

**Universidad de las Ciencias Informáticas
Facultad 5**



**Título: Multimedia para la comprensión del tema de
procesos de Ingeniería de Requisitos.**

Trabajo de Diploma para optar por el título de
Ingeniero Informático

Autor: Enrique Linares Carmona

Tutor: MSc. Yamilis Fernández Pérez

Consultante: MSc. Karina Pérez Teruel

“Ciudad de la Habana, Julio de 2008”

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

Enrique Linares Carmona

Firma del Autor

Yamilis Fernández Pérez

Firma del Tutor

DATOS DE CONTACTO

Tutor: Yamilis Fernández Pérez

Correo electrónico: yamilisf@uci.cu

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

*"Never let the future disturb you. You will meet it,
if you have to, with the same weapons of reason which today arm you against
the present."*

Marcus Aurelius Antoninus

DEDICADO A:

A toda mi familia que tanto me apoyó durante estos años de estudio; en especial a mis abuelos Nilo, Mirtha y a mi abuela Miche que siempre está presente. A mi mamá y mi hermana. A mis tíos.

AGRADECIMIENTOS

Este trabajo es el fruto de 5 años de estudio, todo este camino recorrido no hubiera sido posible sin la ayuda de muchas personas que estaban ahí cuando las necesité. Quiero agradecer primeramente a mi mamá por todo lo que ha hecho por mí durante toda mi vida y en especial estos últimos 5 años. Agradecer a mi abuela Mirtha por estar siempre pendiente de mí y por su constante preocupación, a mis tías María, Xiomara, Bertha y Eva por todo su cariño y apoyo. A mi tío Jesús por toda su ayuda en este tiempo; gracias por confiar en mí, a mi abuelo Nilo por darme todo cuanto pudo y por creer siempre en mí, a mi abuela Miche por haber sido alguien tan especial y a quien debo gran parte de lo que hoy soy, a mis demás tíos, a mi hermano Yosvany y mi prima Yenma, a mi hermana Yarita, a mi papá, a toda la familia, gracias. A Lida, sin quien gran parte de esto no hubiera sido posible, gracias. A mi hermana y amiga Ana (mi negra) que siempre estuvo pendiente de mí, a Alidén por ser alguien tan especial y por preocuparse por mi tesis tanto como yo. Quiero agradecer especialmente a Yorji y el Chino por convertirse en lo mejor que uno puede esperar de la palabra amigo, gracias por su paciencia y su apoyo en todo momento, sobretodo en esos en los que más lo necesité. Al Retso (socito) por toda su ayuda. A Nivia y Nivita por toda su ayuda en estos años, a la familia Cuba, a los amigos del grupo, a Yanaris, al Migue, Leo, a Marlén, a la tutora Yamilis por toda su ayuda, a la profesora Zoraida por tanta ayuda y confianza (you know there is a piece of you in this work) y a todos los que de una forma u otra contribuyeron a que pudiera ver un sueño hecho realidad. A los que creyeron en mí; y a los que no lo hicieron esta es una forma de demostrarles cuán equivocados estaban.

....a todos ustedes, gracias.

RESUMEN

En los últimos años ha ido en aumento el interés por los requisitos de software. El alto costo de los errores en actividades relacionadas con los requisitos ha hecho que la Ingeniería de Requisitos (IR) se convierta en un elemento clave que engloba los procesos de desarrollo y gestión de requisitos en el ciclo de vida del software. En la Universidad de Ciencias Informáticas (UCI), es necesario alcanzar un nivel de conocimiento en esta disciplina que permita reducir los errores y demoras en la entrega de productos de software.

En este trabajo se realiza una multimedia para apoyar el proceso de aprendizaje de la disciplina de IR para el segundo perfil de calidad, apoyándose en la confección de mapas conceptuales que engloben los principales conceptos que se relacionan con la disciplina. Se analizan algunas herramientas que puedan ser útiles para el desarrollo de la misma y se da una visión de la distribución del contenido de la misma y de los lenguajes usados para el manejo de datos y la programación de la misma.

Palabras clave

Ingeniería de Requisitos, multimedia, mapas conceptuales.

ÍNDICE

DEDICATORIA	I
AGRADECIMIENTOS	II
RESUMEN	III
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	8
1.1. CALIDAD	8
1.1.2. <i>Definiciones de calidad de software</i>	8
1.2. REQUISITOS DE SOFTWARE	9
1.2.1. <i>Definiciones y conceptos de requisitos</i>	9
1.2.2. <i>Características de un Requisito</i>	10
1.2.3. <i>Requisitos funcionales y no funcionales</i>	11
1.2.4. <i>La Ingeniería de Requisitos</i>	12
1.2.5. <i>Actividades de la Ingeniería de requisitos</i>	13
1.2.6. <i>Importancia de la Ingeniería de Requisitos</i>	14
1.3. MULTIMEDIA. PRINCIPALES CONCEPTOS ASOCIADOS	15
1.3.1. <i>Definición de Multimedia</i>	16
1.4. TENDENCIAS DE TECNOLOGÍAS ACTUALES	17
1.4.1. <i>Los Entornos Virtuales de Aprendizaje (EVA)</i>	18
1.4.2. <i>Herramientas EVA</i>	18
1.4.3. <i>Herramientas para el desarrollo de multimedias</i>	20
1.4.4. <i>El lenguaje de marcas XML</i>	25
1.4.5. <i>Lenguaje de programación. ActionScript 2.0</i>	25
1.5. METODOLOGÍA DE DESARROLLO Y LENGUAJE DE MODELADO	26
1.5.1. <i>RUP</i>	26
1.5.2. <i>OMMMA-L (Extensión UML para multimedia)</i>	27
1.6. HERRAMIENTAS EXISTENTES PARA EL ESTUDIO DEL TEMA	27
1.6.1. <i>Problems and Programmers y SimSE</i>	27
1.6.2. <i>CONTROLA</i>	28
1.6.3. <i>Otros</i>	28
1.7. HERRAMIENTA A UTILIZAR	28
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN	30
2.1. ESPECIFICACIÓN DEL CONTENIDO	30
2.2. DESCRIPCIÓN DEL MODELO DEL DOMINIO	31
2.3. DIAGRAMA DE CLASES DEL MODELO DE DOMINIO	32
2.4. MAPA DE NAVEGACIÓN	33
2.5. REQUISITOS FUNCIONALES DEL SISTEMA.(TABLA 1)	37
2.6. REQUISITOS NO FUNCIONALES	38
2.6.1. <i>Requisitos de Apariencia</i>	38
2.6.2. <i>Requisitos de usabilidad</i>	38
2.6.3. <i>Requisitos de software</i>	38
2.6.4. <i>Requisitos de Hardware</i>	39
2.6.5. <i>Requisitos de diseño e implementación</i>	39
2.6.6. <i>Requisitos de soporte</i>	39
2.7. MODELADO DE CASOS DE USO DEL SISTEMA	39

2.8. MODELO DE CASOS DE USO DEL SISTEMA.....	40
2.8.1. <i>Los actores del sistema. Justificación.</i>	40
2.8.2. <i>Breve resumen de los casos de uso del sistema.</i>	41
2.9. MAPAS CONCEPTUALES.	43
CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA.....	50
3.1. PRINCIPIOS DE DISEÑO.	50
3.1.1. <i>Diseño claro, consistente y con contraste.</i>	50
3.1.2. <i>Diseño de pantallas.</i>	50
3.1.3. <i>Colores.</i>	50
3.2. DESCRIPCIÓN DE ARCHIVOS XML.	51
3.2.1. <i>Estructura de los archivos XML.....</i>	51
3.3. DIAGRAMAS DE PRESENTACIÓN.	57
3.4. MODELO DE IMPLEMENTACIÓN.....	62
3.4.1. <i>Diagrama de Despliegue.....</i>	62
3.4.2. <i>Diagrama de Componentes.</i>	63
CONCLUSIONES.....	65
RECOMENDACIONES.....	66
BIBLIOGRAFÍA.....	67
REFERENCIAS.....	70
ANEXOS.....	73
GLOSARIO.....	76

INTRODUCCIÓN

Hoy en día las tecnologías de la Información han crecido considerablemente en el mercado. Las necesidades de las empresas por gestionar y automatizar el uso de su información son de mayor importancia para la toma de decisiones, los nuevos proyectos y los panoramas a corto y mediano plazo. Sin embargo, aún no se ha tomado conciencia de la importancia de gestionar esta información de manera responsable y con una visión amplia. En ocasiones es más importante tener un producto en el menor tiempo, y a veces el precio de esto es el decremento de la calidad y funcionalidad a muchos sistemas que gestionan esta información. (Jacinto, 2007)

Luis Jacinto en un artículo publicado en el sitio de SIGA¹ titulado "Metodologías de Prueba" expresa que "un producto con calidad es más competitivo en el mercado, se evita que el cliente obtenga un producto defectuoso, promueve metodologías, planificación, fijación de objetivos y promueve la filosofía de la misma" (Jacinto, 2007), idea que revela algunas de las ventajas que reporta producir con calidad.

El desarrollo de la industria del software constituye un sector de vital importancia y se encuentra actualmente a merced de un sinnúmero de transformaciones a nivel mundial. En esta dirección, el control del proceso de calidad del mismo juega un papel primordial a la hora de satisfacer las necesidades del cliente y a la hora de insertar el producto en el mercado mundial.

Cuba no está exenta de la evolución del software, es a partir de 1996, en momentos en que apenas se empezaba a salir de la crisis producida por el derrumbe del campo socialista en Europa y la desintegración de la URSS, que se empiezan a dar los primeros pasos para el ordenamiento de un trabajo continuo con el objetivo de impulsar el uso y desarrollo de las TIC en el país.

La Industria Cubana del Software (ICSW) está llamada a convertirse en una significativa fuente de ingresos nacional. La Universidad de las Ciencias Informáticas (UCI) y el sistema de empresas cubanas vinculadas a este trabajo jugarán un papel importante en el desarrollo de la Industria Cubana del Software, y en la materialización de los proyectos asociados al programa cubano de informatización. (MINREX, 2004)

¹ Empresa Argentina dedicada al desarrollo, comercialización, y soporte de software.

La promoción de la industria cubana del software en el ámbito internacional ha tenido como línea estratégica aprovechar la enorme credibilidad que tiene Cuba en sectores tales como la salud, la educación y el deporte. Continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tiene ya una positiva repercusión en el incremento de la exportación. (MINREX, 2004)

El auge de la industria del software se convierte en un gran reto para la enseñanza de las carreras de Informática en las Universidades Cubanas. Este escenario obliga a que las universidades garanticen la formación de profesionales informáticos fuertemente preparados para el trabajo en el ámbito de un desarrollo de software de forma industrial regido por procesos y controlado por la aplicación de métricas. (Febles, y otros, 2005)

La UCI se ha convertido en un importante centro para la formación de profesionales altamente calificados que puedan asumir la producción de software para impulsar el desarrollo de la nación. Ligado a esto, el proceso de desarrollo y gestión de requisitos juega un papel importante a la hora de lanzar un producto al mercado, ya que inciden directamente en la calidad del producto.

A través de los años, y con el progresivo desarrollo de la Ingeniería del Software se ha podido constatar que los requisitos son la pieza fundamental en un proyecto de desarrollo de software, ya que marcan el punto de partida para actividades como la planeación, básicamente en lo que se refiere a las estimaciones de tiempos y costos, así como la definición de recursos y la elaboración de cronogramas que será uno de los principales mecanismos con los que se contará durante la etapa de desarrollo. (Chaves, 2008)

Que un proyecto falle es común, y según estudios, el 71% de los proyectos que no terminan con éxito, lo hacen por una mala gestión de requisitos, y la principal causa de esta habitual mala gestión de requisitos es la discordancia entre los requisitos de negocio que se esperan, y los que los técnicos entienden al principio y ofrecen después. (ALS, 2008)

Gran parte de los problemas asociados al desarrollo de software están relacionados con los requisitos. El Standish Group estudió en el año 1995 a 352 compañías y 8 000 proyectos de software (Figura 1) y demostró el impacto de estos problemas en los proyectos. Los tres factores que más inciden coinciden con actividades vitales en el ciclo de vida del software y pilares de la Ingeniería Requisitos.

1. Falta de entrada de información por parte de los usuarios: 12.8% de todos los proyectos.

2. Requisitos y especificaciones incompletas: 12.3% de todos los proyectos.
3. Requisitos y especificaciones cambiantes: 11.8% de todos los proyectos.

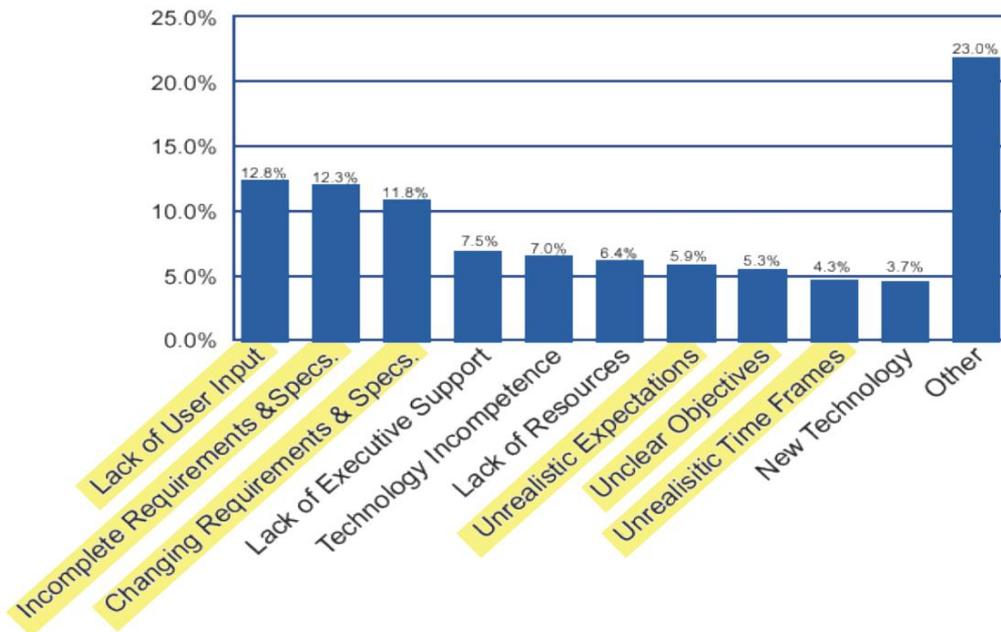


Figura 1: Causas de fracaso de los proyectos de desarrollo de software (Pérez, 2008).

Estudios de la European Software Process Improvement Training Initiative (ESPITI) arrojaron, en una encuesta realizada para identificar la importancia relativa de diversos tipos de problemas del software en la industria, los resultados mostrados en la figura 2, basadas en 3800 respuestas.

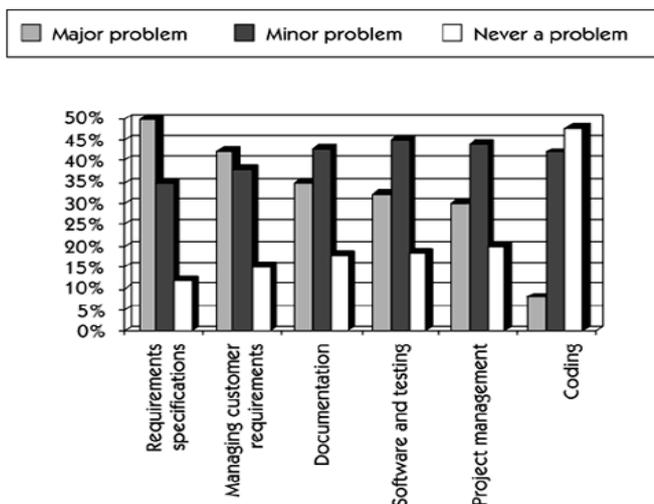


Figura 2: Los mayores problemas de desarrollo de software por categoría (Pérez, 2008).

Los dos mayores problemas que aparecen en aproximadamente la mitad de las encuestas son:

1. Especificación de Requisitos.
2. Gestión de requisitos del cliente.

La UCI no queda exenta de estos problemas. Muchos de los proyectos productivos de la universidad presentan problemas en etapas posteriores a la fase de levantamiento de requisitos incidiendo directamente en la calidad de los mismos, ello se debe en lo esencial a la falta de preparación y de conocimiento del personal encargado de trabajar en esta fase.

En la UCI, en pregrado se enseña la disciplina de requisitos del Rational Unified Process (RUP). Sin embargo no se conocen todas las técnicas, herramientas y actividades necesarias para la identificación, descripción, análisis y validación de los mismos.

Algunos de los datos mostrados a continuación, provenientes de encuestas sobre el tema, confirman lo anteriormente planteado.

Una encuesta realizada por Daliany Soler (Soler, 2007) a líderes de proyectos de la UCI reveló los siguientes datos en relación con el grado de importancia de los problemas más frecuentes en la captura de requisitos:

- El 62% considera que la mala comunicación entre clientes, desarrolladores y usuarios es uno de los factores que genera problemas frecuentes en la captura de requisitos.
- El 72% considera que la falta de preparación del personal que realiza la captura de requisitos es uno de los factores con mayor grado de importancia en los problemas generados en esta etapa.
- El 50% considera además que la mala gestión de cambios en los requisitos es otro factor de importancia en los problemas encontrados en la etapa de captura de requisitos.

En otro orden, la encuesta develó además que el 99% de los estudiantes y el 100% de los líderes de proyectos a los que se les aplicó la encuesta consideraron que había necesidad de aumentar la capacitación sobre temas de la Ingeniería de Requisitos.

Los datos anteriormente expuestos, son un ejemplo claro de la necesidad de aumentar la preparación en el tema de Ingeniería de Requisitos por parte de los integrantes de proyectos productivos y estudiantes de manera general en la Universidad. Esta situación hace que sea necesario buscar una vía para capacitar a los estudiantes que cursen el perfil de Calidad de Software en el tema de IR de manera que alcancen un nivel de conocimiento que evite que proyectos productivos de la UCI sigan presentando dificultades en el desarrollo y entrega de los mismos.

La anterior situación problemática nos presenta el siguiente problema científico: **¿Cómo contribuir a mejorar la capacitación de estudiantes en el tema de Ingeniería de Requisitos?**

Se determina por consiguiente que el objeto de estudio de la investigación es el **proceso de desarrollo de multimedias**, y su campo de acción lo forma **el proceso de desarrollo de multimedias para la enseñanza aprendizaje de los procesos de Ingeniería Requisitos de software**.

El objetivo general es:

Implementar una multimedia interactiva que apoye el proceso de enseñanza aprendizaje del tema de Ingeniería de Requisitos en la UCI.

Para dar cumplimiento a este objetivo se desarrollaron las siguientes tareas:

1. Revisar bibliografía en lo referente el estado del arte del tema, así como, las herramientas, estándares y métodos para enseñar estos temas.

2. Confeccionar mapas conceptuales que contengan los conceptos necesarios para la comprensión de los procesos de Ingeniería de Requisitos.
3. Diseñar un producto multimedia que contenga los mapas conceptuales y asociando a los conceptos diversos medios (videos, juegos, multimedias pequeñas) que ayuden a la comprensión de los conceptos.
4. Investigar sobre las diferentes herramientas de desarrollo de multimedias.
5. Investigar sobre el lenguaje Action Script cómo lenguaje de programación para multimedias y XML como lenguaje de marcas para almacenar el contenido.

La idea a defender del trabajo es:

La utilización de una multimedia interactiva que muestre videos, textos, imagen, mapas conceptuales en el tema de Ingeniería de Requisitos hace posible que se alcance una mejor preparación en los estudiantes involucrados en el desarrollo de productos de software.

Entre los métodos de trabajo científico se destacan:

Los **métodos empíricos** a través de las entrevistas realizadas a estudiantes, profesores y especialistas en la rama que trabajan en la UCI y que tienen conocimientos acerca del tema de calidad, específicamente a aquellos que se dedican en esencia a estudiar los requisitos o a aquellos que hayan hecho trabajos sobre dichos temas.

Los **métodos teóricos** usados en la investigación sirvieron para estudiar las características del objeto de la investigación, el **método histórico – lógico** se utilizó para analizar la trayectoria del desarrollo de los procesos de Ingeniería de Requisitos en el tema de calidad del software en los diferentes períodos desde su aparición, y las principales etapas de su desenvolvimiento. El **analítico sintético** es el método empleado para el procesamiento de la documentación, así como para precisar las diferentes características y métodos para desarrollar herramientas de estudio aprendizaje en los temas de Ingeniería de Requisitos.

El documento consta de 3 capítulos que abarcan el contenido de la investigación.

El **capítulo 1** Fundamentación Teórica trata los conceptos relacionados con el objeto de la investigación así como las tendencias de tecnologías actuales; herramientas, lenguajes de programación y metodología.

En el **capítulo 2** Descripción de la Solución se realiza la especificación del contenido, así como la concepción del diagrama de clases del modelo de dominio. Se documentan además los mapas conceptuales como novedad científica del desarrollo de la investigación y se plasman los requisitos funcionales y no funcionales de la aplicación.

El **capítulo 3** Diseño e Implementación contiene lo que conformaría la construcción del sistema. Incluye los diagramas de presentación de la multimedia, así como el modelo de despliegue y diagrama de componentes de la misma. Contiene además, la conformación de las estructuras XML en la multimedia para el manejo de los datos de la misma.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1. Calidad

La calidad se ha visto como la capacidad para identificar las necesidades y expectativas de los clientes y demás partes interesadas para satisfacerlas, cumpliendo los requisitos del producto o servicio ofrecido, esto adquiere cada vez más importancia en la gestión de las empresas de ahí que los gerentes reconozcan que se pueden obtener ventajas competitivas sustanciales mediante el desarrollo de sistemas de gestión de calidad.

La calidad se ha convertido en el mundo globalizado de hoy, en una necesidad ineludible para permanecer en el mercado. Por ello los sistemas de gestión de la calidad basados en las normas ISO 9001, que reflejan el consenso internacional en este tema, han cobrado una gran popularidad, y muchas organizaciones se han decidido a tomar el camino de documentarlo e implementarlo. (Miranda, y otros, 2008)

1.1.2. Definiciones de calidad de software.

A través de los años diferentes autores y organizaciones han definido el término “calidad de software” de manera diferente. Para Phil Crosby² era la “correspondencia con los requisitos de usuario”. Watts Humphrey se refiere a ésta como “lograr niveles excelentes de preparación para su uso,” mientras que IBM acuñó la frase “market-driven quality”, que está basada en lograr la satisfacción total del cliente. El criterio de Baldrige para la calidad organizacional usa una frase similar, “customer-driven quality,” e incluye la satisfacción del cliente como una mayor consideración. El término calidad ha sido definido en (ISO9001-00) como “el grado con el cual cierta cantidad de características inherentes satisfacen requisitos.” (IEEE Computer Society, 2004). R. S. Pressman (1992) define el término calidad de software como “concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos con los estándares de desarrollo explícitamente documentados y con las características implícitas que se espera de todo software desarrollado profesionalmente”. En 1998 R. S. Pressman define nuevamente el término calidad de software como “concordancia del software producido con los requisitos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requisitos implícitos no establecidos formalmente, que desea el usuario.” ISO 8402 (UNE 66-001-92) define

² Philip Crosby es uno de los pensadores sobre calidad más destacados de los Estados Unidos. Ha publicado más de diez libros en su carrera, el primero (best seller) fue "Quality is Free" o "La Calidad No Cuesta". Otros libros importantes han sido "The absolutes of Leadership" o "Los Absolutos de la Calidad".

calidad de software como “el conjunto de características de una entidad, que le confieren su aptitud para satisfacer las necesidades expresadas y las implícitas.”

De manera que de aquí en lo adelante se puede tratar calidad de software como la forma en que se logra una correspondencia entre las funcionalidades del mismo y la forma en la que se satisfacen las necesidades del cliente de acuerdo a estas funcionalidades y en concordancia con los requisitos establecidos, cumpliendo con las normas y estándares de calidad.

Anteriormente se había planteado que la falta de concordancia con los requisitos se traduce en una falta de calidad, según estudios realizados se demuestra que el 71% de los proyectos que no terminan con éxito lo hacen por una mala gestión de requisitos. (ALS, 2008)

La UCI, como centro de desarrollo de software no queda excluida de estos problemas. El conocimiento de términos asociados a este proceso es de vital importancia para lograr una mejor capacitación profesional. Los epígrafes que ha continuación se tratan desarrollan algunos de los conceptos elementales en materia de requisitos.

1.2. Requisitos de software.

La Ingeniería de Requisitos cumple un papel fundamental en el proceso de producción de software, ya que enfoca un área fundamental: la definición de lo que se desea producir. Su principal tarea consiste en la generación de especificaciones correctas que describan con claridad, sin ambigüedades, en forma consistente y compacta el comportamiento del sistema.

1.2.1. Definiciones y conceptos de requisitos.

Una de las tareas más difíciles al crear sistemas de software consiste en implementar lo que realmente resuelve las necesidades del usuario. Aquellos proyectos contruidos con requisitos incompletos o difíciles de entender suelen terminar con un alcance limitado o distinto a lo que se deseaba o bien caen en un círculo vicioso de retrabajo + retrasos que elevan considerablemente el costo de desarrollo de estos sistemas. Se puede llegar incluso a abandonar el sistema por considerarse incosteable o inalcanzable.

Normalmente un tema de la ingeniería del software tiene diferentes significados. De las muchas definiciones que existen para requisito ha continuación se presentan algunas de las dadas por diferentes autores.

En el glosario de la IEEE se define requisito como:

- a. Una condición o capacidad que un usuario necesita para resolver un problema o lograr un objetivo.
- b. Una condición o capacidad que debe tener un sistema o un componente de un sistema para satisfacer un contrato, norma, especificación u otro documento formal.
- c. Una representación en forma de documento de una condición o capacidad como las expresadas en (a) o en (b).

Sommerville define requisito como “una descripción abstracta de los servicios que se espera que el sistema realice y las restricciones bajo las cuales el mismo debe operar. Solo debe especificar el comportamiento externo del sistema y no debe interesarse por las características de diseño.” (Sommerville, 1992)

SWEBOK Guide de la IEEE Computer Society en su versión de 2004 da también su propia definición de requisito: “En lo más básico, un requisito de software es una propiedad que tiene que ser expuesta para resolver un problema determinado del mundo real. Por lo tanto, es una propiedad que tiene que ser exhibida por un software desarrollado o adaptado para resolver algún problema en particular.” (IEEE Computer Society, 2004)

Analizando las definiciones anteriores, un requisito es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso.

1.2.2. Características de un Requisito

Es importante no perder de vista que un requisito debe ser:

- Especificado por escrito: Como todo contrato o acuerdo entre dos partes.
- Posible de probar o verificar. Si un requisito no se puede comprobar, entonces ¿cómo se sabe si se cumplió con él o no?
- Conciso: Un requisito es conciso si es fácil de leer y entender. Su redacción debe ser simple y clara para aquellos que vayan a consultarlo en un futuro.
- Completo: Un requisito está completo si no necesita ampliar detalles en su redacción, es decir, si se proporciona la información suficiente para su comprensión.
- Consistente: Un requisito es consistente si no es contradictorio con otro requisito.

- No ambiguo: Un requisito no es ambiguo cuando tiene una sola interpretación. El lenguaje usado en su definición, no debe causar confusiones al lector.

1.2.3. Requisitos funcionales y no funcionales.

Los requisitos deben ser escritos de tal forma que sean entendibles sin el conocimiento de notaciones especializadas. Los mismos se dividen en dos categorías:

- a. Requisitos funcionales del sistema.
- b. Requisitos no funcionales.

“Los requisitos funcionales describen las funciones que el software debe ejecutar, son conocidos en ocasiones como capacidades.” (IEEE Computer Society, 2004)

“Los requisitos funcionales son los que definen las funciones que el sistema será capaz de realizar, describen las transformaciones que el sistema realiza sobre las entradas para producir salidas. Es importante que se describa el ¿Qué? y no el ¿Cómo? se deben hacer esas transformaciones. Estos requisitos al tiempo que avanza el proyecto de software se convierten en los algoritmos, la lógica y gran parte del código del sistema.” (Chaves, 2008)

De manera que se podría concluir que los requisitos funcionales son las condiciones que un sistema debe cumplir para producir un resultado.

“Los requisitos no funcionales son los que actúan para limitar la solución. En ocasiones son conocidos como requisitos de restricción o de calidad.” (IEEE Computer Society, 2004)

“Los requisitos no funcionales son las limitaciones impuestas al software, y las restricciones impuestas por el diseñador, deben estar expresadas y en concordancia con los requisitos funcionales.” (Sommerville, 1992)

“Los requisitos no funcionales pueden ser clasificados como requisitos de funcionalidad (pertenencia, interoperabilidad, adherencia), mantenimiento, seguridad, confiabilidad, entre otros.” (IEEE Computer Society, 2004)

De manera que se decir que los requisitos no funcionales especifican propiedades del sistema como restricciones de ambiente y desarrollo, funcionalidad, dependencias de plataformas, mantenimiento y confiabilidad.

1.2.4. La Ingeniería de Requisitos.

La ingeniería de requisitos continúa siendo un tema importante en la industria del software. Más y más organizaciones de desarrollo se dan cuenta que no se puede tener éxito a menos que obtengan correctamente los requisitos de software. A menudo, los responsables de liderar el proceso de requisitos están mal preparados para desarrollar este retador rol. Hacen lo mejor que pueden, pero es como tratar de escalar una colina sin el entrenamiento, los recursos y la experiencia necesaria. (Wiegers, 2006)

El proceso de recopilar, analizar y verificar las necesidades del cliente para un sistema es llamado Ingeniería de Requisitos. La meta de la ingeniería de requisitos (IR) es entregar una especificación de requisitos de software correcta y completa.

A continuación se darán algunas definiciones para ingeniería de requisitos.

“La ingeniería de requisitos de software es la disciplina encargada de establecer los servicios que un sistema debe suministrar, así como, las restricciones bajo las cuales el sistema debe operar.” (Sommerville, 1996)

IEEE la define como:

- Proceso de estudio de las necesidades de los usuarios con el objeto de llegar a una definición del sistema hw/sw.
- El proceso de estudio y refinamiento de un sistema.

Según el profesor Loucopoulos³:

- Trabajo sistemático de desarrollo de requisitos, a través de un proceso iterativo y cooperativo de análisis del problema, documentando los resultados en una variedad de formatos y probando la exactitud del conocimiento adquirido.

El Software Engineering Institute en su glosario de términos define Ingeniería de Requisitos de la siguiente manera: “La ingeniería de Requisitos involucra todas las actividades del ciclo de vida

³ Professor Loucopoulos, sostiene un título de MSc. y Dr. en Computación. Actualmente es profesor de Información de Sistemas en el Departamento de Computación en UMIST (Universidad de Manchester). Autor de 4 libros y editor de 2 volúmenes de procedimientos de conferencias.

dedicadas a la identificación de los requisitos de usuario, análisis de requisitos para derivar requisitos adicionales, documentación de los requisitos como una especificación y validación de los requisitos documentados contra las necesidades del usuario, así como los procesos que soportan estas actividades.” (SEI, 2008)

A manera de resumen se puede decir que el proceso de Ingeniería de Requisitos es un conjunto estructurado de actividades que sirven para derivar, analizar, documentar, validar y mantener los requisitos de un sistema (hardware, software o hardware + software) haciendo uso de procedimientos técnicos y herramientas y en función de las necesidades del usuario.

1.2.5. Actividades de la Ingeniería de requisitos.

La Ingeniería de Requisitos está compuesta por una serie de actividades que guían el proceso durante su desarrollo. A continuación se mencionan algunas de las actividades que forman parte del proceso de desarrollo de requisitos:

- ✓ **Establecimiento de la necesidad del sistema:** Un proyecto de sistema de software, empieza con problemas y oportunidades para mejoramiento del negocio y a menudo las organizaciones deben adoptar el cambio.
- ✓ **Factibilidad del sistema:** Posibilidad de realización del software con el hardware y software existentes. Además, se realiza una estimación del costo y se contrasta con los beneficios que el software brindará a la empresa. Esta actividad no debe realizarse hasta asegurar que se establecieron las necesidades del sistema.
- ✓ **Análisis de requisitos:** Este es el proceso de derivación de requisitos del sistema de software a través de la observación de sistemas existentes, discusión con usuarios y proveedores de información potenciales, análisis de tareas y así sucesivamente. Esto puede involucrar el desarrollo de uno o más modelos diferentes del sistema que ayudan al analista a entender el sistema que será especificado. También pueden desarrollarse prototipos para ayudar a entender los requisitos.
- ✓ **Definición de requisitos:** Es la traducción de información recolectada durante la actividad de análisis a un documento que define el conjunto de requisitos. Este podría reflejar con precisión qué es lo que el cliente desea. Debe escribirse de forma que sea entendible por usuarios finales y clientes del sistema.
- ✓ **Especificación de requisitos:** En esta actividad se detalla el documento de Especificación de Requisitos de Software (ERS). Conjuntamente se puede realizar un diseño de alto nivel para

ayudarse a descubrir errores en definición de requisitos, mismos que deben ser corregidos. (Sommerville, 1996) Las especificaciones forman un puente entre requisitos y diseño. Sus entradas son el documento de requisitos y la información de los usuarios y desarrolladores. La especificación de un sistema de software describe aspectos funcionales y de calidad.

- ✓ **Validación de requisitos:** Esta actividad garantiza que los requisitos obtenidos son correctos, que satisficieran las necesidades del cliente y tienen todas las características de alta calidad requeridas. La validación debe guiar al analista a corregir o refinar los requisitos documentados. (Wiegers, 2006)

1.2.6. Importancia de la Ingeniería de Requisitos.

Los principales beneficios que se obtienen de la Ingeniería de Requisitos son: (Herrera, 2003)

- ✓ Permite gestionar las necesidades del proyecto en forma estructurada: Cada actividad de la IR consiste de una serie de pasos organizados y bien definidos.
- ✓ Mejora la capacidad de predecir cronogramas de proyectos, así como sus resultados: La IR proporciona un punto de partida para controles subsecuentes y actividades de mantenimiento, tales como estimación de costos, tiempo y recursos necesarios.
- ✓ Disminuye los costos y retrasos del proyecto: Muchos estudios han demostrado que reparar errores por un mal desarrollo no descubierto a tiempo, es sumamente caro; especialmente aquellas decisiones tomadas durante la Ingeniería de Requisitos.
- ✓ Mejora la calidad del software: La calidad en el software tiene que ver con cumplir un conjunto de requisitos (funcionalidad, facilidad de uso, confiabilidad, desempeño, etc.).
- ✓ Mejora la comunicación entre equipos: La especificación de requisitos representa una forma de consenso entre clientes y desarrolladores. Si este consenso no ocurre, el proyecto no será exitoso.
- ✓ Evita rechazos de usuarios finales: La ingeniería de requisitos obliga al cliente a considerar sus requisitos cuidadosamente y revisarlos dentro del marco del problema, por lo que se le involucra durante todo el desarrollo del proyecto.

Se puede entonces asegurar, una vez analizado todo lo anteriormente planteado que “la falta de concordancia con los requisitos es una falta de calidad.”Y es precisamente ahí donde radica la importancia de desarrollar una multimedia, que ayude a la capacitación del personal en materia de Ingeniería de Requisitos.

1.2.7. La IR en CMMI y el proceso de mejoras en la UCI.

CMMI constituye un modelo de procesos mejorados de manera incremental. CMMI describe 6 áreas de proceso ingenieriles. Las 2 primeras abordan las actividades de la IR: Desarrollo de Requisitos, Gestión de Requisitos, Solución Técnica, Integración del Producto, Verificación y Validación.

En la UCI se comenzará a partir del mes de junio un proceso de mejoras. El servicio consta de 3 fases, y puede tener una duración de 1 a 3 años. A continuación se detallan cada una de las fases.

FASE I. INICIAR EL PROYECTO DE MEJORA: está destinada a promover y planificar detalladamente el proyecto de mejora de procesos de software, mediante seminarios, talleres además se realiza una evaluación de procesos de software de clase B, una revisión documental, evaluación inicial, elaboración del plan de mejora de procesos de software.

FASE II. SOPORTE EN LA IMPLANTACIÓN: destinada a: coordinar y llevar a cabo la implantación de las acciones de mejora definidas en el plan de mejora de procesos de la organización. La implantación con éxito de la mejora de los procesos requiere que ésta se gestione como un proyecto, con sus roles, responsabilidades, recursos y planes.

FASE III. EVALUACIÓN FORMAL SCAMPI: destinada a realizar una evaluación formal basada en el modelo CMMI utilizando el SCAMPI (Standard CMMI Appraisal Method for Process Improvement).

El proceso solo abarca la Gestión de Requisitos que es un área a evaluar a nivel 2 de CMMI aunque se pretende trabajar en un futuro en el Desarrollo de Requisitos que es a nivel 3.

1.3. Multimedia. Principales conceptos asociados.

El término multimedia es actualmente una palabra común, es muy común leer artículos referentes a las posibilidades que ésta ofrece en los procesos de enseñanza aprendizaje.

Multimedia suele presentarse como el último avance que, propiciado por la evolución y expansión de los medios electrónicos viene a resolver algunos de los problemas que tiene planteada la enseñanza. Sin embargo, el término no resulta nuevo para las ciencias de la educación: el convencimiento de la

importancia de la comunicación multi-sensorial en el proceso didáctico, el principio didáctico de la redundancia y la reflexión que ha acompañado a cada aparición de un nuevo medio, han hecho que si no el término (que también), al menos el concepto sea usual en Tecnología Educativa. (Salinas, 1996)

1.3.1. Definición de Multimedia.

Multimedia, en informática, es la forma de presentar información que emplea una combinación de texto, sonido, imágenes, animación y vídeo. Entre las aplicaciones informáticas multimedia más corrientes figuran juegos, programas de aprendizaje y material de referencia. La mayoría de las aplicaciones multimedia incluyen asociaciones predefinidas conocidas como hipervínculos, que permiten a los usuarios moverse por la información de modo más intuitivo e interactivo. (Microsoft Encarta, 2006)

De manera que podemos resumir multimedia como la combinación de imágenes, texto, videos, audio y gráficos.

El uso de técnicas multimedias permitió el desarrollo del hipertexto, una manera de ligar temas mediante palabras en los textos permitiendo el acceso a temas de interés específico en uno o varios documentos sin tener que leerlos completamente haciendo clic con el mouse en las palabras remarcadas (subrayadas o de un color diferente) que estén relacionadas con lo que buscas.

Dado que esta vinculación interactiva no se limitó a textos solamente, sino que también se puede interactuar con sonidos y animaciones relacionadas con el tema que se está tratando surgió otro concepto: hipermedia, que no es más que el resultado de la fusión de los conceptos de hipertexto y multimedia.

A los sistemas de hipermedias podemos entenderlos como organización de información textual, gráfica y sonora a través de vínculos que crean asociaciones entre información relacionada dentro del sistema.

Actualmente estos términos se confunden e identifican entre sí, de tal forma que al nombrar uno de los conceptos anteriores (hipermedia, hipertexto o multimedia) de forma instintiva y casi automática se piensa en los otros dos. (Díaz Cruz, et al., 2007)

1.4. Tendencias de tecnologías actuales.

El entorno multimedia, entendido como la combinación de texto, gráficos animados, vídeo y sonido, ha evolucionado mucho en estos últimos años. Gracias a los nuevos dispositivos, cualquier intercambio de información, tanto en el mundo empresarial, como en el privado y en el del entretenimiento, se ha enriquecido de forma considerable. Ahora, la información se presenta de una manera mucho más atractiva y sugerente y, por tanto, resulta más fácil de entender y utilizar por cualquier tipo de usuario.

Actividades como la educación o formación a distancia, la videoconferencia, la televisión interactiva, el vídeo bajo demanda, la realidad virtual, las bibliotecas digitales o el trabajo colaborativo, han pasado del nivel textual al nivel multimedia de una forma rápida. Hoy en día, esas actividades no serían ni entendidas ni aceptadas si no incorporasen esas características de combinación de texto, imagen y sonido que permite acceder e interactuar con la información con todos los sentidos.

Hace pocos años el entorno multimedia requería de dispositivos analógicos, pero la evolución de la tecnología digital, abaratando los precios y aumentando la capacidad de intercambio de información, ha permitido una amplia difusión de nuevos tipos de dispositivos, así como del uso de aplicaciones diseñadas para los mismos. Debido a esto, en estos últimos años se ha producido una revolución que cambiará la forma de intercambio de la información en la sociedad.

En la actualidad, escribir documentos en un procesador de textos al tiempo que se escuchan los mensajes del correo electrónico, bajar música en diferentes formatos o visualizar vídeos en la pantalla del ordenador son tareas que se han convertido en rutinarias. Hace algunos años podrían ser consideradas ciencia-ficción o simples elucubraciones teóricas.

En este sentido, la rápida expansión del uso de Internet ha favorecido la utilización de ordenadores y dispositivos con capacidades multimedia, obligando con ello a los proveedores de tecnología a realizar una revolución en sus productos y, por tanto, a dotarlos de todas las características necesarias para poder reproducir imagen y sonido de alta calidad. (COIT, 2008)

El entorno web se ha convertido también en un elemento de soporte para aplicaciones multimedias. En la actualidad, con el progresivo desarrollo del internet, los contenidos multimedias en la web son más accesibles para los usuarios de la red. El uso de Entornos Virtuales de Aprendizaje a dado un salto

gigantesco en el desarrollo del aprendizaje a distancia, aportando como principal ventaja el hecho de estudiar desde cualquier lugar donde se tenga una computadora con acceso a un sistema de este tipo.

1.4.1. Los Entornos Virtuales de Aprendizaje (EVA)

Los Entornos Virtuales de Aprendizaje (EVA) son en la actualidad el arquetipo tecnológico que da sustento funcional a las diversas iniciativas de teleformación. Sin embargo, desde su concepción, diseño y posterior empleo en los procesos de aprendizaje, los EVA deben satisfacer una visión pedagógica que enriquezca su constitución tecnológica inherente. Considerar este requisito puede orientar el uso de estas tecnologías más allá de los usos convencionales como simples máquinas, hacia una en que se contemple al aprendizaje como el principal motivo de su inclusión educativa.

1.4.2. Herramientas EVA.

Varias son las herramientas que sirven de apoyo al desarrollo de los EVA. Estas herramientas, cada una con sus características propias, apoyan en proceso de desarrollo de estos entornos, ya sea que su uso sea privativo o no.

A continuación se relacionan algunas de las herramientas usadas para este tipo de entornos:

- ✓ **WebCT:** (Web Course Tools, o Herramientas para Cursos Web) es un sistema comercial de aprendizaje virtual online, el cual es usado principalmente por instituciones educativas para el aprendizaje a través de Internet. La flexibilidad de las herramientas para el diseño de clases hace este entorno muy atractivo tanto para principiantes como usuarios experimentados en la creación de cursos en línea. Los instructores pueden añadir a sus cursos WebCT varias herramientas interactivas tales como: tableros de discusión o foros, sistemas de correos electrónicos, conversaciones en vivo (chats), contenido en formato de páginas web, archivos PDF entre otros.

WebCT tiene actualmente dos versiones: WebCT Vista y WebCT Campus Edition.

Vista es la versión profesional completa dirigida a empresas, y Campus Edition (Versión Universitaria) es ofrecida a instituciones que ya tienen servicios tales como de sistemas de almacenamiento de archivos y herramientas para registro de cursos.

- ✓ **BlackBoard:** Desarrolla aplicaciones de programas empresariales y servicios relacionados en instituciones educativas (en el 2005 más de 60 países). Las instituciones usan el programa BlackBoard para administrar aprendizaje en línea (e-learning), procesamiento de transacciones, comercio electrónico (e-commerce), y manejo de comunidades en línea (online). Esta empresa desarrolla aplicaciones educativas propietarias y licenció aplicaciones de programas empresariales a más de 2200 instituciones educativas en más de 60 países.

La línea de productos BlackBoard incluye:

- **Blackboard Academic Suite** consiste de:
 - ✓ Blackboard Learning System, un entorno de manejo de cursos.
 - ✓ Blackboard Community System, para comunidades en línea y sistemas de portales.
 - ✓ Blackboard Content System, un sistema para el manejo de contenido.
- ✓ **Moodle:** Es un sistema de gestión de cursos de libre distribución (course management system CMS) que ayuda a los educadores a crear comunidades de aprendizaje en línea.

Un profesor que opera desde este punto de vista crea un ambiente centrado en el estudiante que le ayuda a construir ese conocimiento con base en sus habilidades y conocimientos propios en lugar de simplemente publicar y transmitir la información que se considera que los estudiantes deben conocer.

Moodle promueve una pedagogía constructivista social (colaboración, actividades, reflexión crítica, etc.). Su arquitectura y herramientas son apropiadas para clases en línea, así como también para complementar el aprendizaje presencial. Tiene una interfaz de navegador de tecnología sencilla, ligera, y compatible.

La instalación es sencilla requiriendo una plataforma que soporte PHP y la disponibilidad de una base de datos. Moodle tiene una capa de abstracción de bases de datos por lo que soporta las principales marcas de bases de datos.

Se ha puesto énfasis en una seguridad sólida en toda la plataforma. Todos los formularios son revisados, las cookies cifradas, etc. La mayoría de las áreas de introducción de texto (materiales, mensajes de los foros, entradas de los diarios, etc.) pueden ser editadas usando el editor HTML, tan sencillo como cualquier editor de texto de Windows.

- ✓ **Bodington:** Es un entorno virtual de aprendizaje de código abierto en uso en Universidades y Colegios del mundo. El proyecto Bodington existe para proveer un entorno de código abierto para apoyar el aprendizaje, la enseñanza y la investigación. Es particularmente usado para complejas y multidisciplinarias organizaciones y para colaboración inter institucional. Entrega acceso controlado usando estándares abiertos.

1.4.3. Herramientas para el desarrollo de multimedias.

En la actualidad existen diferentes tipos de herramientas (software) para desarrollar multimedias. Las herramientas de autor son una de ellas.

Herramientas de autor.

Las herramientas de autor son aplicaciones que permiten un trabajo multimedia y constructivista para generar un entorno de aprendizaje dinámico, dentro de las funcionalidades que este tipo de herramientas presentan se puede destacar la posibilidad de crear actividades o pequeñas aplicaciones desde la misma herramienta.

Las herramientas de autor proveen generalmente módulos desde los cuáles se pueden organizar actividades o se puede interconectar pequeños componentes y se pueden adecuar a los objetivos, los conocimientos y habilidades que se busque desarrollar por parte del autor. (Wikipedia, 2008)

De Click2learn:

- ✓ **ToolBook:** Es el nombre de una línea de productos que incluye ToolBook Instructor y ToolBook Assistant ToolBook permite a los usuarios desarrollar cursos de capacitación web, capacitación basada en computadoras (CBT por sus siglas en inglés), cursos, aplicaciones multimedias, simulaciones para software y otros tipos de contenido interactivo. Utiliza un lenguaje de programación propio conocido como OpenScript. Aunque ToolBook es una aplicación muy flexible, su principal desventaja radica en el hecho de tener muy poca compatibilidad y mucha programación.

De Adobe:

- ✓ **AuthorWare:** Es la principal herramienta de autoría visual para la creación de aplicaciones de aprendizaje electrónico para entrega en redes corporativas, CD/DVD y la Web. Desarrolla

aplicaciones accesibles que cumplen con los estándares de sistemas de administración de conocimiento (LMS⁴ por sus siglas en inglés). AuthorWare se basa en la programación por iconos. (Adobe, 2008)

AuthorWare proporciona poca flexibilidad, teniendo que ajustarse explícitamente a los patrones preestablecidos. Sirva un ejemplo. En AuthorWare el método de presentación de textos largos consiste en la paginación, o bien, en el desplazamiento del mismo. Si lo que se desea es el desplazamiento del texto pero con una barra de desplazamiento personalizada, es poco menos que imposible. Tras un increíble esfuerzo trabajando con animaciones, máscaras y otros recursos, sería posible llegar a conseguirlo. Pero si lo que deseamos no es una, sino dos cajas de texto con barras de desplazamiento personalizadas, entonces sí que nos podemos olvidar de AuthorWare.

- ✓ **Adobe Director:** Este software permite generar presentaciones multimedia (en archivos ejecutables, por ejemplo) que pueden ser distribuidas a través de CDs. Permite incorporar a las películas múltiples formatos, como imágenes JPEG, BMP, PNG, GIF... vídeos (MOV, AVI...), sonidos (WAV, AIFF...) o animaciones Flash. Incluye editores básicos para texto, mapa de bits, vectores, sonido. (Adobe, 2008)

Además del potente lenguaje incorporado (Lingo), una de sus principales ventajas esta en el uso de los llamados XTRAS. Se trata de “pequeños programas” desarrollados en lenguaje C++ por otros usuarios o terceras empresas, y que proporcionan al usuario infinidad de utilidades.

Se pueden generar varios tipos de archivos, sin embargo, lo más normal es crear un archivo ejecutable (.EXE) para Windows o Macintosh. De esta forma, puede verse la presentación en cualquier ordenador, sin tener instalado Macromedia Director.

⁴ Learning Management System (LMS). Un LMS es un programa (software) instalado en un servidor, que sirve para administrar, distribuir y controlar las actividades de formación presencial o e-Learning de una organización. Un LMS generalmente no incluye posibilidades de autoría (crear sus propios contenidos), pero se focaliza en gestionar contenidos creados por fuentes diferentes. La labor de crear los contenidos para los cursos se desarrolla mediante un LCMS (Learning Content Management Systems).

Como principal desventaja de esta herramienta, se presenta el hecho de no poder ser visualizado en una plataforma con sistema operativo Linux instalado, por lo que los usuarios de este tipo de sistema no tendrían acceso al contenido hecho en esta herramienta.

- ✓ **Flash:** El software Macromedia Flash es el entorno de creación más avanzado para generar contenido interactivo de alta calidad para plataformas móviles, digitales y web.

Es una aplicación en forma de estudio que trabaja sobre un "escenario" y sobre "Fotogramas" destinado a la producción de animación.

Los archivos de Flash aparecen muy a menudo como animaciones en páginas Web y sitios Web multimedia. (Adobe, 2008)

En un principio esta fue una tecnología más, con la necesidad de instalar un plug-in para poder visualizarse.

Cualquier webmaster es recio a incorporar a una página web contenidos para los que sea necesario utilizar plug-ins, pues es de suponer que no todos los tendrán y que se va a marginar a una serie de usuarios. Con el tiempo, flash se ha convertido en un estándar, viene en la instalación básica de los exploradores, se instala automáticamente si el navegador no lo tiene y muchas páginas lo utilizan. Como resultado tenemos que los webmasters o diseñadores del Web lo vienen utilizando para crear todo tipo de efectos, incluso en la home page o diseñando toda la página con Flash.

La clave de Flash es que es un programa de animación vectorial. Esto significa que se pueden crear animaciones complejas: aumentar y reducir elementos de la animación, mover de posición estos objetos, y otras cosas sin que la animación ocupe mucho espacio en el disco. Los vectores con los que trabaja Flash sólo son, por decirlo de alguna manera, siluetas que casi no ocupan espacio y se pueden modificar fácilmente y sin gasto de memoria en disco.

Algunas características fundamentales de Flash es que es un sistema multiplataforma que se puede ejecutar en cualquier sistema operativo. Por medio de un plug-in que se instala en cualquier navegador, se pueden visualizar películas Flash. Según Macromedia (ahora Adobe), empresa creadora de Flash, más del 99% de los usuarios que acceden a las webs, con navegadores de cualquier sistema operativo, soportan la tecnología Flash.

Macromedia Flash cuenta con un editor de gráficos vectoriales, puede importar ficheros con extensión BMP, DIB, EPS, GIF, HTML, JPEG, LRG, Photoshop6 (PSD), SWF, TARGA, TIF, WMF, y gráficos PhotoCD.

Macromedia Flash utiliza como lenguaje de programación ActionScript, un lenguaje de programación orientado a objetos y usado en aplicaciones web animadas y en aplicaciones de Macromedia (ahora Adobe).

Herramientas libres.

Varias son las herramientas libres que existen y su funcionalidad es cada día mayor. Algunas de las herramientas que existen en la actualidad se pueden encontrar en internet. Por ser muchas de éstas herramientas de nueva concepción, no es mucha la documentación que se encuentra sobre las mismas en internet, lo que de alguna forma atenta contra la inclusión de alguna de éstas en la implementación del producto final.

A continuación se describen algunas de las herramientas libres existentes.

MTASC: MTASC (Motion Twin Action Script Compiler) es un compilador open source para ActionScript 2.0 capaz de compilar 100 clases en menos de 5 segundos y es más estricto en muchos aspectos en los que el compilador de Flash no le da importancia.

Todo comenzó con un compilador muy restrictivo, que no aceptaba muchas peculiaridades del compilador de Flash, con lo que la gente exigía más flexibilidad para compilar sus proyectos. Poco a poco se ha avanzado en el proyecto y todo parece indicar que se puede dejar de depender del IDE de Flash para desarrollar sus aplicaciones con ActionScript.

MTASC es independiente de plataforma, se puede compilar para la plataforma que se desee, lo único que se necesita es poder compilar el lenguaje y compilador de Ocalm.

El compilador funciona desde la línea de comandos y por lo tanto obliga a trabajar con Programación Orientada a Objetos (POO), o sea los proyectos tienen que ser un conjunto de clases. Existen dos variantes, se puede escoger trabajar con POO totalmente y exclusivamente con MTASC y la otra es parcialmente con el IDE de Flash y compilar las clases con MTASC.

Si se escoge trabajar totalmente con POO, desarrollar una aplicación desde Linux no se logra ver como hacerlo sin depender del IDE de Flash, incluso aunque solo fuese para instanciar la primera clase, sin embargo se puede trabajar con un sistema de clases en la que una de ellas se auto ejecute al empezar la aplicación.

UIRA: Las alternativas a Adobe Flash en Linux son muy escasas y debido a este déficit de variantes se decidió crear un proyecto propio para ampliar estas tecnologías.

Partiendo del código de Flash4linux (F4l) se fueron rescribiendo varios módulos para solucionar diversos bugs y añadirles más funcionalidades. F4l siempre ha tenido problemas a la hora de crear Flash, se pueden hacer animaciones pero todavía tiene grandes dificultades para generar un archivo swf.

Pasaron unos seis meses y partiendo de ese código se obtuvo QFlash, el mismo puede crear animaciones básicas, añadir textos, polígonos, formas básicas, cuadro de texto, etc.

Entre agosto y septiembre del año 2005 Florian Delizy se puso en contacto con ambos proyectos: F4l y Qflash. Se llegó a un acuerdo general y decidieron crear un nuevo proyecto, y de esta forma surgió UIRA. Se decidió cambiar el nombre por tres razones fundamentales:

1. Los dos proyectos anteriores contienen en su nombre flash y los propietarios no querían problemas de copyright.
2. Flash4linux no contaba con un gran alcance, a sus desarrolladores les interesaba otras plataformas como Mac y Windows.
3. Flash4linux se quedaba corto porque no solo se quería crear flash.

Se decidió poner el nombre de UIRA. Actualmente el proyecto está en pleno desarrollo, por ahora solo se está trabajando en buscar un diseño óptimo que evite que se tenga que deshacer el trabajo por mal diseño. Aún no puede crear flash ya que se está programando librerías básicas, sistemas de plugins y preparando diagramas UML.

La parte gráfica se desarrolla con las librerías Qt4 y también usa el motor gráfico Amanith. Para el editor ActionScript se usa las librerías scintilla, actualmente este proyecto trabaja en conjunto con Ktoon⁵ para compartir parte del código.

1.4.4. El lenguaje de marcas XML.

El lenguaje XML (Extensible Markup Language por sus siglas en inglés), lenguaje de marcas extensible, es un metalenguaje extensible de etiquetas desarrollado por el W3C (World Wide Web Consortium). Es una simplificación y adaptación del lenguaje SGML⁶ y permite definir la gramática de lenguajes específicos. (W3C, 2008)

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

Entre sus principales ventajas se destacan: (Wikipedia, 2008) (Montero Ayala, 2001)

- ✓ Es extensible, lo que significa que una vez diseñado un lenguaje y puesto en producción, igual es posible extenderlo con la adición de nuevas etiquetas de manera que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato.
- ✓ El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles. De esta manera se evitan bugs y se acelera el desarrollo de la aplicación.
- ✓ Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.

1.4.5. Lenguaje de programación. ActionScript 2.0.

Un lenguaje de programación es una construcción mental del ser humano para expresar programas. Está constituido por un grupo de reglas gramaticales, un grupo de símbolos utilizables, un grupo de

⁵ Ktoon es una herramienta para desarrollar animaciones 2D realizada por animadores de Tonnka Films.

⁶ SGML son las siglas de Standard Generalized Markup Language o Lenguaje de Marcación Generalizado. Consiste en un sistema para la organización y etiquetado de documentos. ISO ha normalizado este lenguaje en 1986. Este lenguaje no impone en sí ningún conjunto de etiquetas en especial.

términos mono sémicos (es decir, con sentido único) y una regla principal que resume las demás. Para que ésta construcción mental sea operable en un computador debe existir otro programa que controle la validez o no de lo escrito. (Enciclopedia, 2008)

Existen diferentes tipos de lenguajes, entre ellos podemos nombrar los lenguajes de bajo nivel, los de medio nivel, los de alto y los scripts. Dentro de este último se ubica ActionScript, uno de los lenguajes de programación orientado a objetos utilizado en especial en aplicaciones web animadas en el entorno flash.

ActionScript 2.0.

El ActionScript es el lenguaje de programación que ha utilizado Macromedia Flash desde sus comienzos. A grandes rasgos, podemos decir que el ActionScript nos permitirá realizar con Flash todo lo que se propongan, ya que da el control absoluto de todo lo que rodea a una película Flash. El ActionScript es, como su nombre indica, un lenguaje de script, esto quiere decir que no hará falta crear un programa completo para conseguir resultados, normalmente la aplicación de fragmentos de código ActionScript a los objetos existentes en nuestras películas nos permiten alcanzar nuestros objetivos.

Es un lenguaje de programación orientado a objetos, tiene similitudes 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.

1.5. Metodología de desarrollo y Lenguaje de Modelado.

1.5.1. RUP.

El Proceso Unificado de Rational (RUP), es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. RUP permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual.

Esto permite a los grupos de desarrollo producir aplicaciones informáticas más robustas y flexibles que se adaptan a las necesidades de los usuarios. La correcta aplicación de RUP permite reducir los tiempos de desarrollo, aumentar la calidad de las aplicaciones y disminuir los costes de mantenimiento. Está basado en componentes, lo cuál quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas.

1.5.2. OMMMA-L (Extensión UML para multimedia.)

El Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA - L) se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario siendo este un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. (Sauer, et al.)

Extendiendo el paradigma MVC para multimedia a las peculiaridades de comportamiento estático y dinámico identificadas anteriormente, obtenemos MVCMM (Modelo Vista Controlador para Multimedia), sobre el que se basa las especificaciones de OMMMA – L.

OMMMA-L no presenta cambios con respecto a UML en el flujo de requisitos y casos de uso. Integra dos nuevos diagramas, el mapa de navegación que le da una idea al usuario de cómo será la navegación en la multimedia y el diagrama de presentación; que sirve para describir la parte estática del modelo a través de una descripción intuitiva de la distribución espacial de objetos visuales de la interfaz de usuario.

1.6. Herramientas existentes para el estudio del tema.

Durante la búsqueda del tema específico el autor no encontró herramientas que se dediquen al proceso de enseñanza del tema en cuestión específicamente. Sin embargo, hay desarrolladas algunas herramientas para el estudio de la Ingeniería del Software de forma general.

1.6.1. Problems and Programmers y SimSE

En Irvine, Universidad de California, Estados Unidos, un pequeño grupo de desarrolladores crearon dos herramientas en forma de juego para la enseñanza de Ingeniería del Software, de modo que el juego simulara procesos dándole la oportunidad al estudiante de practicar como si fuera un proyecto real. El primer juego se llama Problems and Programmers y es un software educativo a través de tarjetas y el segundo; SimSE es un juego de simulación de procesos de software, opera un solo jugador quien asume el rol de administrador de proyecto de un equipo de desarrolladores. Al tiempo que el jugador gestiona el proceso para completar un proyecto de ingeniería de software, puede además contratar o despedir empleados, asignar tareas a estos, monitorear su progreso y comprar

herramientas. SimSE permite a los estudiantes practicar un proceso de desarrollo de software (o subproceso) virtual en un ambiente gráfico interactivo. (Oh Navarro, et al.) Estos juegos se proponen dar un acercamiento a lo que se van a enfrentar realmente los estudiantes una vez que salgan al mundo real.

1.6.2. CONTROLA

En la Universidad Federal de Viçosa en Brasil, un grupo de desarrolladores han creado una herramienta de apoyo al proceso de desarrollo de software para las pequeñas compañías, con enfoque en la Administración de Requisitos. La herramienta es usada, tanto por pequeñas compañías como por estudiantes de graduación, posgrado y maestros. (Fraga Filho, 2008)

1.6.3. Otros.

Universidad de los Lagos

En la Universidad de los Lagos, Chile, se estudia el tema de Ingeniería de Software a través de un EVA montado sobre Moodle. El curso enseña de manera general los temas fundamentales de la Ingeniería del Software, el tema de requisitos se imparte en la sección de contenido Introducción a la Ingeniería del Software, donde se estudia un tema de Análisis de Requisitos. (ULA, 2008)

Universidad de Ciencias Informáticas (UCI)

El estudio del tema en la UCI se hace a través de la plataforma Moodle. El curso esta estructurado de manera que se estudia el tema por flujos de trabajo usando las disciplinas de RUP; sin embargo aunque la universidad esta directamente vinculada a la producción de software y que los problemas de gestión de requisitos son reales, el curso no abarca el tema lo suficientemente bien, como para contribuir a capacitar de manera satisfactoria a los estudiantes en cuestión debido a que la documentación se presenta de una forma en que el estudiante lee los materiales disponibles para el tema y luego participa en seminarios o laboratorios y se evalúa. No hay un curso específico para el tema de Ingeniería de Requisitos, sino que se estudia como un tema más dentro de la disciplina de Ingeniería del Software.

1.7. Herramienta a utilizar.

Dada las ventajas que ofrece, ya sea por su estandarización como por su facilidad de uso en diferentes entornos, incluyendo la WEB se decide usar para la confección del curso la herramienta Macromedia

Flash 8 con la utilización del lenguaje ActionScript como lenguaje de programación para Macromedia flash.

Independientemente de ser un software propietario hay que reconocer las facilidades que ofrece tanto de migración a otros entornos como de fusión con otras herramientas de autor como es el caso de la herramienta de autor Mediator.

Como metodología de desarrollo a usar RUP, una potente metodología a nivel mundial y que permite interacción con el usuario durante el proceso de desarrollo de software. Como lenguaje de modelado OMMMA-L, extensión del lenguaje UML para el desarrollo de multimedias y que solamente tiene modificaciones en el flujo de trabajo de análisis y diseño con la aparición de dos nuevos diagramas; los mapas de navegación y los diagramas de presentación.

Consideraciones del capítulo.

En este capítulo se hizo un estudio de las principales tendencias tecnológicas en el desarrollo de multimedias así como los lenguajes usados en el desarrollo de la misma, tanto lenguajes de modelado, como de marcado y lenguaje de script y se tomó como herramienta de desarrollo Macromedia Flash 8, como lenguaje de programación ActionScript 2.0 y XML como lenguaje de marcas para el manejo de los datos de la aplicación. Se dieron algunos conceptos esenciales en materia de Requisitos y de calidad de Software. Se analizó además algunas de las herramientas que a nivel mundial se utilizan para temas relacionados con Ingeniería de Software y se llegó a la conclusión de que no existe ninguna relacionada directamente con la Ingeniería de Requisitos. Se demostró la importancia que tiene la Ingeniería de Requisitos en el desarrollo de sistemas de software y se propuso como metodología de desarrollo a RUP y como lenguaje de modelado a usar OMMMA-L.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN

2.1. Especificación del contenido.

La multimedia contará con tres temas fundamentales que abarcan el contenido de todo el curso, apoyándose en el uso de mapas conceptuales a modo de hacer más claro y entendibles los conceptos de la disciplina.

A continuación se muestra una descripción de cómo estarán distribuidos los contenidos en la multimedia.

✓ Tema 1: Introducción a la Ingeniería de Requisitos (IR):

En este tema se van a tratar 4 puntos fundamentales que abordaran conceptos esenciales en materia de requisitos. Un primer punto se dedicara a abordar los conceptos de ingeniería de requisitos, qué es, importancia de la misma, etc. Un segundo punto se centrará en definir claramente el concepto de requisitos de software y todo lo relacionado éste. Un tercer y cuarto epígrafe abordará los diferentes tipos de requisitos funcionales y no funcionales y la clasificación de éstos últimos. Un último punto de centrará en el tratamiento de requisitos en las diferentes metodologías, estándares y normas internacionales.

✓ Tema 2: Desarrollo de Requisitos.

En el tema de Desarrollo de Requisitos (DR) se abarcará 3 puntos esenciales; un primer punto se centrará en analizar las diferentes actividades en que se divide el DR con la respectiva explicación de cada una de ellas, y un segundo y tercer punto destinado a las técnicas y herramientas respectivamente, el mapa conceptual formara un elemento indisoluble de cada tema a modo de detallar los conceptos elementales por los que transita el DR.

✓ Tema 3: Gestión de Requisitos

El tema de Gestión de Requisitos (GR) se dividirá en tres puntos, un primer punto tratará el concepto de GR, un segundo y tercer punto a desarrollar tratan las actividades que se realizan en el proceso y las herramientas del mismo respectivamente. Un mapa conceptual del proceso ayudará al estudio del tema.

2.2. Descripción del modelo del dominio.

Debido a que los procesos del negocio tienen poca estructuración se plantea un modelo de dominio, de manera que ayude mejor a la comprensión de los conceptos del sistema. Este modelo se hace a través de un diagrama de clases UML (ver Figura 3) en el que se definen las principales clases conceptuales que intervienen en el sistema.

A continuación se presentan algunos conceptos utilizados en el diagrama del modelo de dominio.

- **Usuario:** Es toda persona que interactúe con el sistema.
- **Temas:** Se le denomina temas al contenido del curso que está dividido en 3 tópicos fundamentales, Introducción a la Ingeniería de Requisitos, Desarrollo de requisitos, Gestión de Requisitos.
- Se le llamará **Mapas Conceptuales** a la forma de representar visualmente relaciones significativas entre conceptos.
- Los **Juegos** representan objetos que ejercitan de una manera didáctica los conocimientos adquiridos, aquí podemos encontrar el clásico juego del ahorcado, para el mismo se usaron palabras representativas de la disciplina.
- Las **Autoevaluaciones** representan objetos que comprueban los conocimientos adquiridos en un tema o en el curso de manera general, solo se podrá realizar una vez que se hayan vencido los objetivos de un tema específico.
- La **Bibliografía complementaria** es el objeto que agrupa las diferentes fuentes bibliográficas sobre la que se confeccionó la multimedia o que se vayan a usar durante el curso, ya sean Libros Digitales o sitios de Internet.

2.3. Diagrama de clases del modelo de dominio.

El diagrama de clases del modelo de dominio (Figura 3) define las principales clases conceptuales que intervienen en el sistema

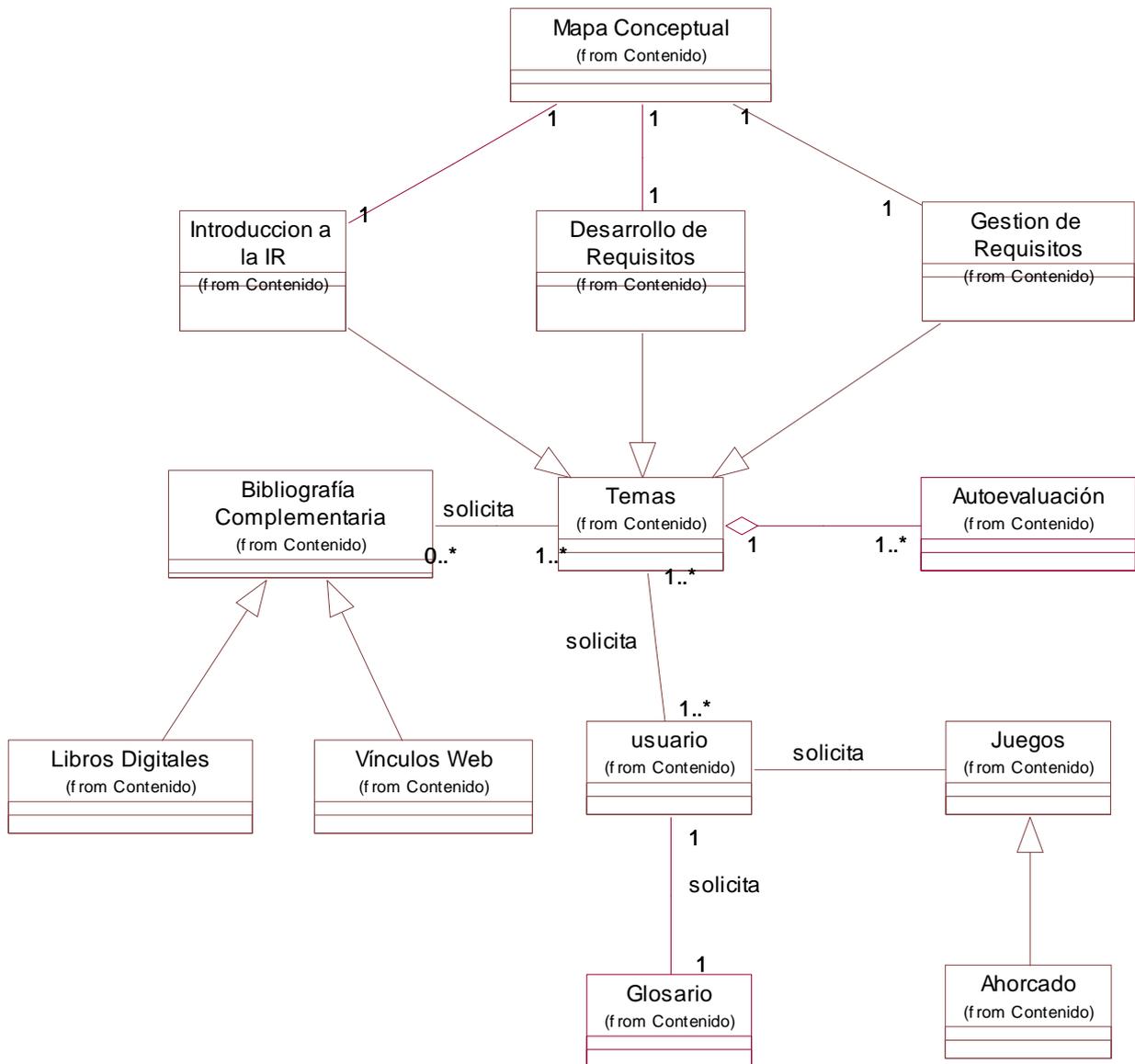


Figura 3: Diagrama de clases del modelo de dominio.

2.4. Mapa de navegación.

El mapa de navegación (Figura 4) es un diagrama que expresa la forma en que nos podemos desplazar entre los diferentes módulos de la aplicación, de manera tal que da una panorámica real del desplazamiento dentro de la misma.

El mapa se ha dividido en sub modelos de navegación. (Figura 5), (Figura 6), (Figura 7), (Figura 8).

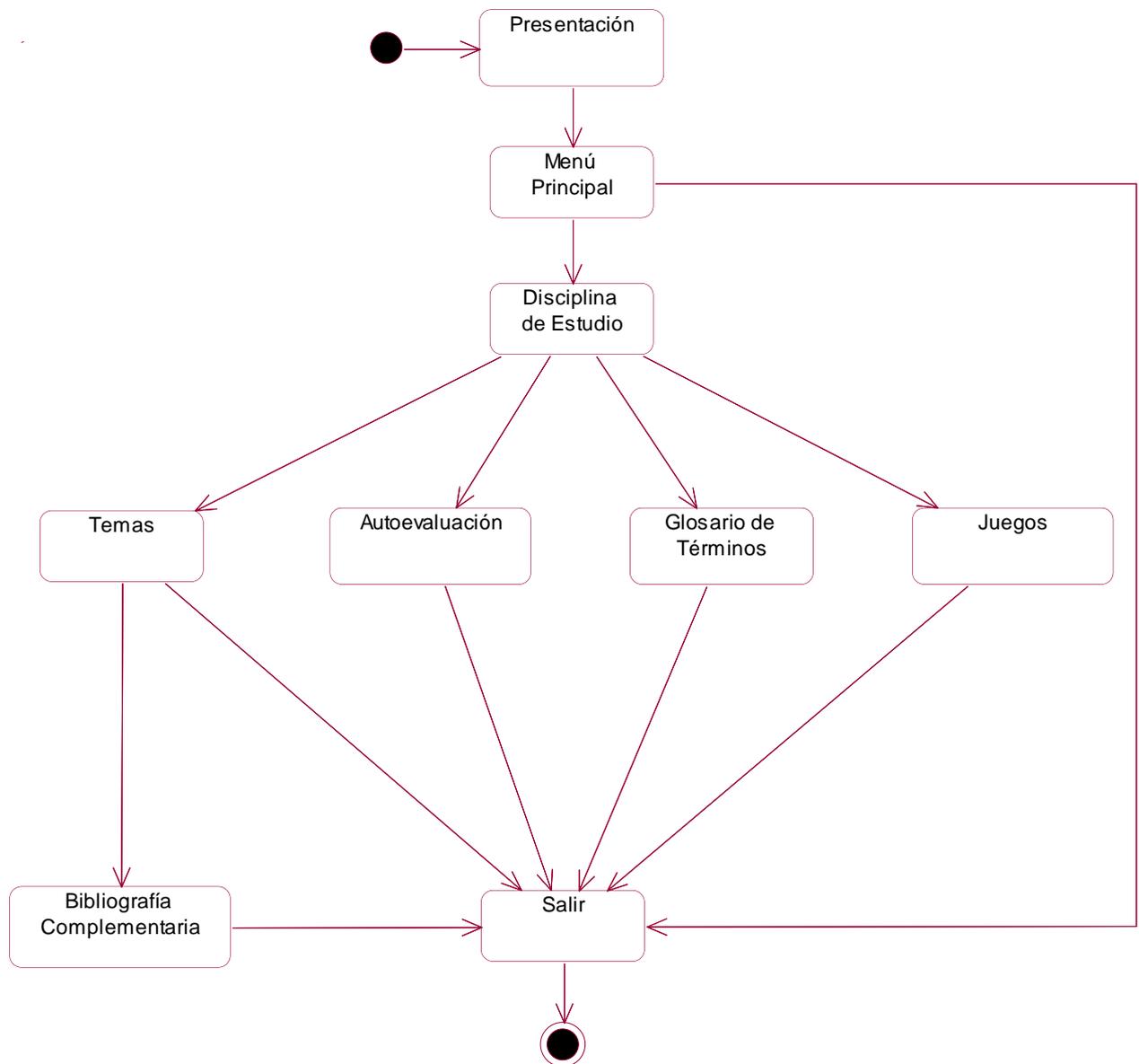


Figura 4: Mapa de navegación general.

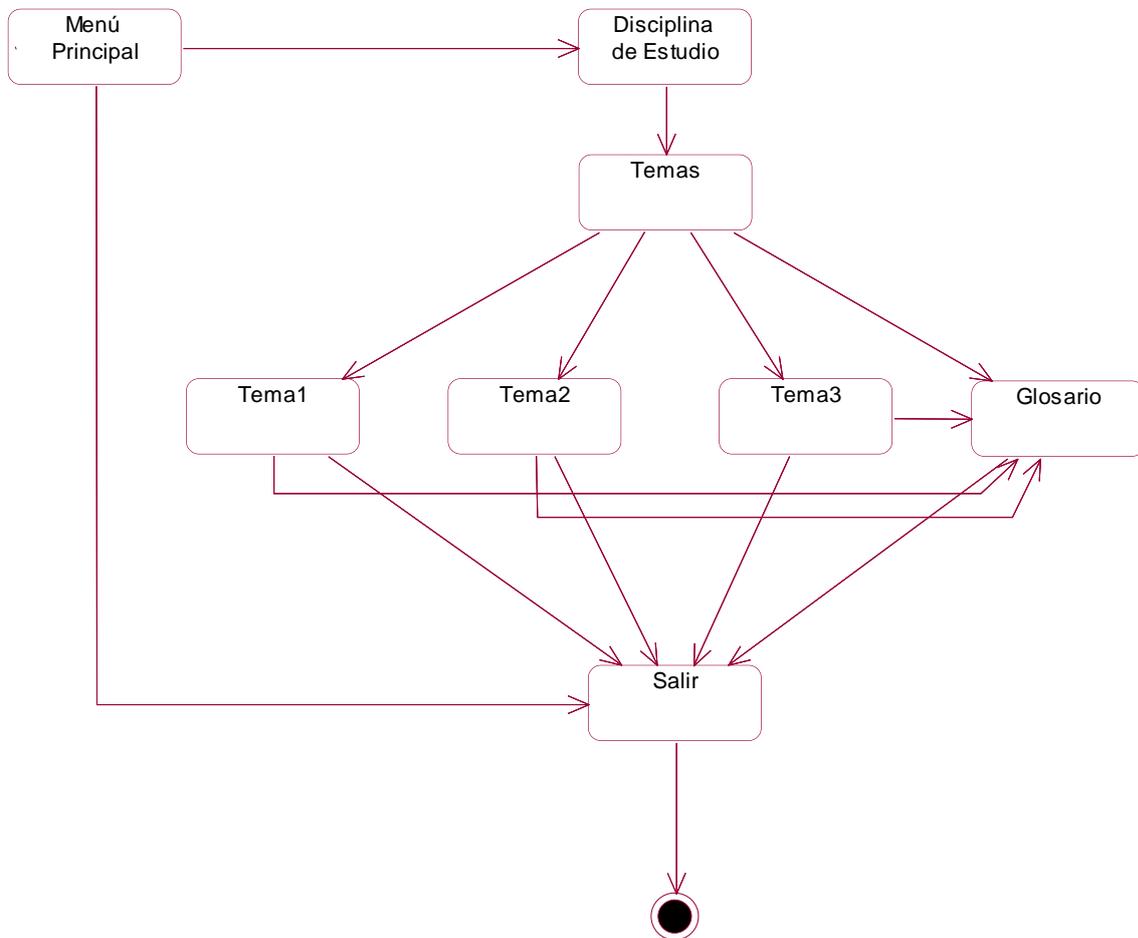


Figura 5: Sub modelo de Navegación Temas.

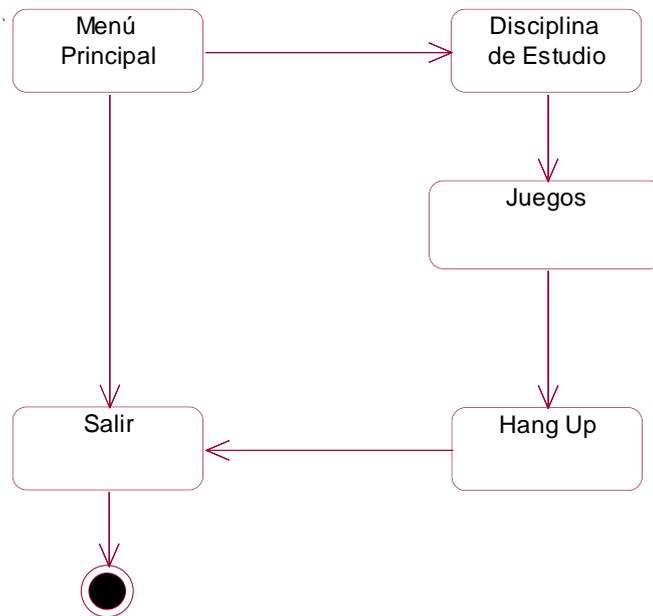


Figura 6: Sub modelo de navegación Juegos.

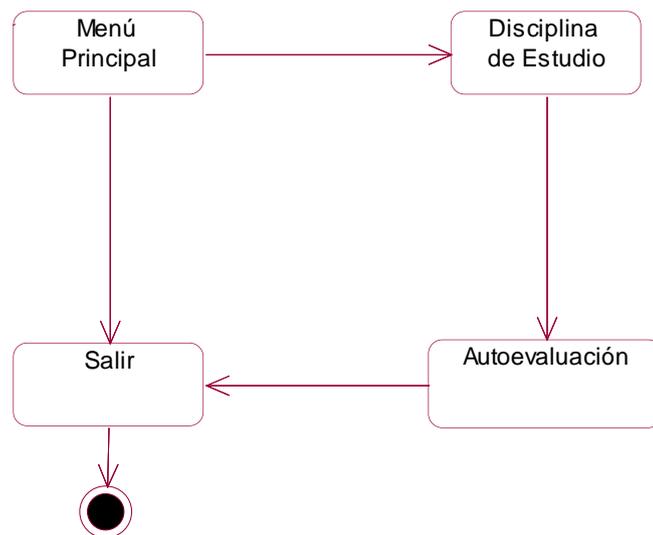


Figura 7: sub modelo de Navegación Autoevaluación.

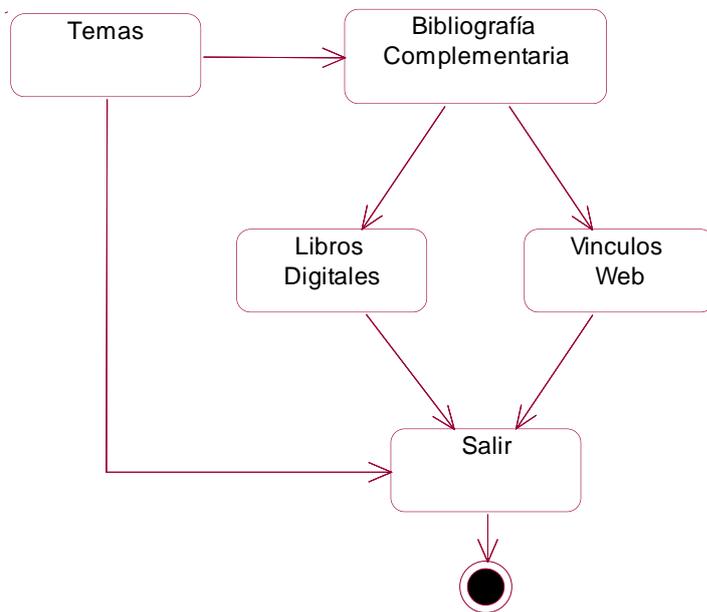


Figura 8: Sub modelo de navegación Bibliografía complementaria

2.5. Requisitos funcionales del sistema. (Tabla 1)

Tabla 1: Requisitos Funcionales del sistema.

Referencia	Función
R1	Mostrar presentación de la aplicación.
R2	Mostrar los temas de un curso seleccionando a partir de los cursos disponibles.
R3	Mostrar los contenidos de un tema.
R4	Ejecutar un juego seleccionando a partir de los juegos disponibles.
R5	Mostrar bibliografía complementaria.
R6	Mostrar términos seleccionado a partir de un glosario
R7	Mostrar el glosario de términos.
R8	Mostrar el contenido del epigrafe.
R9	Ejecutar ejercicios de autoevaluación.
R10	Mostrar el contenido del epígrafe a estudiar.
R11	Mostrar un mapa conceptual para cada tema del curso, posibilitando mostrar el concepto de un término del mapa.
R12	Permitir salir de la aplicación en cualquier momento deseado.

R13	Mostrar créditos al salir de la aplicación.
-----	---

2.6. Requisitos no funcionales.

2.6.1. Requisitos de Apariencia.

- Utilizar botones que expresen su función ya sea que se intuya o expresados con texto.
- Utilizar un diseño de colores que no sobrepase el limite de lo normal, usar colores pasteles en el diseño de la interfaz.
- La aplicación debe utilizar como idioma principal el español excepto aquellas palabras técnicas que no puedan ser traducidas.
- La opción salir de la aplicación estará disponible desde cualquier parte de la aplicación, haciendo clic sobre ella se saldrá de la misma.

2.6.2. Requisitos de usabilidad.

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

2.6.3. Requisitos de software.

Para la ejecución del software en las diferentes plataformas se muestran las siguientes especificaciones en cuanto a sistemas operativos y navegadores (Tabla 2).

Tabla 2: Sistemas Operativos y navegadores.

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

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

2.6.4. 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 256 de RAM. Pantalla de 16 bits de 1024 x 768 (se recomienda de 32 bits) y 710 MB de espacio en disco disponible.

2.6.5. Requisitos de diseño e implementación.

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

2.6.6. Requisitos de soporte.

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

2.7. Modelado de casos de uso del sistema.

El modelado de casos de uso del sistema es confeccionado a partir de los requisitos funcionales del mismo y explica el funcionamiento del sistema o cómo es que el usuario desea que funcione el mismo.

Se definió para ello quiénes van a ser los actores que van a interactuar con el sistema y cuáles son los casos de uso que van a representar las funcionalidades de éste.

2.8. Modelo de casos de uso del sistema.

Los diagramas de Casos de Uso (Figura 9) son una representación gráfica de actores y casos de uso del sistema, incluyendo sus interacciones. Un diagrama de casos de uso muestra, por tanto, los distintos requisitos funcionales que se esperan de una aplicación y como se relaciona con su entorno.

2.8.1. Los actores del sistema. Justificación.

Tabla 3: Actores del sistema y justificación.

Actor	Justificación
Usuario	Es la persona que va a usar el sistema para buscar información sobre curso de Ingeniería de Requisitos.

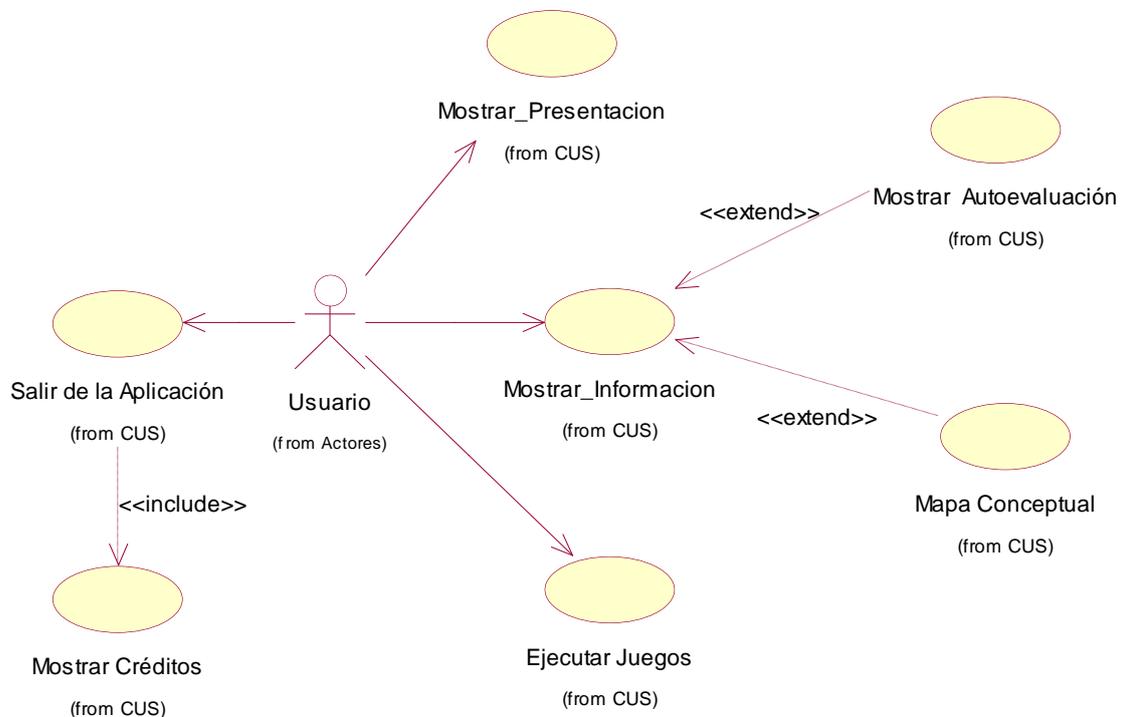


Figura 9: Diagrama de Casos de Uso del sistema.

2.8.2. Breve resumen de los casos de uso del sistema.

Para entender la esencia de cómo funciona cada uno de los casos de uso del sistema, se hace un pequeño resumen (Tabla 3-9) que describe la funcionalidad del caso de uso (CU) y el requisito al que da respuesta.

Tabla 4: Resumen del CU Mostrar Presentación.

Caso de Uso:	Mostrar Presentación
Actor(es):	Usuario
Propósito:	Mostrar la presentación de la multimedia, así como entrar a un curso.
Resumen:	El usuario de la multimedia puede desactivar la presentación si lo desea.
Referencias:	R1

Tabla 5: Resumen del CU Mostrar información.

Caso de Uso:	Mostrar Información
Actor(es):	Usuario
Propósito:	Mostrar la información contenida en la multimedia, ya sea mapas conceptuales, texto, glosario, etc.
Resumen:	El usuario de la multimedia puede seleccionar el tipo de información que desea ver. Considera que debe pasar a ver los mapas conceptuales, selecciona el botón de mapa conceptual del tema y ejecuta esa acción, si desea autoevaluarse va a la opción de autoevaluación y ejecuta la acción también.
Referencias:	R2, R3, R5, R6, R7, R8, R10, R11

Tabla 6: Resumen del CU Mostrar Autoevaluaciones.

Caso de Uso:	Mostrar Autoevaluaciones.
Actor(es):	Usuario
Propósito:	Mostrar y ejecutar los ejercicios de autoevaluación de los temas estudiados en el curso.
Resumen:	El usuario ejecutará la autoevaluación e irá respondiendo las preguntas que le formule la aplicación. Al final la aplicación le dará la cantidad de respuestas

	correctas e incorrectas y el porcentaje obtenido en la misma.
Referencias:	R9

Tabla 7: Resumen del CU Ejecutar Juegos.

Caso de Uso:	Ejecutar Juego
Actor(es):	Usuario
Propósito:	Ejecutar el o los juegos disponibles en el curso.
Resumen:	Una vez que el usuario decide que conoce los términos asociados a la disciplina que esta estudiando puede dirigirse a la sección de juegos y seleccionar el juego en cuestión. Cada vez que adivina una palabra de los términos asociados se le dará un punto por concepto de palabra correcta y menos un punto por concepto de palabra no acertada. Cuando el usuario no desee seguir cerrará el juego e ira a cualquier otra parte de la multimedia que desee.
Referencias:	R4

Tabla 8: Resumen del CU Mostrar Mapa Conceptual.

Caso de Uso:	Mostrar Mapa Conceptual
Actor(es):	Usuario
Propósito:	Mostrar los mapas conceptuales que contienen los principales conceptos de la disciplina.
Resumen:	El usuario puede navegar en el mapa conceptual y desde conceptos de éste ir hasta la explicación del mismo. El usuario puede elegir cualquier mapa conceptual sin necesidad de haber estudiado un tema específico para ello.
Referencias:	R11

Tabla 9: Resumen del Caso de Uso Salir de la Aplicación.

Caso de Uso:	Salir de la Aplicación
Actor(es):	Usuario
Propósito:	Permitir salir de la aplicación en cualquier momento deseado y desde cualquier

	lugar de la multimedia.
Resumen:	El usuario de la multimedia puede seleccionar salir de la aplicación en el momento que lo desee. Cuando el usuario seleccione el botón de salir, la aplicación le mostrará un mensaje de confirmación y el usuario puede confirmar o retractarse de salir. Una vez confirmada la salida, se mostraran los créditos.
Referencias:	R12

Tabla 10: Resumen del Caso de uso Mostrar Créditos.

Caso de Uso:	Mostrar Créditos
Actor(es):	Usuario
Propósito:	Mostrar los créditos de la aplicación una vez que se decida salir de la misma.
Resumen:	Una vez que el usuario de la multimedia decide salir de la aplicación, el sistema mostrará los créditos de la misma, mostrando el nombre de autor, y demás datos necesarios.
Referencias:	R13

2.9. Mapas conceptuales.

Los mapas conceptuales⁷ (Figura 10-16) son una técnica de representación gráfica del conocimiento. Un mapa conceptual es una red de conceptos en la cual los nodos representan los conceptos y los enlaces las relaciones entre estos conceptos en forma de líneas o flechas que van etiquetadas. (Wikipedia, 2008).

El principal aporte de esta investigación radica en la confección de los mapas conceptuales que abarcan de manera sintetizada los principales conceptos de la disciplina y que van a ayudar a la comprensión de conceptos claves en la enseñanza de la disciplina de Ingeniería de Requisitos.

A continuación se muestran los mapas conceptuales confeccionados a partir del tema Ingeniería de Requisitos y que definen sus principales conceptos, así como sus actividades técnicas y herramientas. No se hará mucho detenimiento en ellos ya que cada uno de estos conceptos queda definido en la multimedia.

⁷ El origen de los mapas conceptuales se debe a los trabajos de J. D. Novak (1982) sobre psicología del aprendizaje realizados en la década de los años 60. J. D. Novak estudiaba el desarrollo cognitivo en niños y creo los mapas conceptuales como instrumento de medición para determinar el grado de desarrollo y el tipo de aprendizaje alcanzado por un determinado individuo en un momento dado

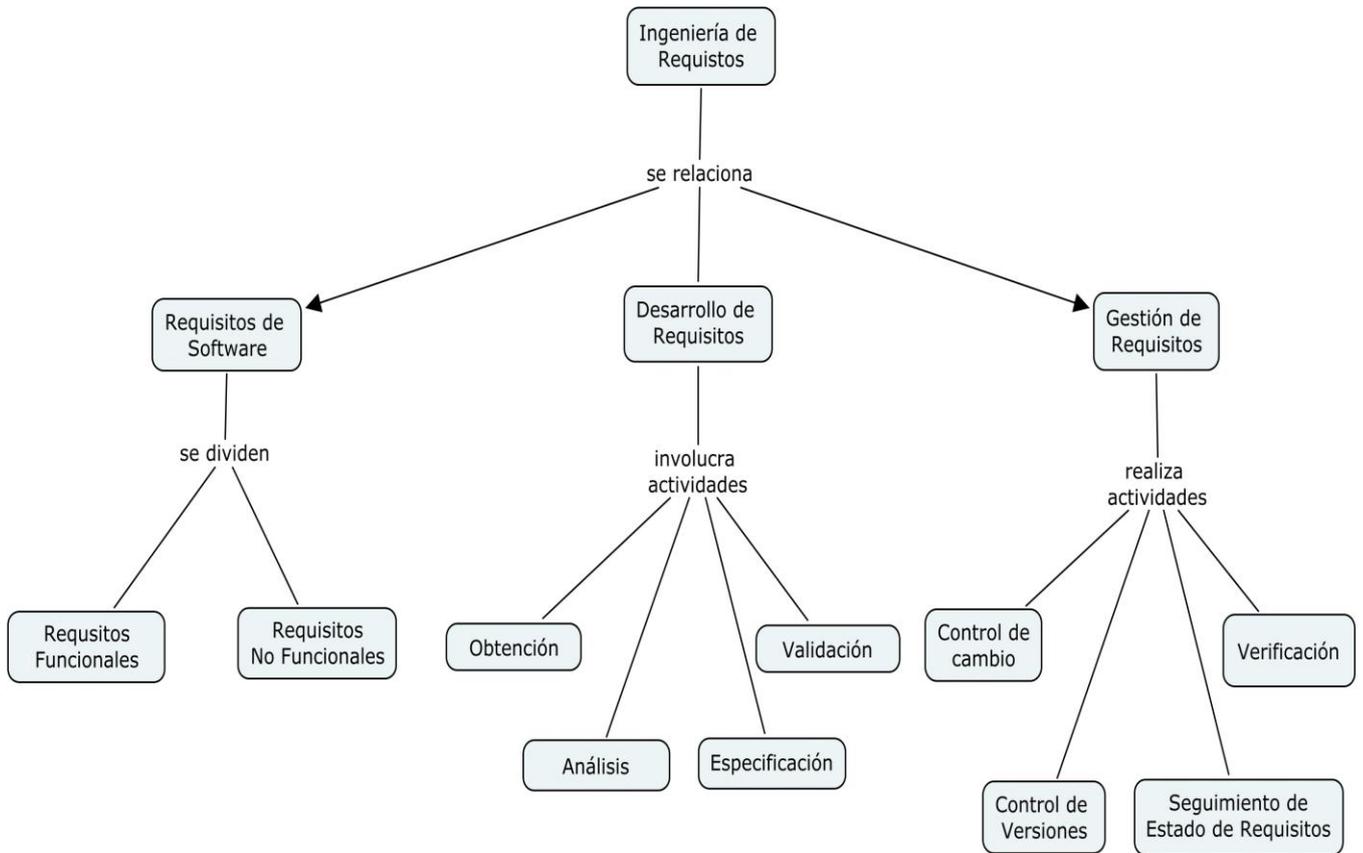


Figura 10: Representación del Mapa General de conceptos relacionados con la IR.

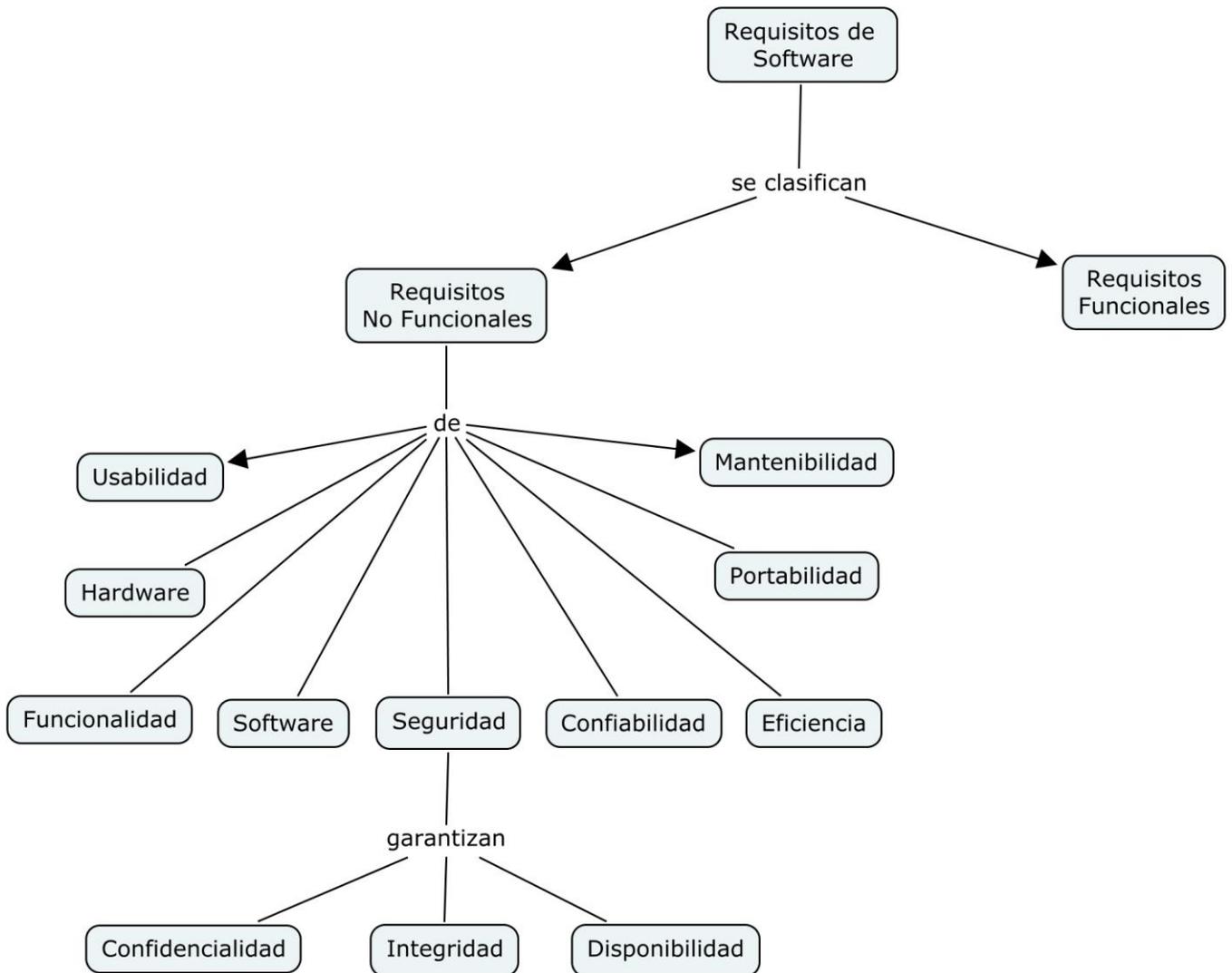


Figura 11: Representación del Mapa que agrupa los conceptos relacionados con los requisitos y su clasificación.

Sub mapas del Desarrollo de Requisitos.

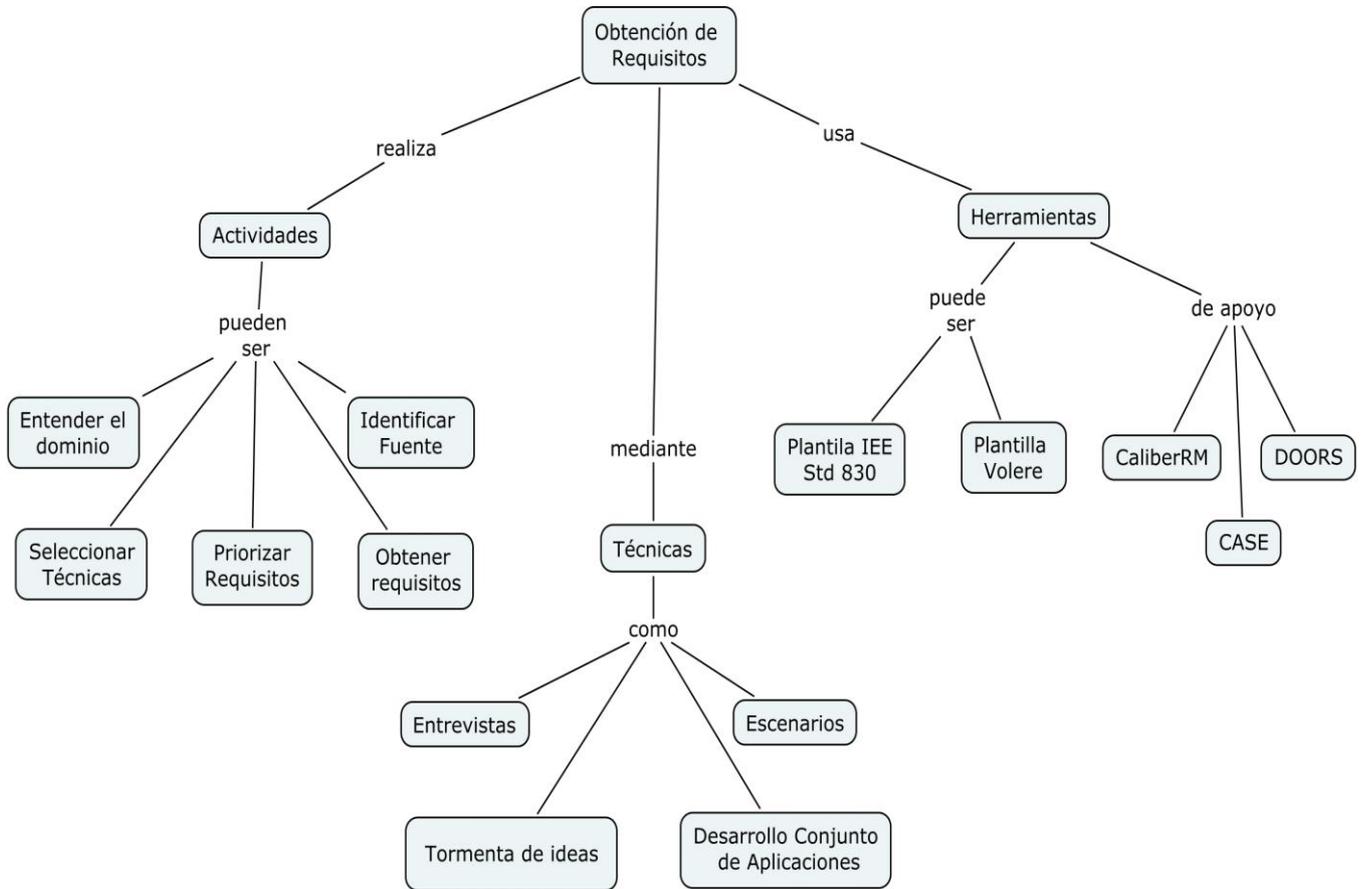


Figura 12: Representación del Mapa de la actividad de Obtención de Requisitos.

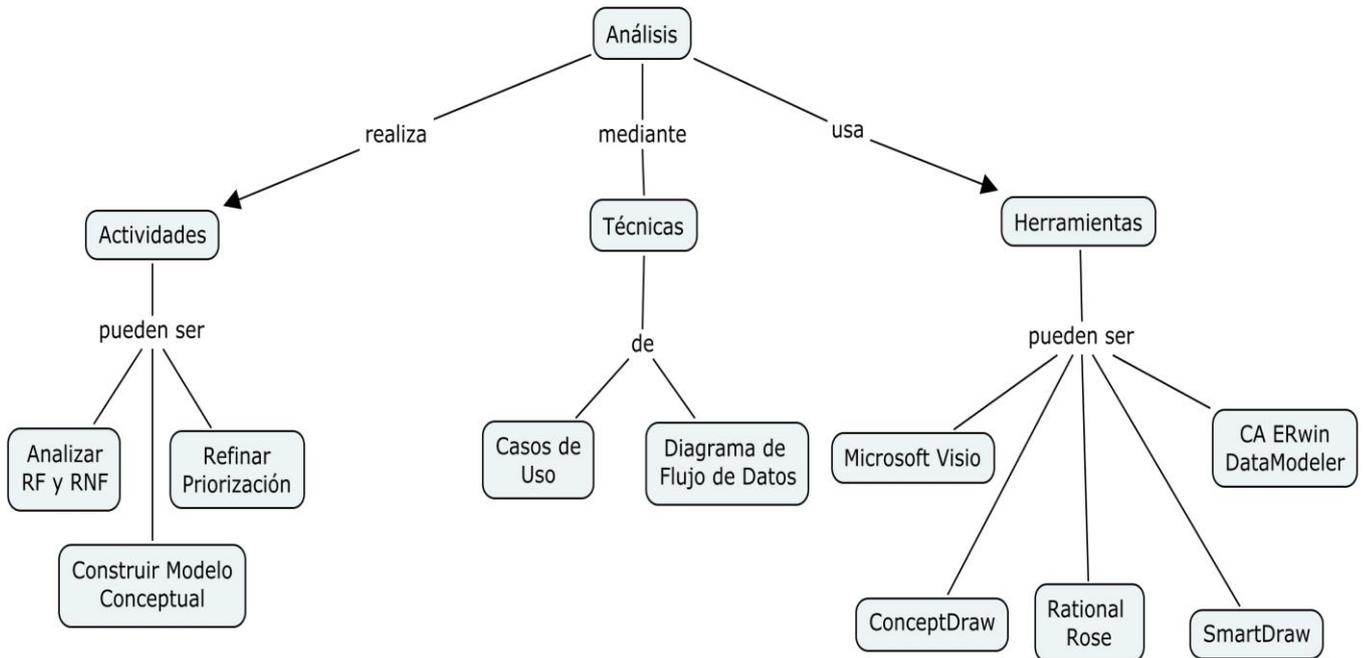


Figura 13: Representación del Mapa de la actividad de Análisis de requisitos.

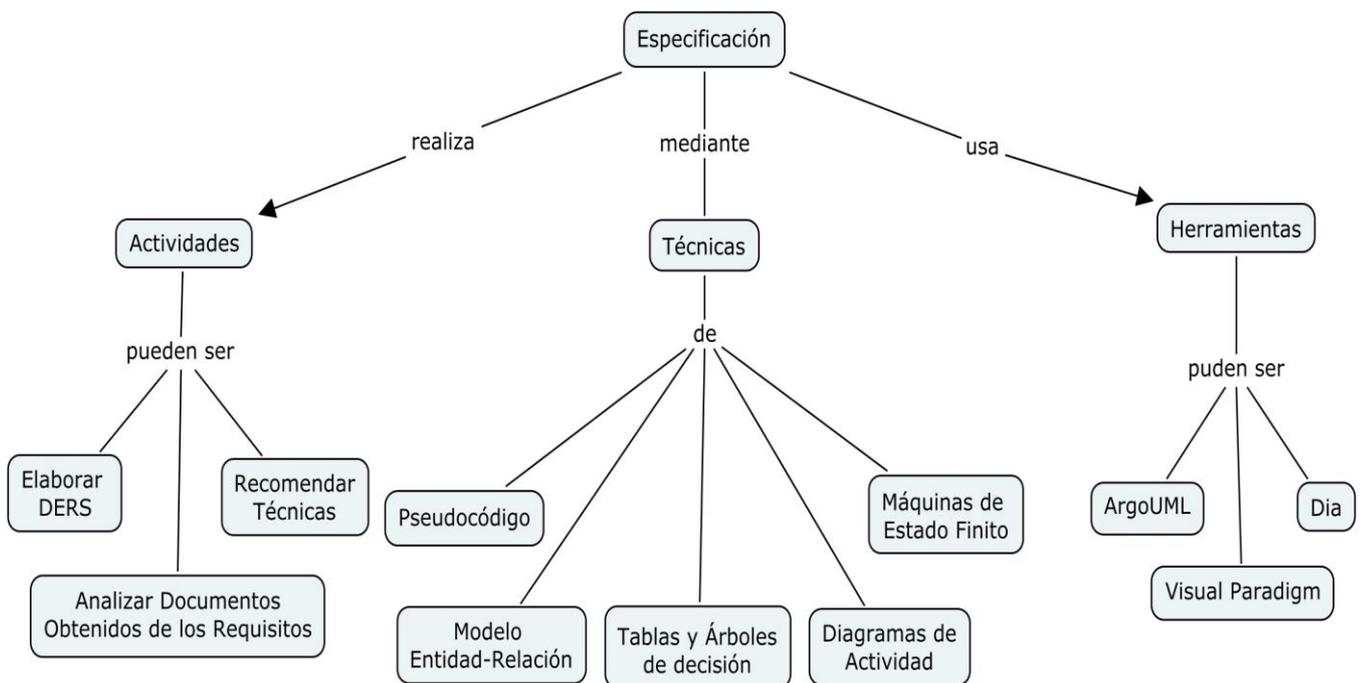


Figura 14: Representación del mapa de la actividad de Especificación de requisitos.

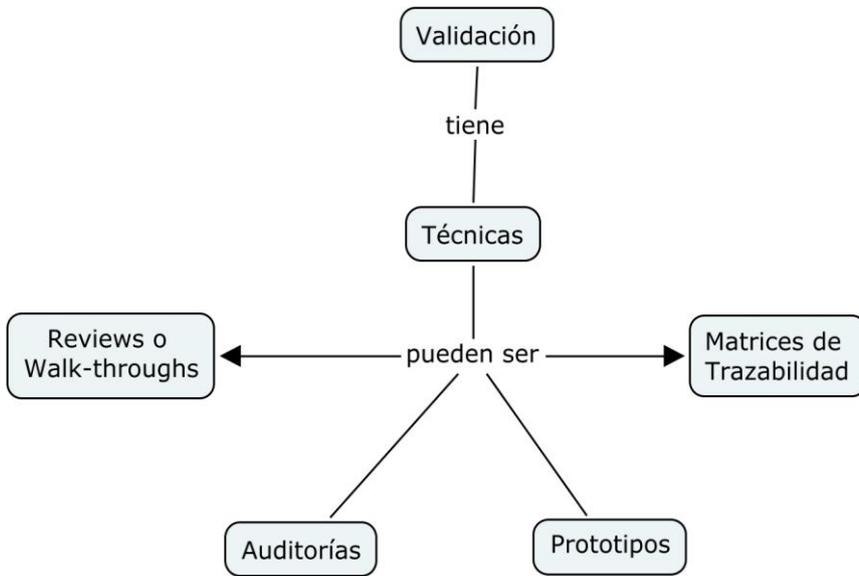


Figura 15: Representación del mapa de la actividad de Validación de requisitos.

Mapa Conceptual de Gestión de Requisitos

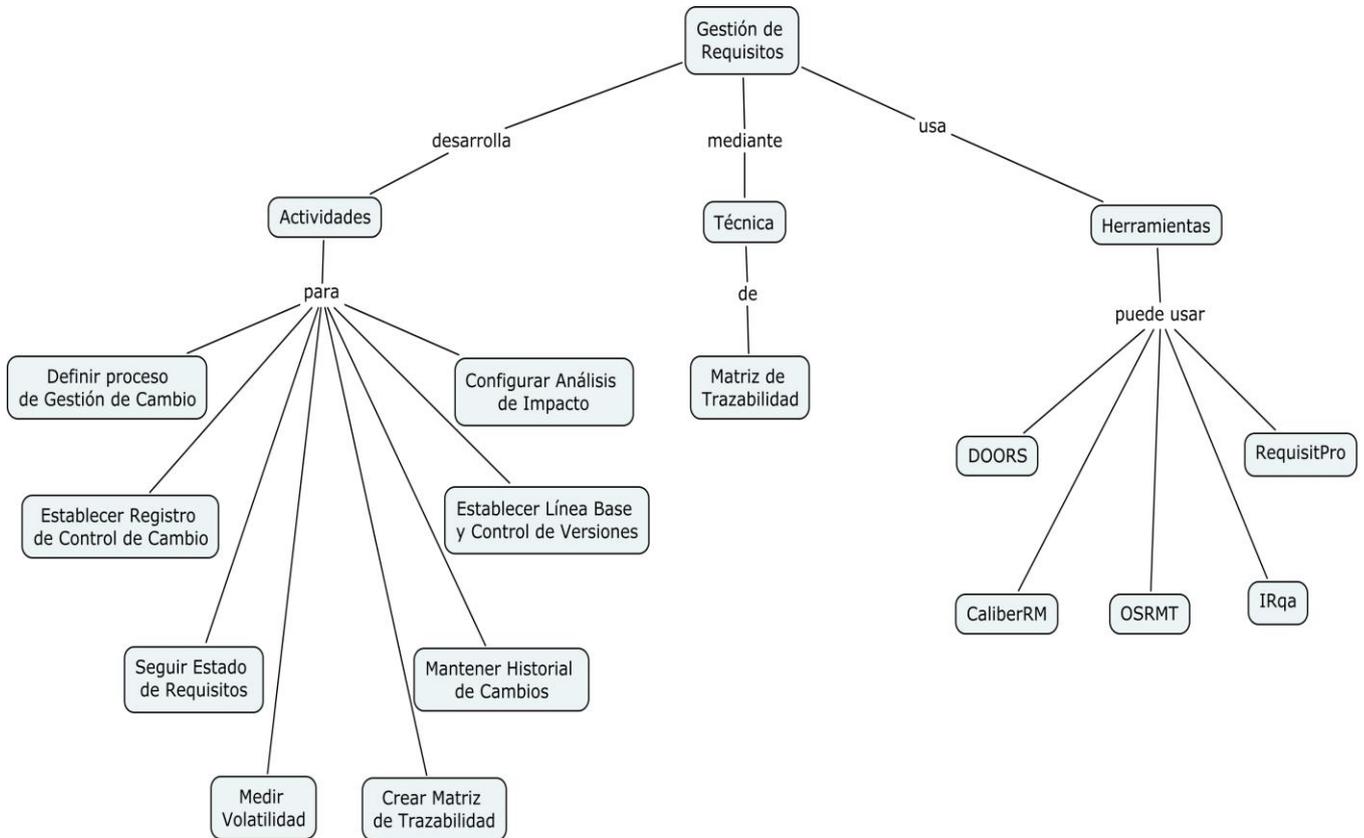


Figura 16: Representación del mapa del proceso de Gestión de Requisitos.

Consideraciones del capítulo.

En este capítulo se hizo la descripción de la solución. Se determinaron los requisitos funcionales a los que debía responder la aplicación y los casos de uso que daban solución a estos requisitos, quedando agrupadas las funcionalidades en un total de 7 casos de uso.

Se construyeron los mapas conceptuales de la disciplina y se distribuyeron en los procesos de desarrollo y gestión de requisitos, distribuyéndose cada uno en actividades que realizan, herramientas que utilizan y técnicas que desarrollan.

CAPÍTULO 3: DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

3.1. Principios de diseño.

Diseñar una aplicación implica una serie de elementos a considerar de manera que el diseño de ésta no atente contra su calidad. La información debe estar correctamente estructurada, y presentada de manera que el impacto afectivo hacia dicha información sea positivo.

Es importante tener en cuenta que la forma en que aparece la información (texto o gráfico) incide en la forma en que la percibimos. (Chance, 2005)

3.1.1. Diseño claro, consistente y con contraste.

El diseño de la multimedia, toma en cuenta un elemento muy importante a la hora de diseñar y es el hecho de no abarrotar la multimedia de contenido en un solo lugar, así se evita la confusión en el momento de navegar a través de ésta. En materia de jerarquía, se tuvo en cuenta, que el contenido más importante debe ir ubicado en el tope o al lado izquierdo de la multimedia.

En la multimedia se tuvo en cuenta evitar el cambio constante de ambiente de la misma ya que el cambio de ambiente constante requiere un ajuste mental constante debido a que la mente ordena la información, en parte, en el ambiente perceptivo (de la forma en que lo capta) y este constante cambio interfiere con el contenido.

3.1.2. Diseño de pantallas.

El diseño de pantallas de la multimedia se basa principalmente en ubicar los botones de contenido al lado izquierdo de la multimedia, poniendo el elemento mas importante en la parte superior y a la izquierda, de este modo, la selección de todos los temas se encuentra a la izquierda, comenzando por el tema 1 como más importante, y los epígrafes o puntos a tratar de ese tema están ubicados en la parte superior con tendencia a la izquierda.

Cada pantalla da la posibilidad de volver al inicio de ésta o de los temas de manera general. Moverse a través de las pantallas no sería un elemento difícil dentro de la multimedia.

3.1.3. Colores.

Los colores ayudan a crear el fondo y las imágenes de primer plano en la multimedia y la distinción entre estos. Los colores claros indican fondo mientras que los más oscuros se identifican con el contenido de un primer plano.

En este sentido se evitó poner colores fuertes como fondo debido a que des enfatiza los textos y los gráficos que deben estar en el plano principal. La multimedia usa 4 colores fundamentalmente debido a que el uso de muchos colores provoca distracción.

El color azul se muestra como predominante en toda la multimedia ya que representa un color confortante, acompañado por el color blanco como fondo para las pantallas que albergan los textos, haciendo contraste con las letras mostradas en color negro y resaltando las mismas, cuando sea necesario, con los colores Negro, Azul Oscuro o Rojo.

3.2. Descripción de archivos XML.

Para almacenar los textos de la multimedia se utiliza la tecnología XML. Para cada película flash, se destina un archivo XML encargado de contener la información referente a esta.

3.2.1. Estructura de los archivos XML.

Tabla 11: Representación de la estructura XML del archivo IR.xml.

```
<Curso>
  <nombrePerfil>< nombrePerfil>
  <nombreCurso> </nombreCurso>
</Curso>
```

Explicación de la estructura que se encuentra representada en la tabla 11

<Curso> es un nodo XML que indica que a partir de aquí cuando el usuario abra para seleccionar el nombre del curso, se le va a mostrar en nombre de la disciplina y el nombre del perfil.

</Curso> es el cierre del nodo **<Curso>**.

<nombreperfil> es un nodo XML que indica que en esta etiqueta estará reflejado el nombre del perfil que se estudia.

<nombreperfil> cierre de nodo.

<nombreCurso> es un nodo XML que indica que en esta etiqueta estará reflejado el nombre del curso en el momento que el usuario seleccione el mismo.

</nombreCurso>

Tabla 12: Representación de la estructura del archivo Escoger Tema.xml.

```
<escogerTema>
  <concepto>
  </concepto>
  <tema1..n>
  </tema1..n>
  <AutoEval >
  </AutoEval>
  <juego>
  </juego>
</escogerTema>
```

<escogerTema> este nodo indica que a partir de aquí se va a mostrar información referente a que contenido va a tener cada tema.

</escogerTema> este es el cierre del nodo.

<concepto> este nodo indica que se va a dar un concepto introductorio a lo que se va a tratar a lo largo del curso de perfil.

</concepto> este es el cierre del nodo.

<tema1.....n> este nodo va a contener la información que se va manejar una vez que se seleccione el tema del botón sobre el que se está.

</tema.....n> cierre del nodo.

<AutoEval> este nodo contiene información sobre el contenido de autoevaluación.

</AutoEval> cierre del nodo.

<juego> este nodo muestra la información que va a contener el botón juego.

</juego> cierre del nodo.

La estructura para todos los XML de los diferentes temas a tratar es la misma (Figura 12).

Tabla 13: Representación de la estructura del archivo Tema.xml.

```
<tema>
  <nombre título=" " />
  <nombre epígrafe=" "/>
  <contenido><![CDATA[ ]]></contenido>
</tema>
```

<tema> este nodo indica que cuando abra este XML se comenzará a tratar lo referente a los temas.

</tema> cierre del nodo.

<nombre título/> este nodo declara una variable título para ser llamada como valor de nodo.

< nombre epígrafe ="" /> este nodo declara una variable epígrafe para ser llamada como valor de nodo, para todos los epígrafes tiene la misma estructura.

![CDATA[]] esta etiqueta es una sección XML para definir una estructura que especifica datos utilizando cualquier caracter sin que se interprete como marcado XML.

<contenido> este nodo indica que dentro de el estará toda la información del tema, aquí se expondrán concepto y demás elementos del curso.

</contenido> cierre del nodo.

Tabla 14: Representación de la estructura del archivo info.xml.

```
<info>
    <nombre título=" " />
    <contenido></contenido>
</info>
```

<info> este nodo indica que a partir de aquí se especificará contenido relación sobre como desarrollar el juego o la evaluación.

</info> cierre del nodo

<nombre título=" " /> este nodo indica que a partir de aquí se especificará que se esta tratando con una evaluación. El nodo XML tiene en su estructura una variable declarada de nombre titulo.

<contenido> este nodo indica que a partir de aquí se expondrá el texto que habla sobre el contenido de los juego o las autoevaluaciones.

</contenido> cierre del nodo.

Aquí se muestra la estructura que tiene el XML de la evaluación.

Tabla 15: Representación de la estructura del archivo quiz.xml.

```
<quiz>

  <title> </title>

  <items>

    <item>

      <question> </question>

      <answer correct="y">Falso</answer>

      <answer>Verdadero</answer>

    </item>

    <item>

      <question></question>

      <answer></answer>

      <answer correct="y"></answer>

    </item>

    <item>

      <question> </question>

      <answer correct="y">Falso</answer>

      <answer>Verdadero</answer>

    </item>

  </items>

</quiz>
```

```
</items>  
  
</quiz>
```

<quiz> este nodo da inicio al XML de contenido de la evaluación.

</quiz> cierre de nodo.

<title> nombre del titulo de la evaluación.

<items> nodo que va a contener todos los elementos de las preguntas de la autoevaluación.

</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> cierre de nodo.

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

El glosario de términos esta compuesto por algunas palabras y sus significados que puedan levantar algún tipo de duda o confusión, o que en algún momento sirva de apoyo a la obtención de conocimiento. La estructura del mismo esta detallada en la tabla 16.

Tabla 16: Representación de la estructura del archivo glosario.xml.

```
<glosario>
```

```
<palabra nombre=" " significado=" "/>
```

```
</glosario>
```

<glosario> Este nodo es el que va a abrir el contenido del XML del glosario de términos.

<palabra nombre=" " significado=" "/> este nodo es el que va a contener las palabras y sus significados. El nodo palabra va a tener dos variables declaradas; nombre que va a tener el término y significado para su definición.

</glosario> cierre de nodo.

3.3. Diagramas de presentación.

El diagrama de presentación (Figura 17-22) es un artefacto nuevo dentro del lenguaje UML, incorporado a éste a partir de la extensión del mismo planteada por OMMMA-L. Éste sirve para describir la parte estática del modelo a través de una descripción intuitiva de la distribución espacial de objetos visuales de la interfaz de usuario. (Jiménez, 2005)

Este diagrama tiene como propósito declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización (texto, gráfico, video, animación) e interacción (scrolls, barras de menú botones, campos de entrada y salida).

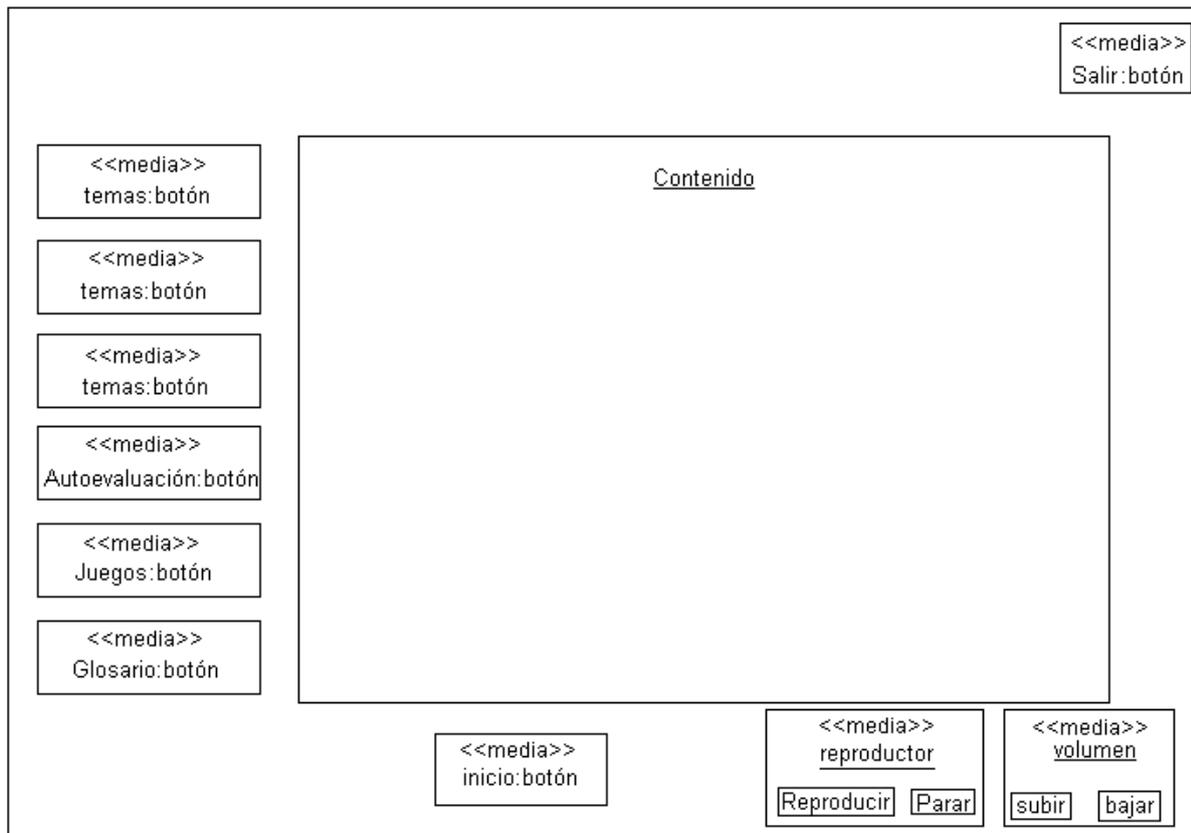


Figura 17: Diagrama de presentación general.

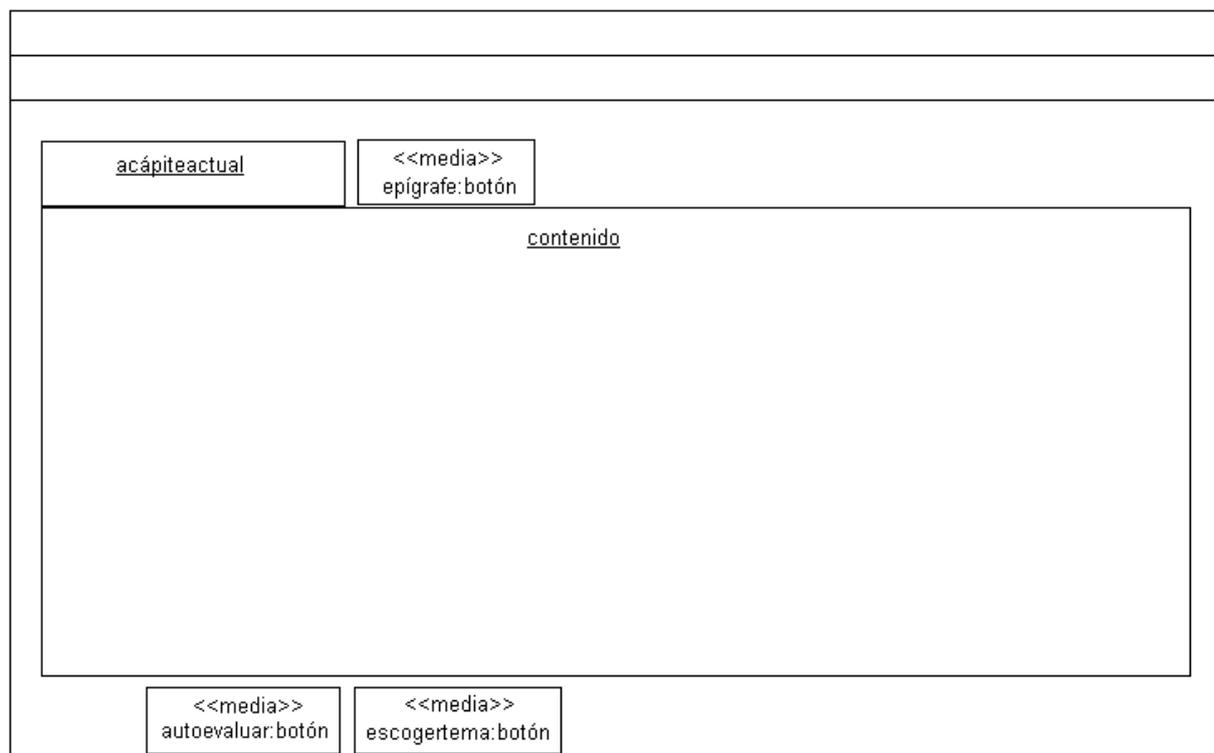


Figura 18: Diagrama de presentación de los temas.

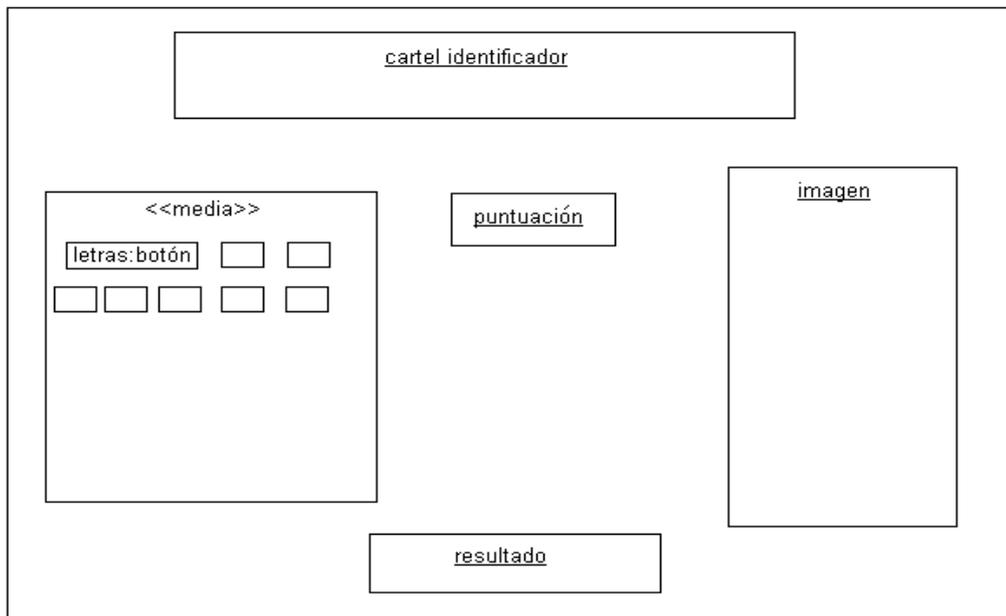


Figura 19: Diagrama de presentación del Juego del Ahorcado.

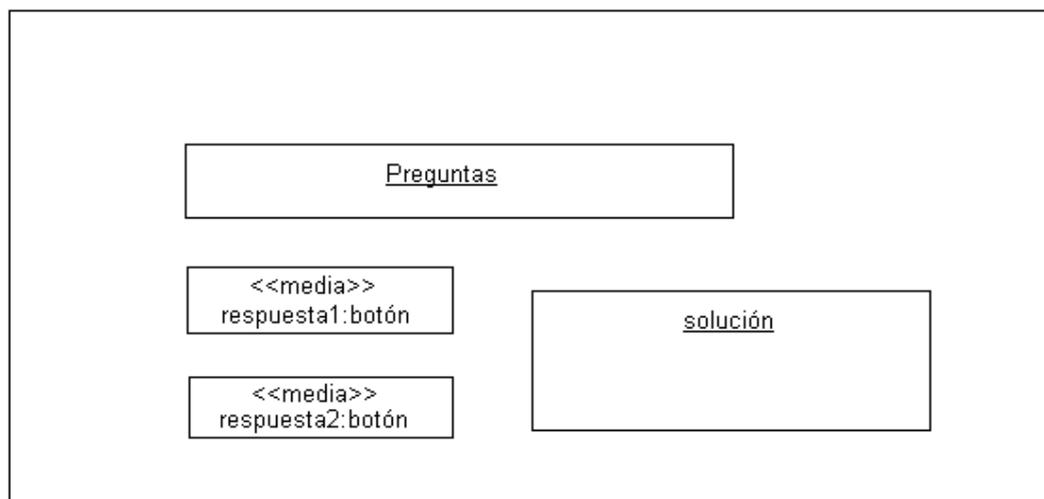


Figura 20: Diagrama de presentación de Autoevaluación.

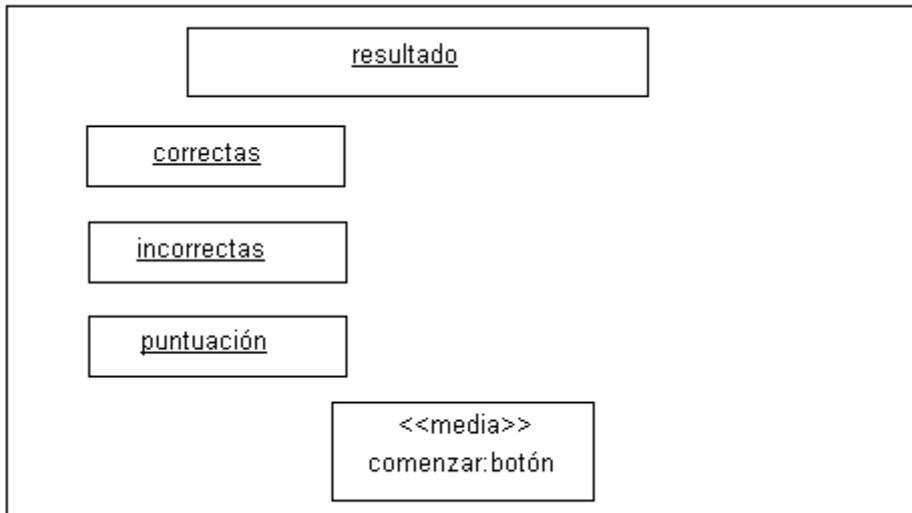


Figura 21: Diagrama de presentación de Resultado de Evaluación.

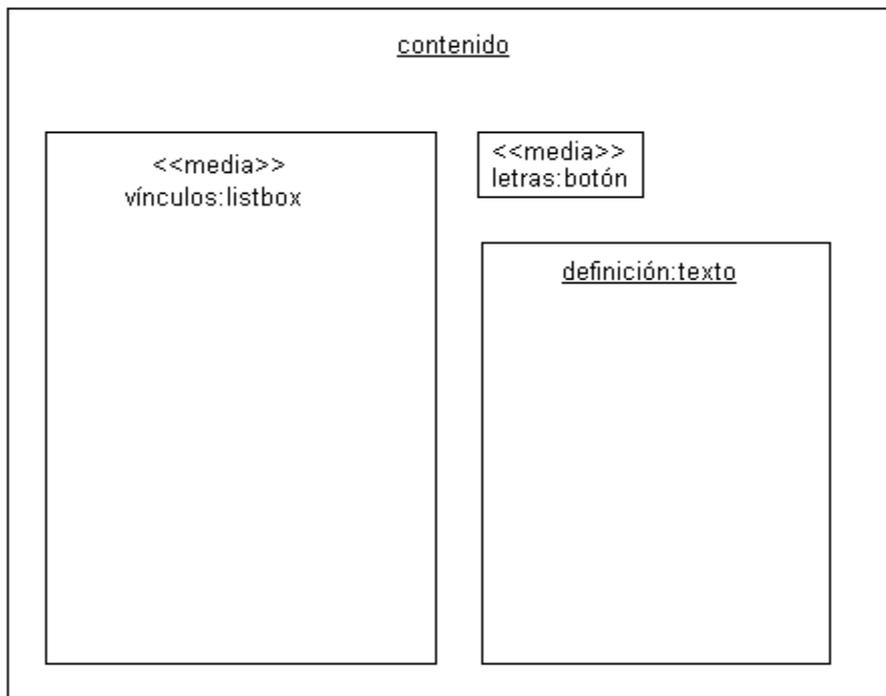


Figura 22: Diagrama de Presentación del Glosario de términos.

3.4. Modelo de implementación.

3.4.1. Diagrama de Despliegue.

El diagrama de despliegue (Figura 23) es un tipo de diagrama del Lenguaje Unificado de Modelado que se utiliza para modelar el hardware utilizado en las implementaciones de sistemas y las relaciones entre sus componentes.



Figura 23: Diagrama de despliegue.

3.4.2. Diagrama de Componentes.

Un diagrama de componentes (Figura 24) representa la separación de un sistema de software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes, etc.) y muestra las dependencias entre estos componentes.

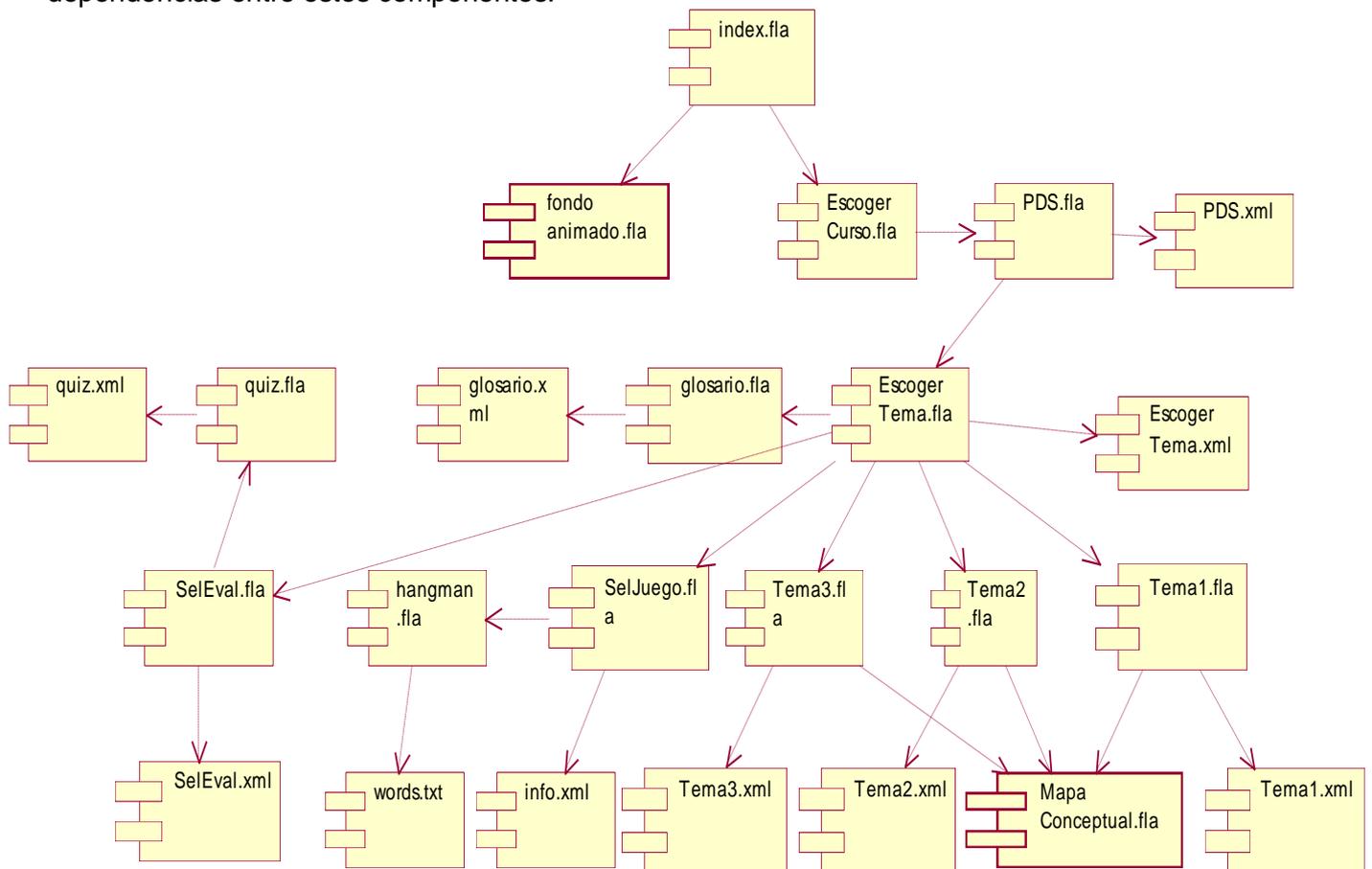


Figura 24: Diagrama de Componentes de la multimedia.

Consideraciones del capítulo.

En este capítulo se hizo un análisis de los principios de diseño en los que se basó la aplicación, así como las diferentes estructuras XML usadas. Se dio además, una panorámica de la multimedia a través de las pantallas de presentación así como los artefactos de la implementación. Se representó la manera en que quedaron distribuidos los diferentes ficheros de la multimedia y las relaciones existentes entre éstos.

CONCLUSIONES

Una vez terminado el trabajo se puede concluir que:

- Se desarrolló con éxito la multimedia de la Disciplina de Ingeniería de Requisitos (IR).
- Se confeccionaron todos los temas concebidos para el curso de segundo perfil de Calidad de Software y como valor agregado se construyeron y adicionaron a la los mapas conceptuales que reúnen los principales conceptos asociados al desarrollo de la disciplina de la IR.
- La multimedia cuenta con una interfaz amigable que integra la utilización del lenguaje XML para agrupar y gestionar los datos en volúmenes compactos de información, quedando evidenciado el vínculo que existe entre XML y la tecnología multimedia.
- La puesta en marcha de esta aplicación trae consigo un valor educacional que se expresa en el aumento del conocimiento sobre temas de la IR apoyándose en mapas conceptuales que agrupan los principales conceptos de la disciplina.
- Todo lo anteriormente expuesto permitió que se alcanzara el objetivo propuesto y que todas las tareas a desarrollar fueran cumplidas con éxito, especialmente la de diseño e implementación de la multimedia.

RECOMENDACIONES

Se recomienda para futuras versiones, la integración de la multimedia, a un conjunto que agrupe las multimedias de las demás disciplinas.

Se recomienda la constante actualización del contenido de la multimedia debido a los rápidos cambios que en materia de Ingeniería de Requisitos de producen a nivel mundial.

BIBLIOGRAFÍA

Adobe. 2008. [Online] 2008. [Cited: 2 19, 2008.] <http://www.adobe.com/products/>.

ALS. 2008. ALS Software Lifecycle Optimization. [Online] 2008. [Cited: 2 20, 2008.] <http://www.als-es.com/home.php?location=herramientas/entorno-desarrollo/optimaltrace>.

Chaves, Michael Arias. 2008. La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [Online] febrero 12, 2008. [Cited: junio 16, 2008.] http://www.intersedes.ucr.ac.cr/10-art_11.html. ISSN 1409-4746.

COIT. 2008. Tendencias en tecnología multimedia: el reto de Internet móvil. [Online] 2 20, 2008. [Cited: 2 20, 2008.] <http://www.coit.es/publicac/publbit/bit125/especial1.htm>.

CREWS. 2008. Cooperative Requirements Engineering with Scenarios. [Online] 2008. [Cited: 4 28, 2008.] <http://sunsite.informatik.rwth-aachen.de/CREWS/workshop99.htm>.

Díaz Cruz, Yaremis and Pérez Hernández, Yenislely. 2007. *Multimedia para la enseñanza aprendizaje del tema Teoría. TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA.* 2007.

Enciclopedia. 2008. Enciclopedia Libre Universal en Español. [Online] 2008. [Cited: 6 14, 2008.] http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n.

Febles, Aylin y Álvarez, S. 2005. *La Gestión de Configuración y el Desarrollo de Software en las Universidades.* s.l. : Revista Cubana de Educacion Superior, 2005.

Fraga Filho, Clayton Vieira. 2008. Controla: Herramienta de apoyo al proceso de ingeniería de software en pequeñas empresas. [Online] 2008. [Cited: 6 25, 2008.] <http://www.mastermagazine.info/articulo/8666.php>.

Herrera, Lizka Johany. 2003. Ingeniería de Requerimientos. *Ingeniería de Requerimientos, Ingeniería de Software.* [Online] 2003. [Cited: 2 11, 2008.] <http://www.monografias.com/trabajos6/resof/resof.shtml> - 74k.

IEEE Computer Society. 2004. *Guide to the Software Engineering Body of knowledge.* 2004.

—. **1997.** *IEEE Software Requirements Engineering.* [ed.] Richard H. Thayer and Merlin Dorfman. Second Edition. New York : s.n., 1997.

IEEE. 1998. *IEEE Recommended Practice for Software Requirements Specifications.* 1998. ISBN 0-7381-0332-2.

Jacinto, Luis. 2007. SIGA. [Online] 11 26, 2007. [Cited: 11 26, 2007.] http://www.siga.com.mx/index.php?option=com_content&view=section&layout=blog&id=5&Itemid=36 .

Jiménez, S.V. 2005. *Propuesta del Proceso de Producción para el departamento de multimedia educativa de la Universidad de Ciencias informáticas.* Ciudad de la Habana : s.n., 2005.

- Labañino Rizzo, César A and del Toro Rodríguez, Mario. 2001.** *Multimedia para la educación.* s.l. : Pueblo y Educación, 2001. ISBN 959 - 13 - 0847 - 7.
- Leffingwell, Dean and Widrig, Don. 2003.** *Managing Software Requirements: A Use Case Approach.* s.l. : Addison Wesley, 2003. Vol. Second Edition. 0-321-12247-X.
- Magadán Labrada, Yaritza and León Perdomo, Yeniset. 2007.** *¿Cómo gestionar la configuración para lograr un producto con calidad en la UCI?* 2007. Tesis para Optar por el Título de Ingeniero en Informática.
- Microsoft Encarta. 2006.** *Multimedia.* 2006.
- MINREX. 2004.** La informatización en Cuba. *MINREX.* [Online] 2004. [Cited: 12 5, 2007.] http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm.
- Miranda, Sandor Luis y Romero, Arturo Luis. 2008.** La calidad, su evolución histórica y algunos conceptos y términos asociados. [En línea] 2008. [Citado el: 24 de 4 de 2008.] www.grupokaizen.com/sig/EVOLUCION_HISTORICA_DE_LA_CALIDAD_Y_OTROS_CONCEPTOS.doc.
- Montero Ayala, Ramón. 2001.** *XML Iniciación y Referencia.* Madrid : McGraw-Hill, 2001. 84-48 1 -2894-X.
- Oh Navarro, Emily and van der Hoek, André.** [Online] [Cited: Junio 2, 2008.] <http://www.ics.uci.edu/~emilyo/publications.html>.
- P.Sawyer, I.Sommerville and. 1997.** *Requirements Engineering: A Good Practice Guide.* New York : John Wiley & Sons, Inc, 1997.
- Palmkvist, Karin and Övergaard, Gunnar. 2004.** *Use Cases Patterns and Blueprints.* s.l. : Addison Wesley Professional, 2004. 0-13-145134-0.
- Pérez, Karina. 2008.** *Procesos, Técnicas y Herramientas para IR.* 2008.
- Pressman, Roger S. 2005.** *Ingeniería del Software Un enfoque práctico.* s.l. : Félix Varela, 2005.
- Rational Software Corporation. 2003.** *Rational Unified Process.* 2003. 2003.06.00.65.
- Robertson, Suzanne and Robertson, James. 2006.** *Mastering the Requirements Process Second Edition.* s.l. : Addison Wesley Professional, 2006. ISBN-13: 978-0-321-41949-1.
- Salinas, Jesús M. 1996.** MULTIMEDIA EN LOS PROCESOS DE ENSEÑANZA-APRENDIZAJE:ELEMENTOS DE DISCUSION. [Online] 1996. [Cited: 2 20, 2008.] <http://edutec.rediris.es/documentos/1996/multimedia.html>.
- Sauer, Stefan and Engels, Gregor.** [Online] [Cited: Junio 2, 2008.] <http://wwwcs.upb.de/cs/ag-engels/Papers/2001/SauerHCC01.pdf>.
- SEI. 2008.** Software Engineering Institute. [Online] 2008. [Cited: 3 6, 2008.] <http://www.sei.cmu.edu/str/indexes/glossary/requirements-engineering.html>.

Soler, Dalianny. 2007. *Propuesta de realización de la Ingeniería de Requisitos en los proyectos de la Universidad de las Ciencias Informáticas (UCI)*. UCI. 2007. TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA.

Sommerville, I and Sawyer, P. 1997. *Requerimientos Engineering: A Good Practice Guide*. s.l. : John Wiley and Sons, 1997.

Sommerville, Ian. 1996. *Software Engineering*. s.l. : Addison-Wesley, 1996.

—. **1992.** *Software Engineering*. s.l. : Addison Wesley, 1992.

U.S. Department of Energy. 2000. *Guideline for Requirement Management*. 2000.

ULA. 2008. [Online] 2 25, 2008. [Cited: 2 25, 2008.] www.ulagos.cl.

Wiegers, Karl E. 2006. *More About Software Requirements: Thorny Issues and Practical Advice*. s.l. : Microsoft Press © , 2006. 0735622671.

Wikipedia. 2008. [Online] 11 28, 2008. [Cited: 4 28, 2008.] http://es.wikipedia.org/wiki/Herramienta_de_Autor.

Wohlin, Claes and Aurum, Aybüke. 2005. *Engineering and Managing Software Requirements*. [ed.] Aybüke Aurum and Claes Wohlin. s.l. : Springer Berlin Heidelberg New York, 2005. 13 978-3-540-25043-2.

Young, Ralph R. 2004. *The Requirements Engineering Handbook*. s.l. : Artech House, 2004. 1-58053-266-7.

REFERENCIAS

- Adobe. 2008.** [En línea] 2008. [Citado el: 19 de 2 de 2008.] <http://www.adobe.com/products/>.
- ALS. 2008.** ALS Software Lifecycle Optimization. [En línea] 2008. [Citado el: 20 de 2 de 2008.] <http://www.als-es.com/home.php?location=herramientas/entorno-desarrollo/optimaltrace>.
- Chaves, Michael Arias. 2008.** La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. [En línea] 12 de febrero de 2008. [Citado el: 16 de junio de 2008.] http://www.intersedes.ucr.ac.cr/10-art_11.html. ISSN 1409-4746.
- COIT. 2008.** Tendencias en tecnología multimedia: el reto de Internet móvil. [En línea] 20 de 2 de 2008. [Citado el: 20 de 2 de 2008.] <http://www.coit.es/publicac/publbit/bit125/especial1.htm>.
- CREWS. 2008.** Cooperative Requirements Engineering with Scenarios. [En línea] 2008. [Citado el: 28 de 4 de 2008.] <http://sunsite.informatik.rwth-aachen.de/CREWS/workshop99.htm>.
- Díaz Cruz, Yaremis y Pérez Hernández, Yenislely. 2007.** *Multimedia para la enseñanza aprendizaje del tema Teoría. TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA.* 2007.
- Enciclopedia. 2008.** Enciclopedia Libre Universal en Español. [En línea] 2008. [Citado el: 14 de 6 de 2008.] http://enciclopedia.us.es/index.php/Lenguaje_de_programaci%C3%B3n.
- Febles, Aylin y Álvarez, S. 2005.** *La Gestión de Configuración y el Desarrollo de Software en las Universidades.* s.l. : Revista Cubana de Educacion Superior, 2005.
- Herrera, Lizka Johany. 2003.** Ingeniería de Requerimientos. *Ingeniería de Requerimientos, Ingeniería de Software.* [En línea] 2003. [Citado el: 11 de 2 de 2008.] <http://www.monografias.com/trabajos6/resof/resof.shtml> - 74k.
- IEEE Computer Society. 2004.** *Guide to the Software Engineering Body of knowledge.* 2004.
- . **1997.** *IEEE Software Requirements Engineering.* [ed.] Richard H. Thayer y Merlin Dorfman. Second Edition. New York : s.n., 1997.
- IEEE. 1998.** *IEEE Recommended Practice for Software Requirements Specifications.* 1998. ISBN 0-7381-0332-2.
- Jacinto, Luis. 2007.** SIGA. [En línea] 26 de 11 de 2007. [Citado el: 26 de 11 de 2007.] http://www.siga.com.mx/index.php?option=com_content&view=section&layout=blog&id=5&Itemid=36 .
- Jiménez, S.V. 2005.** *Propuesta del Proceso de Producción para el departamento de multimedia educativa de la Universidad de Ciencias informáticas.* Ciudad de la Habana : s.n., 2005.
- Labañino Rizzo, César A y del Toro Rodríguez, Mario. 2001.** *Multimedia para la educación.* s.l. : Pueblo y Educación, 2001. ISBN 959 - 13 - 0847 - 7.

- Leffingwell, Dean y Widrig, Don. 2003.** *Managing Software Requirements: A Use Case Approach*. s.l. : Addison Wesley, 2003. Vol. Second Edition. 0-321-12247-X.
- Magadán Labrada, Yaritza y León Perdomo, Yeniset. 2007.** *¿Cómo gestionar la configuración para lograr un producto con calidad en la UCI?* 2007. Tesis para Optar por el Título de Ingeniero en Informática.
- Microsoft Encarta. 2006.** *Multimedia*. 2006.
- MINREX. 2004.** La informatización en Cuba. *MINREX*. [En línea] 2004. [Citado el: 5 de 12 de 2007.] http://www.cubaminrex.cu/Sociedad_Informacion/Cuba_SI/Informatizacion.htm.
- Miranda, Sandor Luis y Romero, Arturo Luis. 2008.** La calidad, su evolución histórica y algunos conceptos y términos asociados. [En línea] 2008. [Citado el: 24 de 4 de 2008.] www.grupokaizen.com/sig/EVOLUCION_HISTORICA_DE_LA_CALIDAD_Y_OTROS_CONCEPTOS.doc.
- Montero Ayala, Ramón. 2001.** *XML Iniciación y Referencia*. Madrid : McGraw-Hill, 2001. 84-48 1 -2894-X.
- Oh Navarro, Emily y van der Hoek, André.** [En línea] [Citado el: 2 de Junio de 2008.] <http://www.ics.uci.edu/~emilyo/publications.html>.
- P.Sawyer, I.Sommerville and. 1997.** *Requirements Engineering: A Good Practice Guide*. New York : John Wiley & Sons, Inc, 1997.
- Pérez, Karina. 2008.** *Procesos, Técnicas y Herramientas para IR*. 2008.
- Pressman, Roger S. 2005.** *Ingeniería del Software Un enfoque práctico*. s.l. : Félix Varela, 2005.
- Rational Software Corporation. 2003.** *Rational Unified Process*. 2003. 2003.06.00.65.
- Robertson, Suzanne y Robertson, James. 2006.** *Mastering the Requirements Process Second Edition*. s.l. : Addison Wesley Professional, 2006. ISBN-13: 978-0-321-41949-1.
- Salinas, Jesús M. 1996.** MULTIMEDIA EN LOS PROCESOS DE ENSEÑANZA-APRENDIZAJE: ELEMENTOS DE DISCUSION. [En línea] 1996. [Citado el: 20 de 2 de 2008.] <http://edutec.rediris.es/documentos/1996/multimedia.html>.
- Sauer, Stefan y Engels, Gregor.** [En línea] [Citado el: 2 de Junio de 2008.] <http://www.wcs.upb.de/cs/ag-engels/Papers/2001/SauerHCC01.pdf>.
- SEI. 2008.** Software Engineering Institute. [En línea] 2008. [Citado el: 6 de 3 de 2008.] <http://www.sei.cmu.edu/str/indexes/glossary/requirements-engineering.html>.
- Soler, Daliany. 2007.** *Propuesta de realización de la Ingeniería de Requisitos en los proyectos de la Universidad de las Ciencias Informáticas (UCI)*. UCI. 2007. TRABAJO PARA OPTAR POR EL TÍTULO DE INGENIERÍA EN INFORMÁTICA.

Sommerville, I y Sawyer, P. 1997. *Requerimientos Engineering: A Good Practice Guide*. s.l. : John Wiley and Sons, 1997.

Sommerville, Ian. 1996. *Software Engineering*. s.l. : Addison-Wesley, 1996.

—. **1992.** *Software Engineering*. s.l. : Addison Wesley, 1992.

U.S. Department of Energy. 2000. *Guideline for Requirement Management*. 2000.

Vieira Fraga Filho, Clayton. Controla: Herramienta de apoyo al proceso de ingeniería de software en pequeñas empresas. [En línea] [Citado el: 25 de 6 de 2008.] <http://www.mastermagazine.info/articulo/8666.php>.

—. Controla: Herramienta de apoyo al proceso de ingeniería de software en pequeñas empresas. [En línea] [Citado el: 25 de 6 de 2008.] <http://www.mastermagazine.info/articulo/8666.php>.

Wiegers, Karl E. 2006. *More About Software Requirements: Thorny Issues and Practical Advice*. s.l. : Microsoft Press © , 2006. 0735622671.

Wikipedia. 2008. [En línea] 28 de 11 de 2008. [Citado el: 28 de 4 de 2008.] http://es.wikipedia.org/wiki/Herramienta_de_Autor.

Wohlin, Claes y Aurum, Aybüke. 2005. *Engineering and Managing Software Requirements*. [ed.] Aybüke Aurum y Claes Wohlin. s.l. : Springer Berlin Heidelberg New York, 2005. 13 978-3-540-25043-2.

ANEXOS

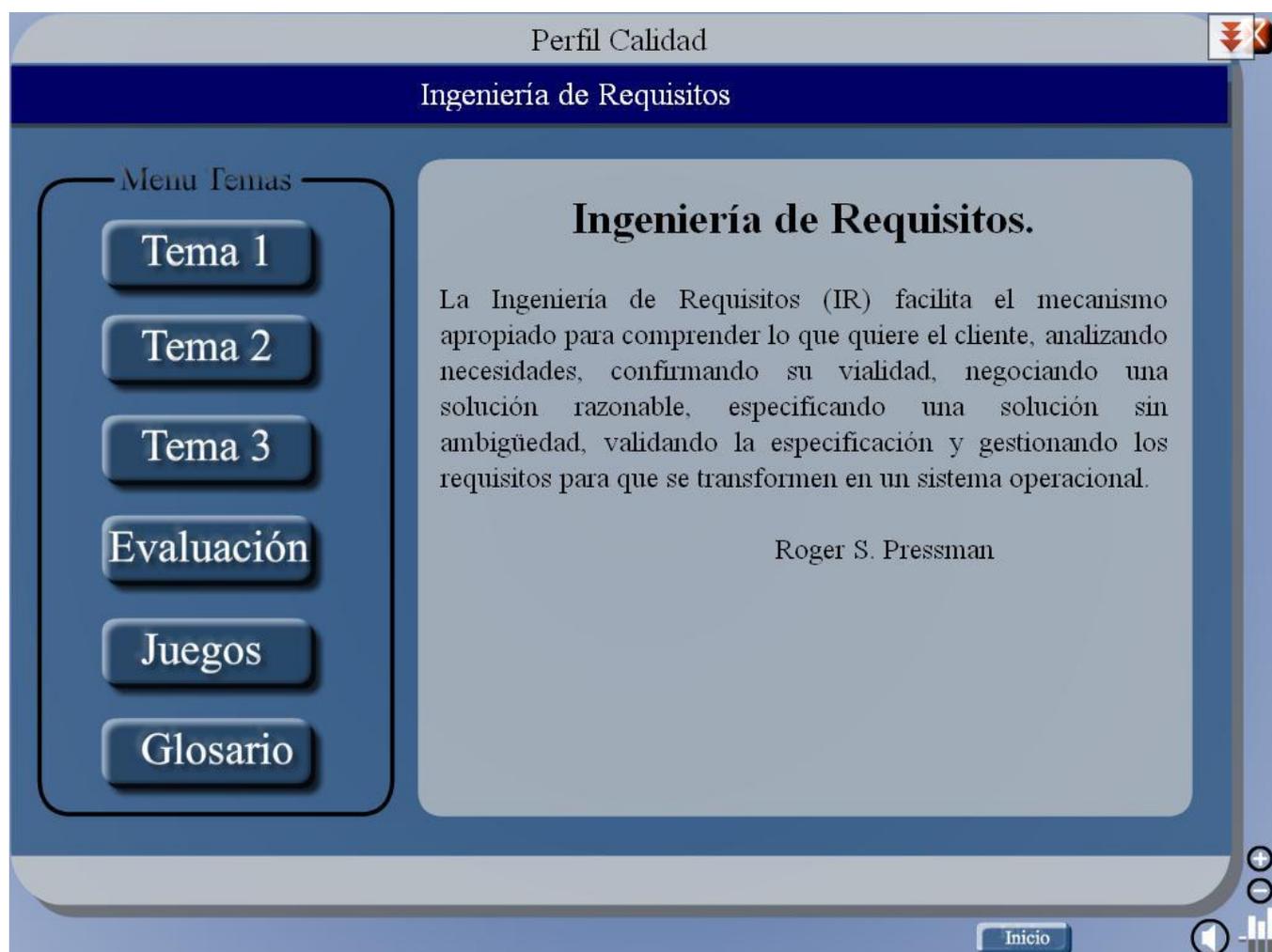


Figura 25: Pantalla Escoger Temas de la Multimedia

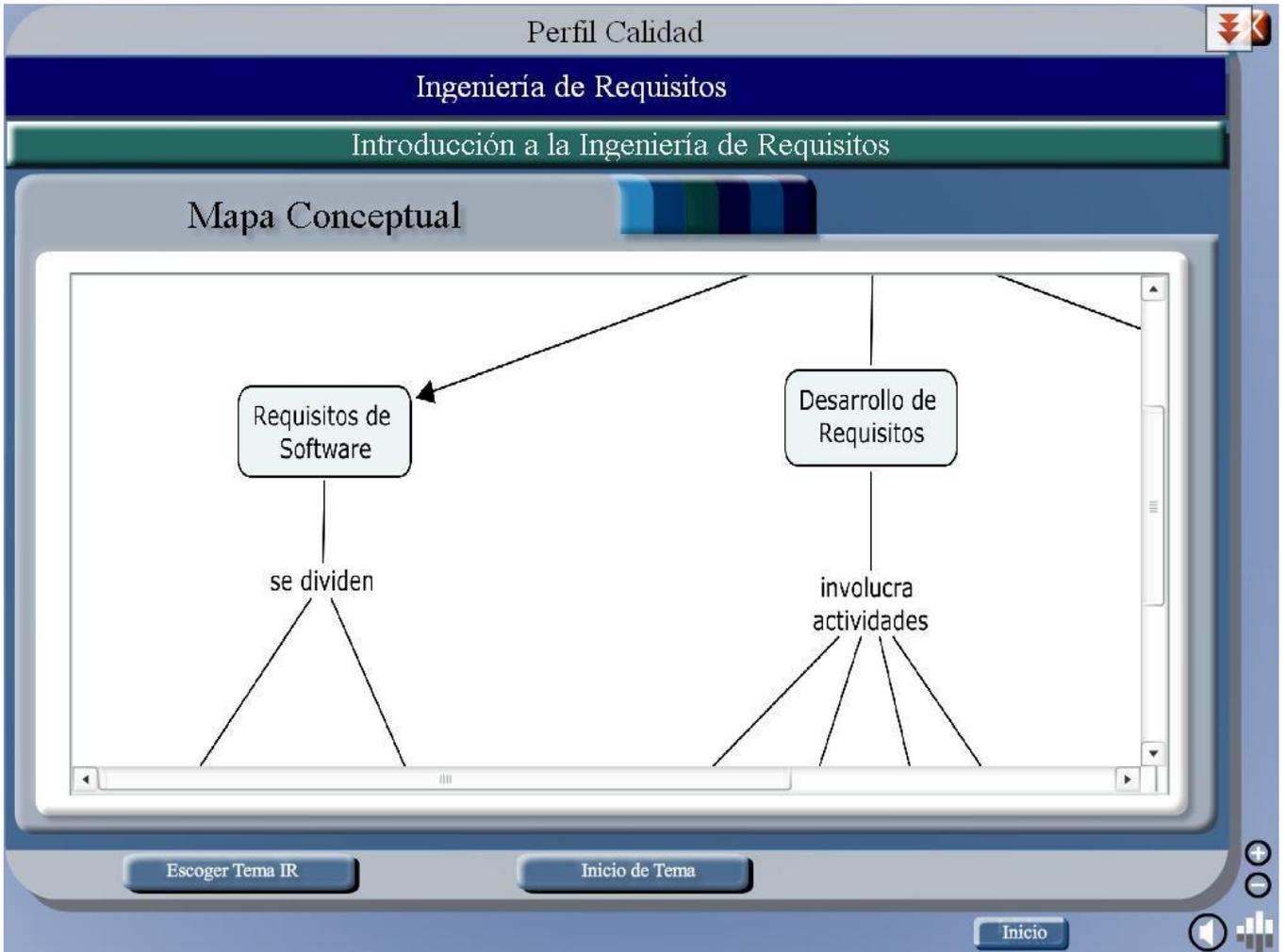


Figura 26 Pantalla con el contenido del mapa conceptual.

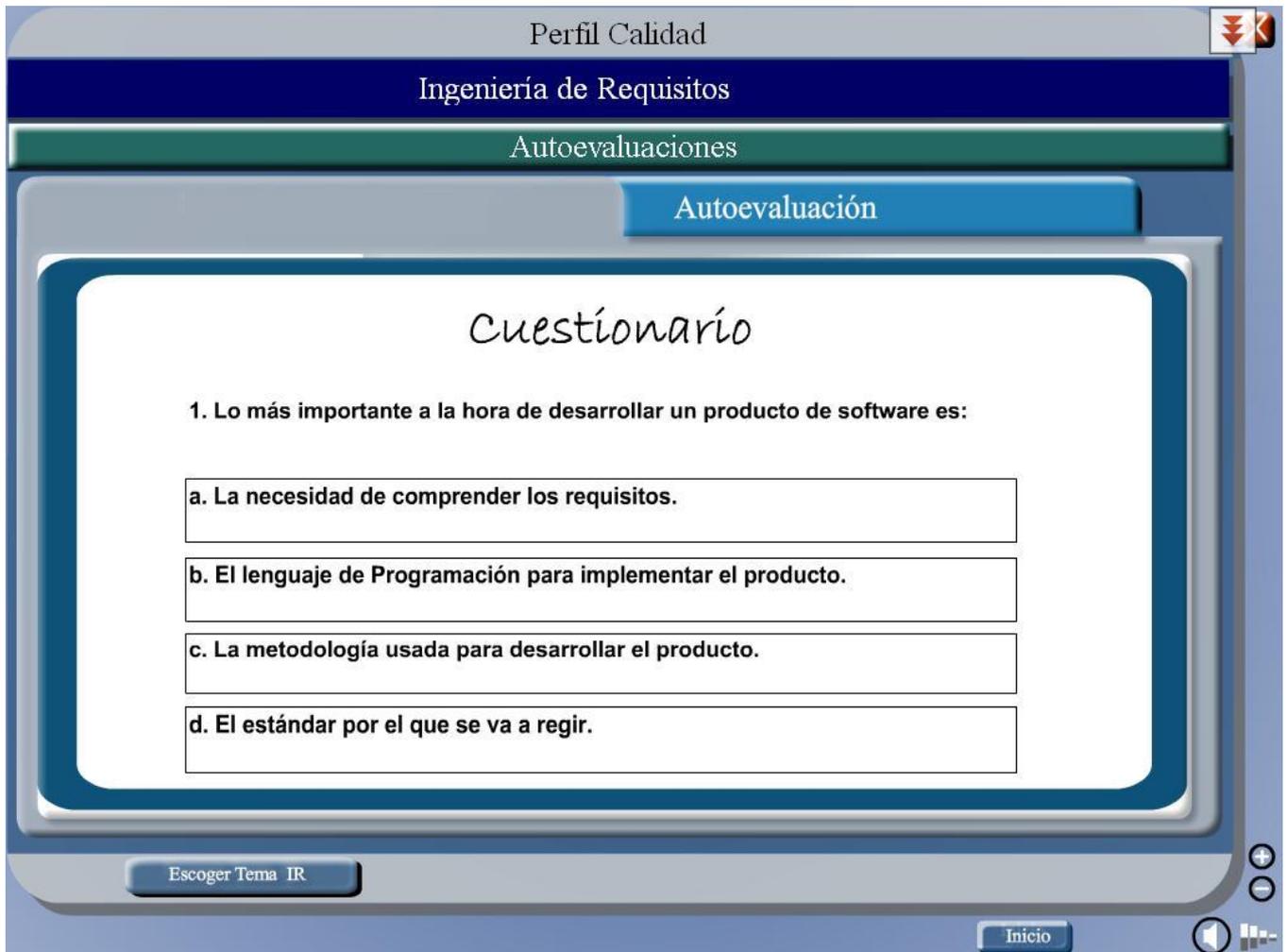


Figura 27 Pantalla de Autoevaluación de los Temas

- 1 GLOSARIO
- 2 **IEEE:** Instituto de Ingenieros Eléctricos y Electrónicos. Una sociedad profesional que mantiene una
3 serie de estándares la gestionar y ejecutar software y proyectos de ingeniería de sistemas.
- 4 **OMMMA-L:** Lenguaje de modelado visual para la programación orientada a objetos de aplicaciones
5 multimedia basado en el lenguaje UML. Es una extensión de UML que adapta los diagramas de éste
6 de acuerdo a las características de la multimedia.
- 7 **RUP:** Rational Unified Process. Proceso Unificado de Rational, divide el proceso de desarrollo en
8 cuatro fases diferentes en las que cada una involucra modelado del negocio, análisis y diseño,
9 implementación, pruebas y despliegue.
- 10 **Software Engineering Institute (SEI):** es un instituto federal estadounidense de investigación y
11 desarrollo, fundado por el Congreso de los Estados Unidos en 1984 para desarrollar modelos de
12 evaluación y mejora en el desarrollo de software, que dieran respuesta a los problemas que generaba
13 al ejército estadounidense la programación e integración de los sub-sistemas de software en la
14 construcción de complejos sistemas militares.
- 15 **UML:** Lenguaje de Modelado usado en la programación orientada a objetos.