación (Figura 10-14) requeridos también por el uso de OMMMA-L, conocidos como escenas de un guión o escenarios de interacción con el usuario, se hace una ubicación espacial de todos los elementos que conforman una aplicación con tecnología multimedia.

Para una mejor visualización de la correspondencia entre los diagramas de navegación y las pantallas de la multimedia. Ver Anexos (8-13).

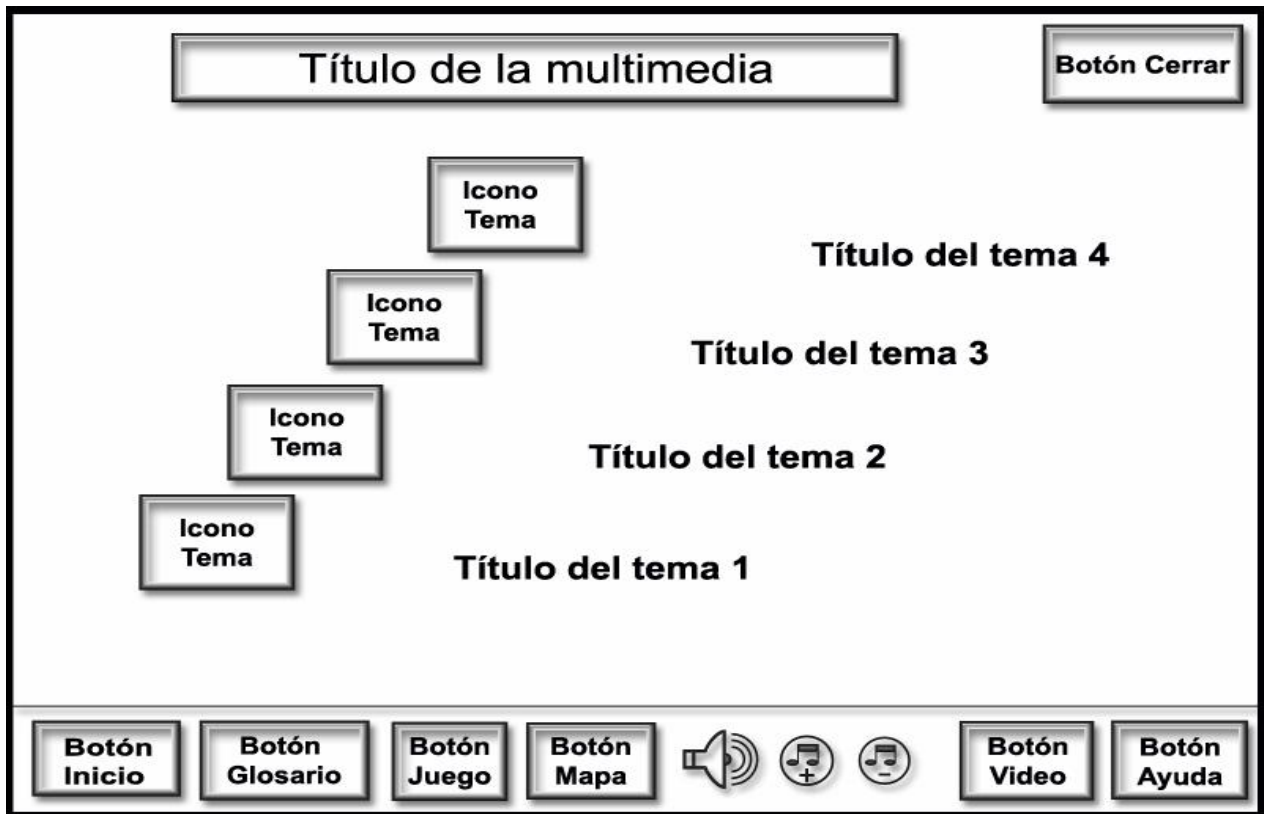


Figura 10: Diagrama de presentación Pantalla Presentación

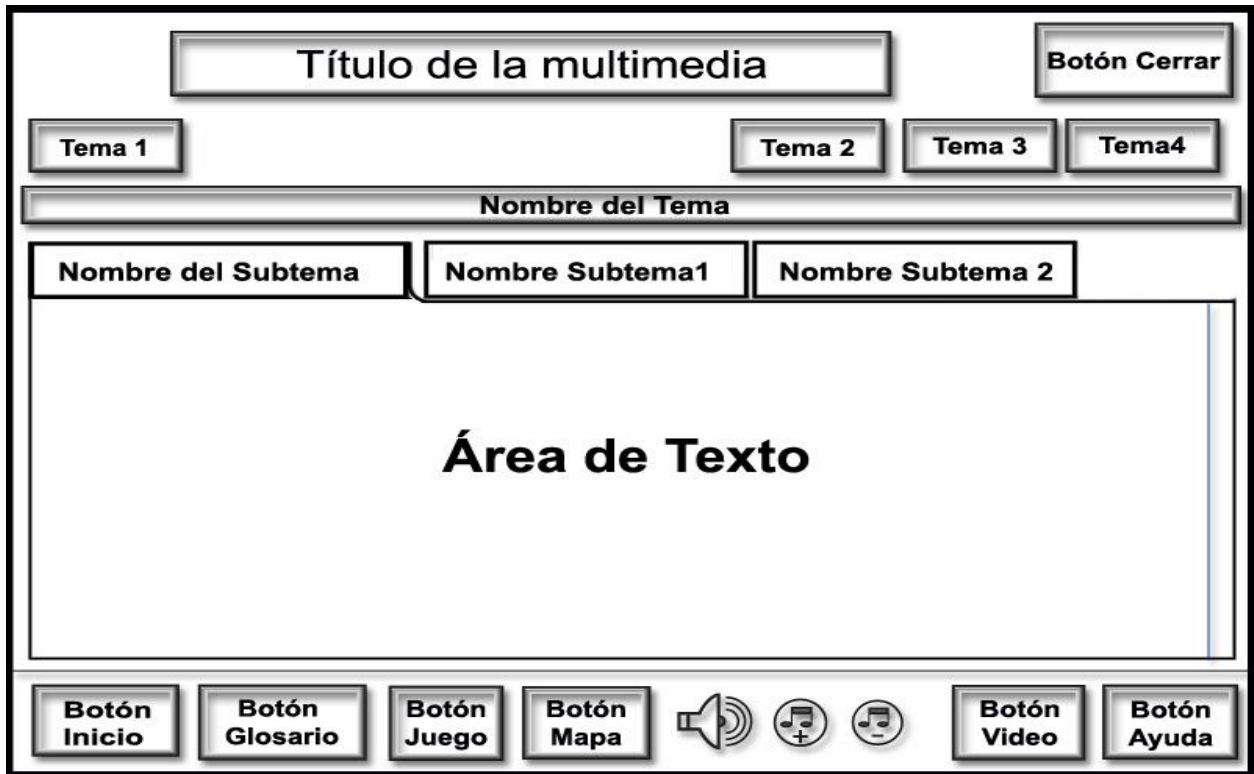


Figura 4 Diagrama de presentación Pantalla Temas

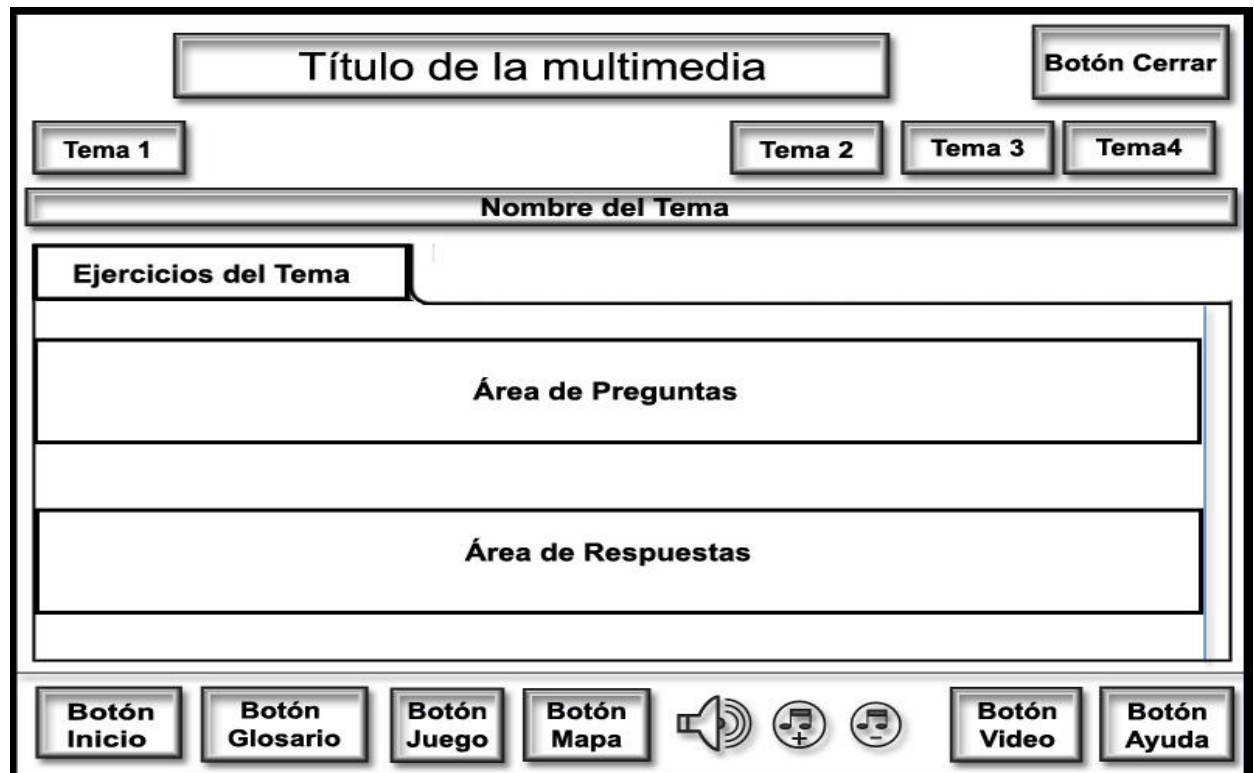


Figura12: Diagrama de presentación Pantalla Ejercicios

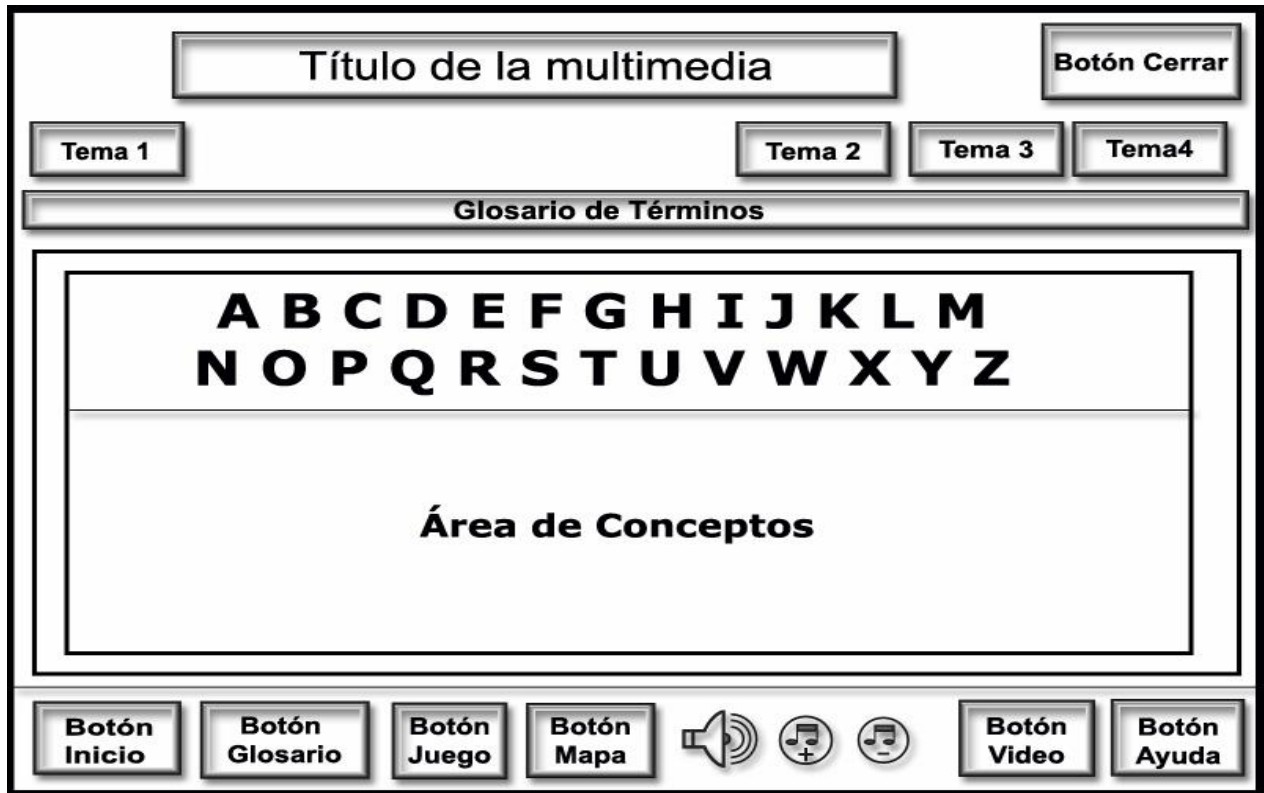


Figura 13: Diagrama de presentación Pantalla Glosario



Figura 5: Diagrama de presentación Pantalla Temas

### 3.4 Diagrama de Componentes

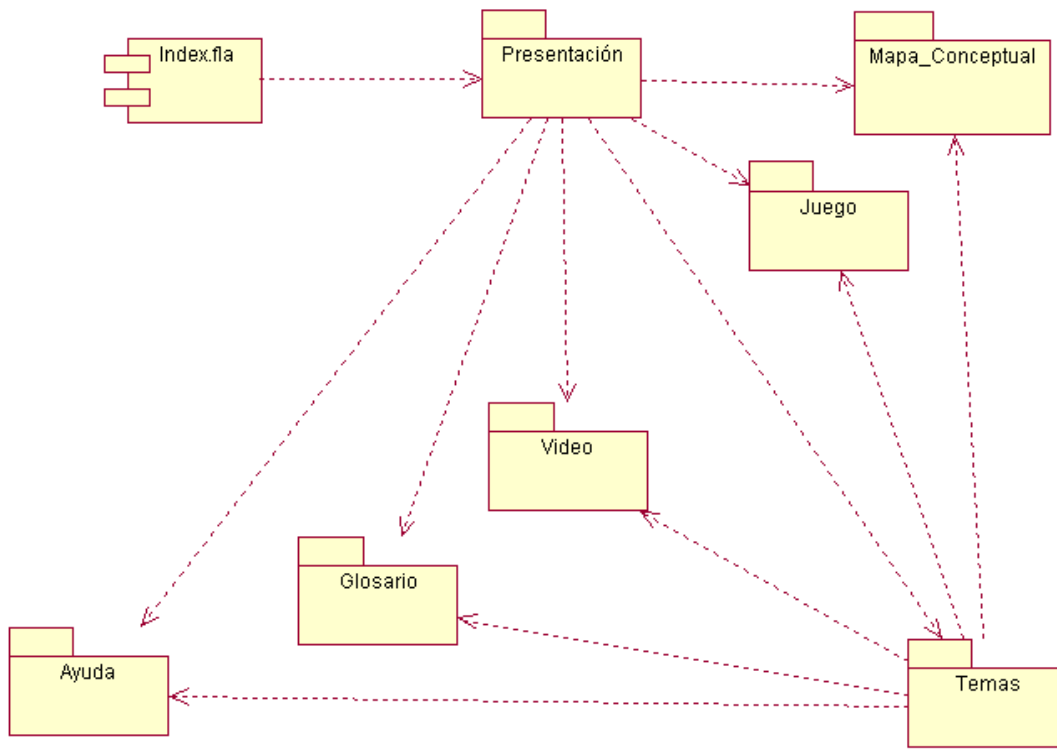


Figura 6: Diagrama de componentes ajustado por paquetes

Para ver los diagramas de cada paquete por separado. Ver Anexos (1-7).

### Conclusiones

En este capítulo se determinaron todas las normas de diseño e implementación necesarias para cumplir con los requisitos de la aplicación. Se realizó una descripción de los XML usados en la aplicación. En este capítulo es donde se le dio uso al lenguaje OMMMA-L, con la elaboración de los diagramas de navegación y presentación. Se realizó el diagrama de componentes ajustado por paquetes y no se realizó un diagrama de despliegue debido a la ausencia de la necesidad de un servidor que almacene la aplicación.

## CONCLUSIONES GENERALES

Después de haber realizado un análisis de todos los elementos necesarios para el desarrollo de esta aplicación, y haberle dado cumplimiento a las tareas planteadas en la investigación, se puede dar por concluido que:

- Se desarrolló una aplicación multimedia interactiva del tema de Gestión de Calidad de Software, que cumple con todos los requisitos que se plantearon en la descripción de la solución, utilizando los lenguajes de modelado y programación seleccionados.
- Se utilizó la metodología de desarrollo seleccionada Racional Unificad Process (RUP), porque además de las características que presenta, es la más conocida por el autor y tiene antecedentes en la colección de multiendias.
- Se utilizó la herramienta Macromedia Flash 8, con su lenguaje de programación ActionScript, en su versión 2.0 para el desarrollo del producto.
- La aplicación cumple con todas las normas y estándares de diseño e implementación planteados en la solución de la descripción propuesta.
- Se realizó el mapa de conceptos relacionando el contenido de todos los temas de la multimedia.

## RECOMENDACIONES

- Se recomienda integrar en un único producto las aplicaciones multimedia: Gestión de Calidad de Software, Métricas de Software, Gestión de la Configuración y Auditorías y Revisiones, con el objetivo de que los estudiantes puedan acceder desde una misma aplicación a cualquiera de los cursos que desee consultar.
- Incrementar los materiales complementarios del software, para lograr una variedad de documentación que permita un estudio más profundo del tema. (AulaClic, 2004)
- Mejorar los mapas conceptuales desarrollados, agregándoles nuevos conceptos y relaciones.
- Valorar la posibilidad de migrar la aplicación a software libre.
- Adicionar al contenido de la multimedia los resultados del programa de mejoras que se está llevando a cabo en la universidad.

## REFERENCIAS

- AulaClic. 2004.** Introducción a ActionScript 2(l). [En línea] 2004. [http://aulaclip.es/flashmx\\_2004/t\\_17\\_1.htm](http://aulaclip.es/flashmx_2004/t_17_1.htm).
- . **2006.** Lo nuevo de flash 8. [En línea] 2006. [http://flash.ciberaula.com/articulo/flash\\_8](http://flash.ciberaula.com/articulo/flash_8).
- Bianchini, Adelaide. 2000.** Conceptos y definiciones de hipertexto. [En línea] 2000. <http://www ldc.usb.ve/~abianc/hipertexto.html>.
- Borges, Jorge Luis. 2008.** Narrativas hipertextuales: hacia una redefinición del concepto de hipertexto. [En línea] 2008. <http://cultura.wordpress.com/2008/10/>.
- Ciudad, Febe Angel. 2008.** *DISEÑO DE APLICACIONES EDUCATIVAS MULTIMEDIAS UTILIZANDO UNA NUEVA VERSIÓN DEL LENGUAJE DE MODELACIÓN ApEM – L*. Ciudad de la Habana : s.n., 2008.
- CMMI. 2006.** *Adapting CMMI for Acquisition Organizations: A Preliminary Report*. Pittsburgh : Carnegie Mellon University, 2006.
- Crosetti, Barbara de Benito. 2000.** Herramientas para la creación, distribución y gestión de cursos a través de Internet. [En línea] 2000. <http://www.uib.es/depart/gte/edutec-e/Revelec12/deBenito.html>.
- Díaz, Abel Ernesto Lorente. 2005.** PLATAFORMA PARA EL DESARROLLO Y GESTIÓN DE CURSOS EDUCATIVOS MULTIMEDIA. [En línea] 3 de 3 de 2005. [http://www.informaticahabana.com/evento\\_virtual/files/MUL064.pdf](http://www.informaticahabana.com/evento_virtual/files/MUL064.pdf).
- Díaz, Barriga F. 1997.** *Estrategias docentes para un aprendizaje significativo. Una interpretación constructivista*. Mexico : McGraw Hill, 1997.
- Díaz, Dionisio. 2004.** Multimedia en la enseñanza. [En línea] 2004. <http://www.dionisiodiaz.com/multimensenanza/multimediaensenanza.html>.
- Dirección de Calidad. 2008.** *Resultado de la Revisión de los proyectos*. Ciudad de la Habana : s.n., 2008.
- Escalona, Fernanda. 2006.** *ESTUDIO COMPARATIVO DE LOS MODELOS Y ESTANDARES DE CALIDAD DEL SOFTWARE*. Buenos Aires : s.n., 2006.
- Febles, Aylin. 2003.** *Un modelo de Referencia para la Gestión de Configuración en la PYME de Software*. Ciudad de la Habana : s.n., 2003.



**Graells, Pere Marqués. 2009.** MULTIMEDIA EDUCATIVO: CLASIFICACIÓN, FUNCIONES, VENTAJAS E INCONVENIENTES. [En línea] 2009. <http://www.pangea.org/peremarques/funcion.htm>.

**IBM Corporation. 2006.** *Rational Unified Process versión 7.2.* 2006.

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

**ISO 9000. 2000.** *ISO/IEC 9000:2000 Standar.* 2000.

**ISO 9001. 2000.** *ISO 9001:2000.* 2000.

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

**Jacobson, Ivar. 2004.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Félix Varela, 2004.

**Jiménez, Sonia. 2008.** Impacto de la tecnología en la Educación. [En línea] 2008. <http://impactodelatecnologia.blogspot.com/2008/02/la-multimedia-como-herramienta-en-el.html> .

**Multimediam. 2006.** ¿Qué es Multimedia Interactiva? [En línea] 2006. <http://portal.educar.org/multimediam/blog/queesmultimediainteractiva>.

**OMG. 2007.** Titulo Introduccion a los OMG Lenguaje unificado de modelado(UML). [En línea] 9 de 11 de 2007. [www.uml.org](http://www.uml.org).

**Pérez, Yancy Martínez. 2006.** PLANTILLA PARA EL MONTAJE DINÁMICO DE LOS. [En línea] 4 de 2006. [http://www.informaticahabana.com/evento\\_virtual/files/MUL067.pdf](http://www.informaticahabana.com/evento_virtual/files/MUL067.pdf).

**Pressman, R. 1998.** *Ingeniería de software. Un enfoque práctico.* Madrid : Mc Graw-Hill Interamericana de España S.A : s.n., 1998.

**Ramos, Lourdes. 2008.** ¿Software educativo, hipermedia o entorno educativo? . [En línea] 2008. [http://bvs.sld.cu/revistas/aci/vol18\\_4\\_08/aci61008.htm](http://bvs.sld.cu/revistas/aci/vol18_4_08/aci61008.htm).

**Rubinstein, D. 2007.** Standish Group Report: There's less development chaos today. SD Times. [En línea] marzo de 2007. <http://www.sdtimes.com/article/story-20070301-02.html>. 169.

**Sabatini, Alejandro Gustavo. 2008.** Hipermedia (Hiperfilmes + Hipertexto + Hipergrama). [En línea] 2008. [http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.1.1.8.-Hipermedia+\(Hiperfilmes+%2B+Hipertexto+%2B+Hipergrama\)](http://interfacemindbraincomputer.wetpaint.com/page/2.A.1.1.1.8.-Hipermedia+(Hiperfilmes+%2B+Hipertexto+%2B+Hipergrama)).

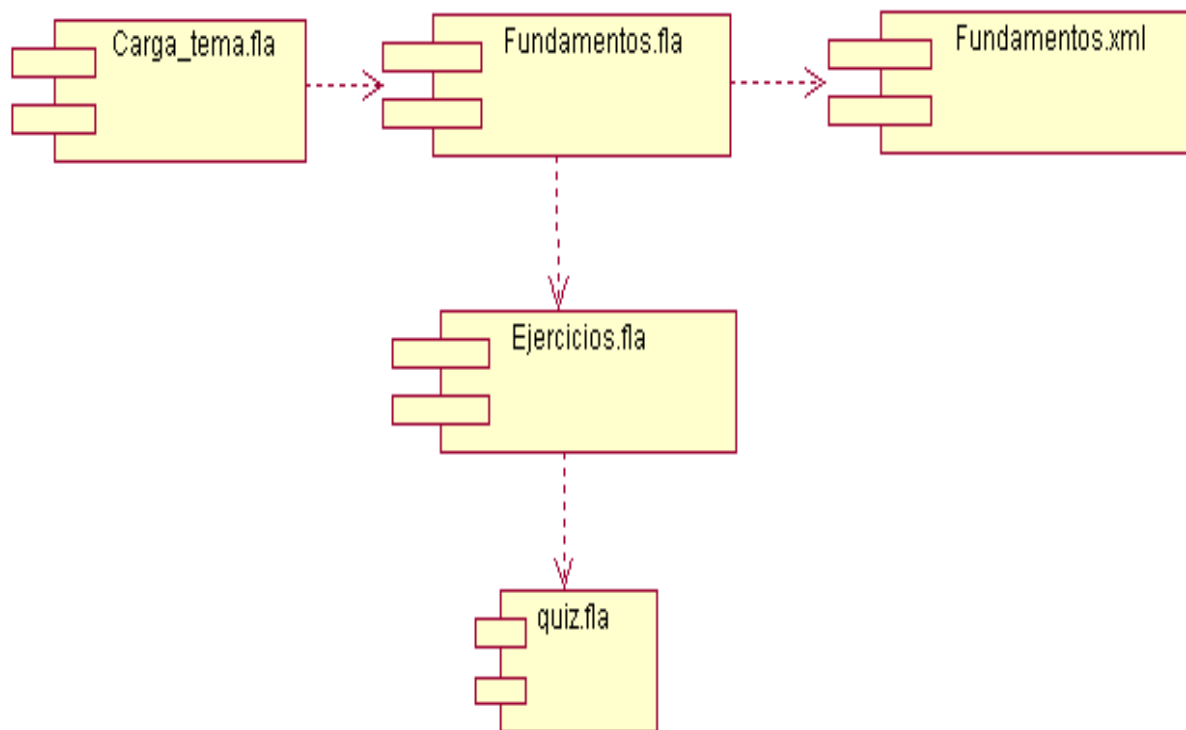
**SWEEBOK. 2004.** *Software Engeniering. Body of Knowlged.* California : s.n., 2004.

**Watts, Humphrey. 2000.** *Introduction to the Team Software Process.* Addison Wesley : s.n., 2000.

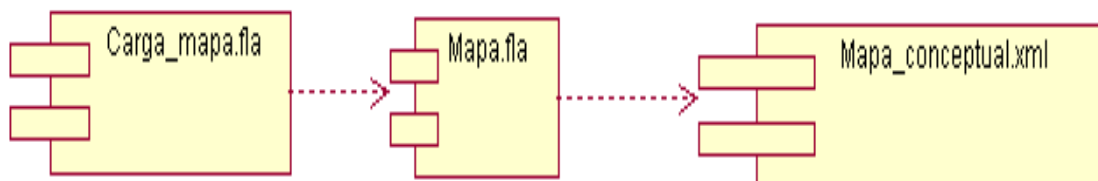
## BIBLIOGRAFÍA

- CMMI. 2006.** *Adapting CMMI for Acquisition Organizations: A Preliminary Report.* Pittsburgh : Carnegie Mellon University, 2006.
- Dirección de Calidad. 2008.** *Resultado de la Revisión de los proyectos.* Ciudad de la Habana : s.n., 2008.
- IBM Corporation. 2006.** *Rational Unified Process versión 7.2.* 2006.
- IEEE. 1990.** *IEEE Standard Glossary of Software Engineering Terminology.* 1990.
- ISO 9000. 2000.** *ISO/IEC 9000:2000 Standar.* 2000.
- ISO 9001. 2000.** *ISO 9001:2000.* 2000.
- ISO/IEC. 1991.** *Norma internacional ISO/IEC-9126.* 1991.
- Jacobson, Ivar. 2004.** *El Proceso Unificado de Desarrollo de Software.* s.l. : Félix Varela, 2004.
- OMG. 2007.** Título Introduccion a los OMG Lenguaje unificado de modelado(UML). [En línea] 9 de 11 de 2007. [www.uml.org](http://www.uml.org).
- Pressman, R. 1998.** *Ingeniería de software. Un enfoque práctico.* Madrid : Mc Graw-Hill Interamericana de España S.A : s.n., 1998.
- SWEEBOK. 2004.** *Software Engeniering. Body of Knolowged.* California : s.n., 2004.
- Watts, Humphrey. 2000.** *Introduction to the Team Software Process.* Addison Wesley : s.n., 2000.

## ANEXOS



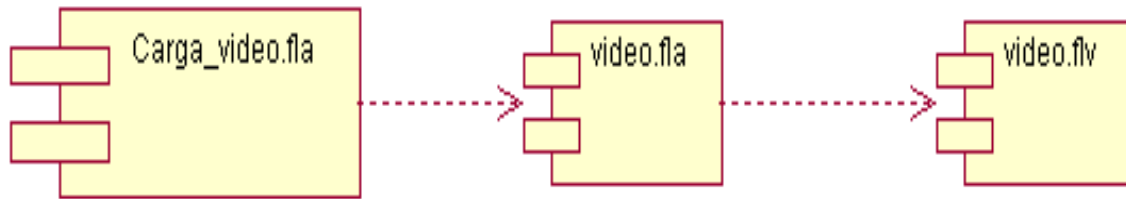
Anexo 1: Paquete Temas



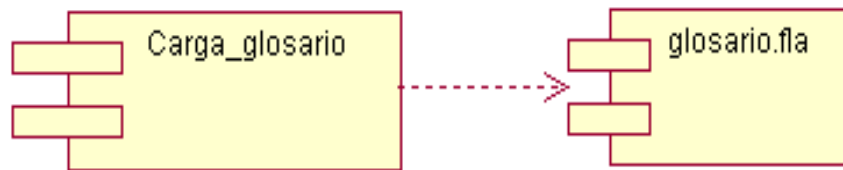
Anexo 2: Paquete Mapa



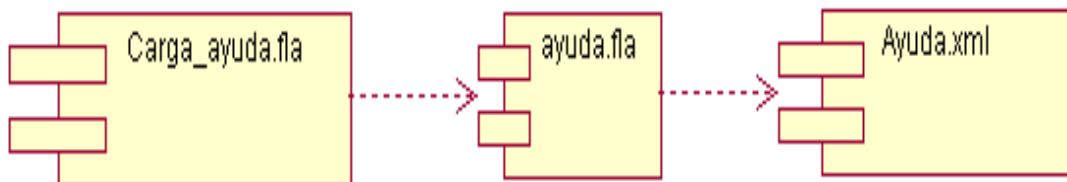
Anexo 3: Paquete Juego



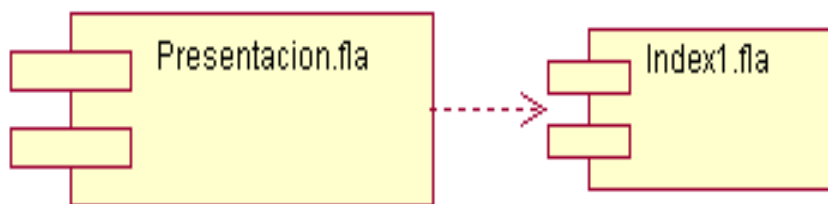
Anexo 4: Paquete Video



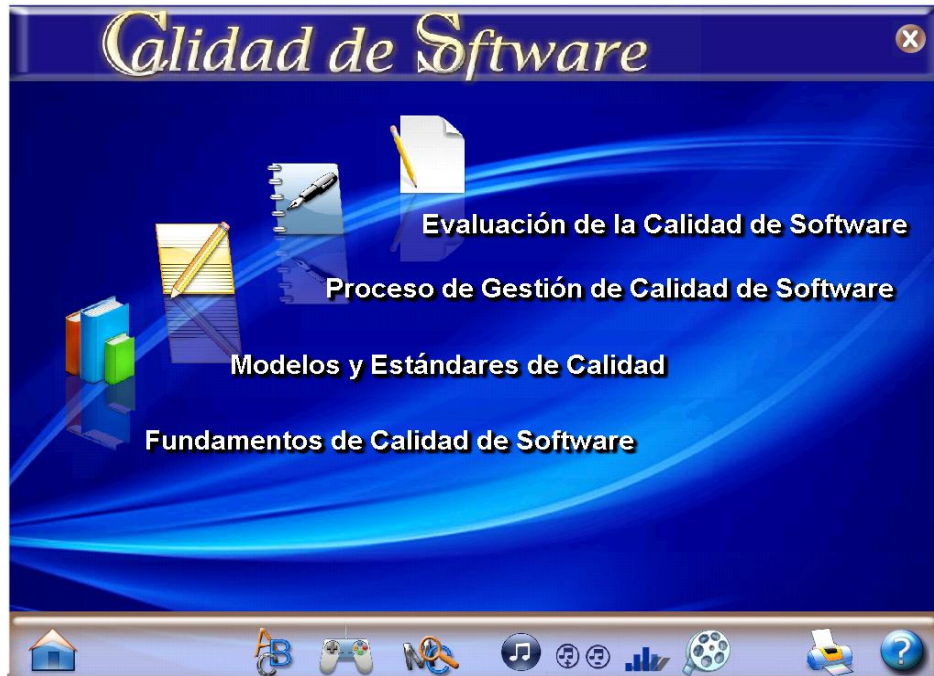
Anexo 5: Paquete Glosario



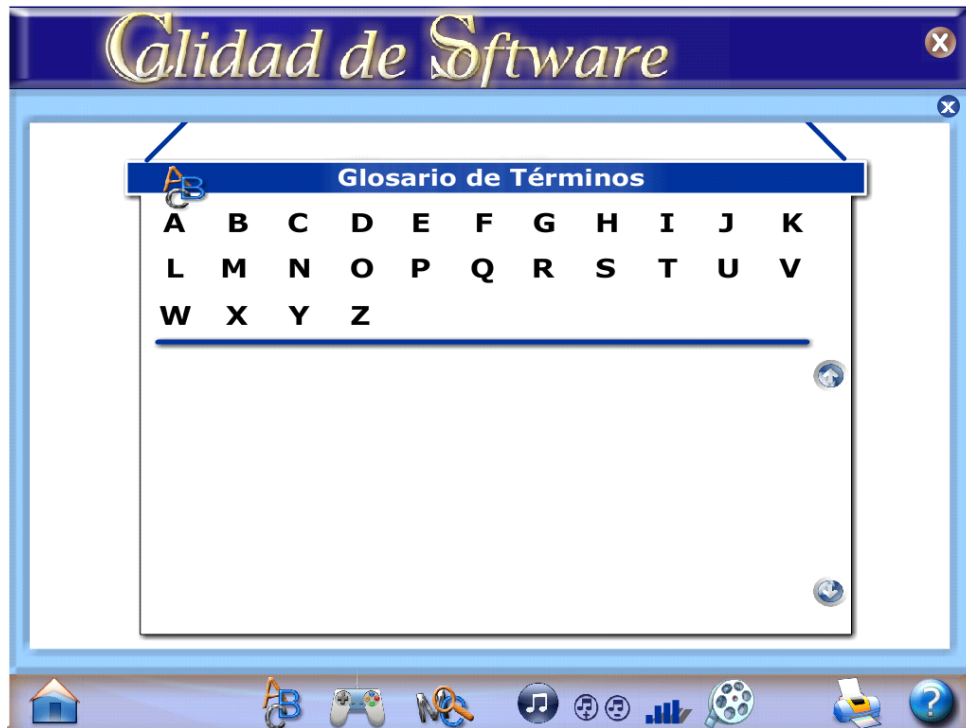
Anexo 6: Paquete Ayuda



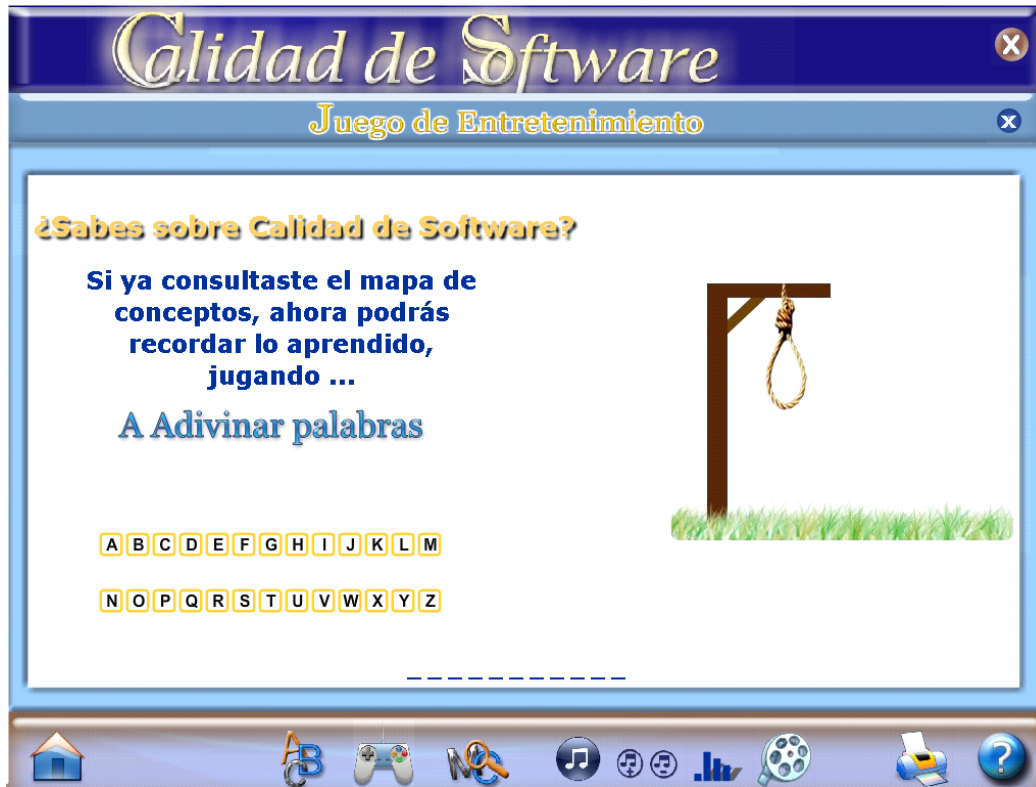
Anexo 7: Paquete Presentación



Anexo 8: Pantalla principal de la multimedia



Anexo 9: Pantalla Glosario de la multimedia



Anexo 10: Pantalla Juego de la multimedia



Anexo 11: Pantalla Mapa de la multimedia



Anexo 12: Pantalla Video de la multimedia



Anexo 13: Pantalla Ayuda de la multimedia



## GLOSARIO DE TÉRMINOS

**Actor:** Cada tipo de fenómeno externo con el que debe interactuar el sistema. (IBM Corporation , 2006)

**Atributo:** Denota las propiedades de un objeto, también puede denotar las propiedades de una clase. (IBM Corporation , 2006)

**Cliente:** La parte (individuo, proyecto u organización) responsable para aceptar el producto o por el que se autoriza el pago. El cliente es externo al proyecto, pero no necesariamente externo a la organización. El cliente puede ser un proyecto de alto nivel. (CMMI, 2006)

**COTS:** Elementos que se pueden adquirir a partir de un proveedor comercial. (COTS significa "fuera de la plataforma comercial."). (CMMI, 2006).

**Diagrama:** Representación gráfica en la que se muestran las relaciones entre las diferentes partes de un conjunto o sistema. (IBM Corporation , 2006)

**Media:** Hace referencia a sonido, texto, imágenes, animaciones, video.

**Metodología:** Es un conjunto de procedimientos, técnicas, instrumentos y documentos que ayudan a los analistas y programadores en sus esfuerzos para obtener un nuevo sistema informativo. Consiste en fases que guían al diseñador en la elección de las técnicas más apropiadas en cada paso del proyecto y lo ayudan a planificar, dirigir, controlar y evaluar el mismo.

**Modelos:** Abstracción semánticamente cerrada de un sistema; descripción completa de un sistema desde una perspectiva determinada (completa significa que no es necesario añadir información adicional para comprender el sistema desde dicha perspectiva). (IBM Corporation , 2006)

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

**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

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

**RUP:** El Proceso Unificado Racional o RUP (Rational Unified Process) es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se

caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso.

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

**Sistema:** Colección de unidades conectadas que se organiza para lograr un propósito. El sistema es el “modelo completo”.

**Software Engineering Institute (SEI):** Instituto de Ingeniería de Software (Software Engineering Institute) de la Universidad Carnegie Mellon (EEUU). Autor de los modelos CMMI y del método SCAMPI.

**Taxonomía:** método de clasificar el pensamiento cognitivo en distintos niveles, cada uno basado en el previo, desde lo más simple hasta lo más abstracto.

**UML:** Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. El UML ofrece un estándar para escribir un “plano” del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables, es un “lenguaje” para especificar y no un método o un proceso. El UML se usa para definir un sistema de software; para detallar los artefactos en el sistema, para documentar y construir.

**Usuario:** Representa a todo aquel que interactúa con la aplicación.

## GLOSARIO DE TÉRMINOS DE LA MULTIMEDIA

**Auditoría:** Un exámen independiente de un producto o conjunto de productos de trabajo para evaluar el cumplimiento de especificaciones, normas, acuerdos contractuales, o por otros criterios. (IEEE, 1990)

**Áreas de proceso:** Conjunto de prácticas relacionadas, que cumplen una serie de objetivos importantes, que ejecutadas colectivamente, satisfacen un conjunto de metas consideradas importantes para hacer mejoras significativas en esa área. (CMMI, 2006)

**Aseguramiento de Calidad:** Forma planificada y sistemática de todas las medidas necesarias para proporcionar suficiente confianza en que un artículo o producto se ajusta a los requisitos técnicos establecidos. (IEEE, 1990)

**Calidad:** El grado en que un sistema, componente o proceso cumple con las necesidades de los usuarios o clientes o expectativas. (IEEE, 1990)

**Caso de Prueba:** Secuencia de acciones (condiciones de ejecución) que identifica comportamientos de interés en el contexto de la ejecución de pruebas. (IBM Corporation , 2006)

**CMMI:** Modelo Integrado de Madurez de la Capacidad. (CMMI, 2006)

**Error:** Condición accidental que hace que la anomalía de un componente del modelo de implementación lleve a cabo su comportamiento requerido. Un error es la causa original de uno o más defectos identificados al observar una o más anomalías. (IBM Corporation , 2006)

**Estándar:** Se refiere a los requisitos formales obligatorios desarrollados y utilizados para prescribir enfoques coherentes al desarrollo (por ejemplo, normas ISO / IEC, las normas de IEEE, normas de organización). (CMMI, 2006)

**Gestión:** Disciplina del proceso de ingeniería de software, cuyo propósito es planificar y gestionar el proyecto de desarrollo. (IBM Corporation , 2006)

**Ingeniería de Software:** La aplicación de un sistemático y disciplinado, cuantificables enfoque al desarrollo, operación y mantenimiento de software. (CMMI, 2006)

**Inspecciones:** Una técnica de análisis que se basa en el examen visual de desarrollo de productos para detectar errores, violaciones de las normas de desarrollo, y otros problemas. (IEEE, 1990)

**Métrica:** Una medida es un atributo de una entidad que se puede medir. Por ejemplo, el esfuerzo del proyecto es una medida del tamaño del proyecto. (CMMI, 2006)

**Procedimiento:** Manera especificada de realizar una actividad. (Manera especificada de realizar una actividad). (ISO 9000, 2000)

**Proceso:** Conjunto de recursos y actividades relacionadas entre si que transforman elementos entrantes en elementos salientes. (ISO 9000, 2000)

**Producto:** Un producto que se destina para la entrega a un cliente o usuario final. (CMMI, 2006)

**Revisión:** Una revisión es una actividad de grupo que se lleva a cabo para descubrir defectos potenciales y para valorar la calidad de un conjunto de artefactos. (IBM Corporation , 2006)

**Rol:** Es un elemento de contenido del método que define un conjunto de habilidades, competencias y responsabilidades que están relacionadas. Los roles sirven a las tareas para especificar quién las realiza así como para definir un conjunto de productos de trabajo de los que son responsables. (IBM Corporation , 2006)

**Validación:** Confirmación de que el producto, según lo previsto, cumplirá con su uso. En otras palabras, validación asegura que "se construye correcto." (CMMI, 2006)

**Verificación:** Confirmación de que el trabajo reflejará adecuadamente los productos y requisitos establecidos para ellos. En otras palabras, la verificación asegura que "se construye bien. (CMMI, 2006)