

Universidad de las Ciencias Informáticas
Facultad 5



Título: Multimedia Interactiva de Auditorías y
Revisiones de Software.

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

Autor: Maikel Verdecia Guevara

Tutor: MSc. Yamilis Fernández Pérez.

Ing. Miguel Ángel del Pino Zincke

DATOS DE CONTACTO

Síntesis del Tutor: MSc. Yamilis Fernández Pérez

Graduada de Ingeniera en Sistema Automatizado de Dirección, en 1992 en el ISPJAE, y profesora asistente desde 1995. Se graduó de MSc en Informática Aplicada en 1995. Imparte docencia en universidades desde 1992. Ha desarrollado trabajos con universidades extranjeras en Brasil, Bolivia y Canadá. Fue La Jefa de Departamento Docente Central de Ingeniería y Gestión de Software de la UCI desde su fundación hasta el 2008. Desde octubre del 2008 es Directora de Formación de Posgraduado.

Síntesis del Tutor: Ing. Miguel Ángel del Pino Zincke

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

PENSAMIENTO

"Una palabra bien elegida puede economizar no sólo cien palabras sino cien pensamientos".

Henri Poincaré

AGRADECIMIENTOS

AGRADECIMIENTOS

A la Revolución por darme la oportunidad de estudiar.

A mis padres por brindarme su apoyo y comprensión en estos 5 años.

A mi hermana, por estar siempre a mi lado incondicionalmente.

A mi familia por ser tan comprensiva y darme la fuerza para seguir adelante.

A mi tutora, por toda la paciencia que ha tenido y la ayuda que me ha brindado.

A todos aquellos que de una manera u otra han contribuido con la realización de este trabajo.

RESUMEN

En estos últimos tiempos ha habido un desarrollo de las Tecnologías de la Información y Comunicaciones (TCIs), lo que ha permitido desarrollar nuevas formas de representar el conocimiento. La Universidad de las Ciencias Informáticas (UCI) ha estado apoyando la creación de sistemas multimedia destinados al apoyo del proceso de enseñanza, como una nueva forma de representar el conocimiento por medios no tradicionales. En la UCI es necesario alcanzar un nivel de conocimiento en el tema de auditorías y revisiones, que permita la terminación y entrega de los productos de software en tiempo y con el costo previsto.

En el presente trabajo se realiza una multimedia para apoyar el proceso de aprendizaje del tema de auditorías y revisiones en la UCI, apoyándose en la confección de un mapa conceptual que contiene los principales conceptos que se relacionan con el tema. También se analizan las herramientas que puedan ser útiles para el desarrollo de la misma y se da una visión de la distribución de su contenido de la misma y de los lenguajes usados para el manejo de datos y programación.

ÍNDICE

Introducción.....	9
Capítulo 1: Fundamentación Teórica.....	15
Introducción.....	15
1.1 <i>Definición de Calidad</i>	16
1.1.1 Definición de calidad de software.....	16
1.2 <i>Control de la Calidad de Software</i>	16
1.3 <i>Conceptos Auditorías y Revisiones</i>	17
1.3.1 Objetivos de las Auditorías y Revisiones.....	18
1.3.2 Ventajas de auditorías y revisiones.....	18
1.3.3 Swebook.....	18
1.4 <i>Software Educativo</i>	22
1.5 <i>Definición de Multimedia</i>	22
1.5.1 Definición de Multimedia Interactiva.....	23
1.5.2 Definición de Hipermedia.....	23
1.5.3 Definición de Hipertexto.....	23
1.6 <i>Tendencias de tecnologías actuales</i>	24
1.6.1 Herramientas para el desarrollo Multimedia.....	25
1.6.2 Herramienta a utilizar.....	28

INDICE

1.6.3 Metodología de desarrollo y lenguaje de modelado	28
1.6.4 Lenguajes utilizados en la multimedia	31
1.7 <i>Análisis de las herramientas existentes</i>	33
<i>Conclusiones</i>	33
Capítulo 2: Descripción de la Solución	34
Introducción	34
2.1 <i>Especificación del contenido</i>	34
2.1.1 Mapa conceptual	35
2.2 <i>Modelo de dominio</i>	36
2.2.1 Diagrama de clases del modelo de dominio	37
2.3 <i>Mapa de navegación</i>	38
2.4 <i>Descripción de la funcionalidad</i>	39
2.4.1 <i>Requisitos funcionales</i>	40
2.4.2 <i>Requisitos no funcionales</i>	40
2.4.3 <i>Requisitos de apariencia</i>	40
2.4.4 <i>Requisitos de usabilidad</i>	40
2.4.5 <i>Requisitos de software</i>	40
2.4.6 <i>Requisitos de Hardware</i>	41
2.4.7 <i>Requisitos de diseño e implementación</i>	42
2.4.8 <i>Requisitos de soporte</i>	42

INDICE

<i>2.5 Modelo de casos de uso del sistema</i>	42
2.5.1 Actores del sistema.	42
2.5.2 Diagrama de casos de uso del sistema.....	43
2.5.3 Breve resumen de los casos de uso del sistema.	44
<i>Conclusiones</i>	47
Capítulo 3: Solución de la Descripción Propuesta.....	48
3.1 Principios y normas de diseño.....	48
3.2 Descripción de archivos XML.....	49
3.3 Diagramas de Presentación del modelo de Análisis.....	52
3.4 Modelo de Despliegue.....	55
3.5 Diagrama de Componentes.....	56
<i>Conclusiones</i>	57
CONCLUSIONES GENERALES	58
RECOMENDACIONES	59
Referencias Bibliográficas	60
Anexos	64
Glosario	67

INTRODUCCIÓN

En la actualidad, los negocios de todo el mundo han ingresado a la era de la información. Las organizaciones de todo tipo y tamaño utilizan las tecnologías de la información para ser más eficientes. En este contexto global, la actividad de desarrollo de software resulta ser primordial para satisfacer estas nuevas necesidades del mercado, necesidades crecientes y cambiantes, lo que genera una demanda de las mismas características.

En el mundo actual, en el que se necesita obtener productos con alta calidad, que integran varias tecnologías y cuyo tiempo de elaboración sea mínimo, se impone mejorar el proceso de desarrollo de software, incrementando la productividad de los equipos involucrados en el desarrollo de los proyectos, (Humphrey,1997).

El software de computadora se ha convertido en el alma mater. Es la máquina que conduce a la toma de decisiones comerciales. Sirve de base para la investigación científica moderna y de resolución de problemas de ingeniería. Es el factor clave que diferencia los productos y servicios modernos. Está inmerso en sistemas de todo tipo: de transporte, médicos, de telecomunicaciones, militares, procesos industriales, entretenimientos, productos de oficina..., la lista es casi interminable. El software es casi ineludible en un mundo moderno. A medida que nos adentremos en el siglo XXI, será el que nos conduzca a nuevos avances en todo, desde la educación elemental hasta la ingeniería genética.

En el mundo la mayoría de las empresas, universidades, compañías, industrias y otras organizaciones productoras de software presentan problemas en la realización del producto de software, los problemas son:

- Los proyectos terminan costando 1.4 veces más que el presupuesto inicial.
- La duración de los proyectos se prolonga 1.83 veces más que el plazo estimado.
- Terminan implementándose con un 53% de las funcionalidades definidas inicialmente.
- No se garantiza y controla la calidad.
- Revisados por primera vez en las pruebas de liberación y/o aceptación.
- Los productos requieren de varias iteraciones para lograr su liberación y/ aceptación.

INTRODUCCION

- La mayoría no cumple con los requisitos documentados o se encuentran defectos de gran envergadura.
- Se elevan potencialmente el esfuerzo y el costo de los productos por corrección de defectos.
- Poca retroalimentación para los directivos.

Las industrias desarrolladoras de software están haciendo un mejor trabajo en la creación de los productos con respecto a hace 12 años atrás, según el informe de la revista Chaos del 2006, que pertenece al grupo de Standish Group (1 de marzo de 2007). A continuación se muestran algunos de estos datos.

Descripción	1994	2006
Proyectos completados en tiempo y según los costos previstos	16.2%	35%
Proyectos cancelados	31%	19%
Proyectos con grandes problemas de desarrollo	53%	46%

El desarrollo de una Industria Nacional de Software es una tarea de gran prioridad para el estado cubano, debido a la alta perspectiva económica que posee, así como para el aseguramiento de un grupo importante de actividades del país, (Moreno,2003).

La situación actual y las perspectivas de la industria de software cubana están caracterizadas por el trabajo que se viene realizando en materia de capacitación y en la inserción tanto en el mercado nacional como en el internacional.

El desarrollo de la industria del software en el mundo y en particular en nuestro país, exige la aplicación de procedimientos, estándares y modelos que garanticen la calidad del software elaborado. Esto requiere que las universidades garanticen la formación de profesionales informáticos diestros en la producción de software con un alto conocimiento sobre las mejores prácticas internacionales, (Febles, 2005)

INTRODUCCION

La Universidad de las Ciencias Informáticas (UCI) es una Universidad que surgió al calor de la batalla de ideas, en la cual hoy se forman profesionales calificados para la producción de software.

En la UCI se realizan los procesos de auditorías y revisiones en los proyectos productivos de la universidad, con el objetivo de mejorar la calidad de los productos. Estos procesos son de vital importancia ya que están diseñados para el descubrimiento temprano de defectos; aseguran el cumplimiento de procedimientos y estándares establecidos; buscan las soluciones de los problemas y evaluación sistemática de las 4 P (Producto, Proceso, Proyecto y Personas).

A partir de las entrevistas realizadas a líderes de proyectos, vicedecanos de producción, asesores de calidad, jefes de polo, entre otros factores, por consultores del SIE Center de Monterrey, México, en la UCI, en noviembre del 2008, se detectaron los siguientes problemas en los proyectos productivos:

- Problemas en la realización de la revisión sistemática de todos los procesos de realización del software.
- Problemas, deficiencias en la revisión de la realización de los productos y los servicios de trabajo.
- En el transcurso del desarrollo del software, no se está realizando un trabajo de acuerdo con los objetivos planteados por falta de comunicación entre el desarrollador y el supervisor.
- Algunos de estos problemas no eran registrados o solamente no había constancia de ellos.
- Al personal no se le imparte cursos o clases de superación para su capacitación en su puesto de trabajo.
- No se está realizando una buena gestión y configuración de los proyectos.
- El plan de trabajo del proceso de realización del producto no se está realizando correctamente.

INTRODUCCION

En la UCI existe un grupo de trabajo que se dedica a la realización de auditorías y revisiones de software (Grupo de Auditorías y Revisiones) a todos los proyectos productivos como son AKademos, Simulador Quirúrgico, UCI TV, Laboratorios Virtuales, entre otros. En estas auditorías y revisiones se detectaron deficiencias en:

- La definición y empleo de las herramientas a utilizar en el desarrollo del proyecto
- La gestión de los requisitos
- El completamiento del Expediente de Proyecto
- El establecimiento de la gestión de configuración
- El establecimiento del plan de aseguramiento de la calidad
- El establecimiento y cumplimiento del cronograma del proyecto

Con el surgimiento de las auditorías y revisiones de software como actividades de control de la calidad, la mayoría de las industrias y empresas desarrolladoras de software ha logrado obtener una mayor calidad en la elaboración del producto.

Hoy en la UCI hoy se presentan grandes problemas en los procesos de auditorías y revisiones de software, por lo que ha surgido la necesidad de mejorar la capacitación de los estudiantes y profesores que participan en dichos procesos, para mejorar la calidad de los productos de software que se desarrollan; y para ésto se hace necesario la capacitación del personal por medio de cursos, clases, videoconferencias, documentos, libros, etc.

Luego del análisis de los problemas anteriormente planteados, el presente trabajo define como **problema científico**:

¿Cómo sustentar la inclusión de buenas prácticas en el proceso de enseñanza de Auditorías y Revisiones de software a través de las TIC?

El **objeto de estudio** de este trabajo son los medios para el aprendizaje de las Auditorías y Revisiones.

El **campo de acción**:

Proceso de desarrollo de una Multimedia Interactiva que apoye el proceso de enseñanza - aprendizaje de las auditorías y revisiones de software en la UCI.

INTRODUCCION

Para dar solución al problema científico planteado se define como **objetivo general**:

Implementar una multimedia interactiva que apoye el proceso de enseñanza -aprendizaje del tema auditorías y revisiones de software en la UCI.

Para el cumplimiento del objetivo planteado se realizarán las siguientes **tareas científicas**:

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

Durante el desarrollo de esta investigación se utilizarán varios **métodos teóricos**:

Histórico lógico: se utilizó para la realización de una investigación a fondo de los procesos de auditorías y revisiones de software.

Analítico-sintético para el procesamiento de la documentación recopilada durante la investigación.

Como **método empírico** se utilizó:

Entrevista a profesores y especialistas, específicamente a aquellos que participan en el proceso de auditorías y revisiones de software en la UCI.

INTRODUCCION

El contenido de toda la investigación esta estructurado de la siguiente forma:

Capítulo 1: “Fundamentación teórica.”

En este capítulo se resumen todos los conceptos relacionados con el objeto de estudio, así como las tendencias, tecnologías actuales, herramientas y lenguajes de programación utilizados en el desarrollo del software.

Capítulo 2: “Descripción de la solución propuesta”

En este capítulo se realiza una especificación del contenido de la multimedia. Se hace el mapa de navegación y el levantamiento de los requerimientos funcionales y no funcionales. Se realiza además el modelado del sistema, así como la descripción de los casos de uso correspondientes.

Capítulo 3: “Solución de la descripción propuesta.”

Este capítulo presenta el proceso de la construcción del sistema a desarrollar. Incluye la realización de los diagramas de presentación, modelo de despliegue y diagrama de componentes. También se abordan algunos archivos XML utilizados en la multimedia.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Introducción

En el presente capítulo se definen conceptos claves como calidad, calidad de software, auditorías y revisiones. Se realizará un estudio de las posibles metodologías, herramientas y lenguajes a utilizar en la construcción del software y se tendrán en cuenta las tendencias, tecnologías actuales, así como los conceptos fundamentales relacionados con este tema.

1.1 Definición de Calidad.

La palabra calidad tiene múltiples significados. Es un conjunto de propiedades inherentes a un objeto que le confieren capacidad para satisfacer necesidades implícitas o explícitas. La calidad de un producto o servicio es la percepción que el cliente tiene del mismo. Es una fijación mental del consumidor que asume conformidad con dicho producto o servicio y la capacidad del mismo para satisfacer sus necesidades.

La calidad se puede definir como "una característica o atributo de una cosa". De esta forma se podría decir que la calidad de los productos puede medirse como una comparación de sus características y atributos. Así, este concepto puede aplicarse a cualquier producto.

El autor considera el concepto de calidad más completo dado por ISO 8402. En esta norma se define calidad como el "Conjunto de propiedades y de características de un producto o servicio, que le confieren aptitud para satisfacer las necesidades explícitas o implícitas".

1.1.1 Definición de calidad de software.

La obtención de un (software) con calidad implica la utilización de metodologías, procedimientos, estándares para el análisis, diseño, programación y prueba del (software) que permitan uniformar la filosofía de trabajo, en aras de lograr una mayor confiabilidad, mantenibilidad y facilidad de prueba, a la vez que eleven la productividad, tanto para la labor de desarrollo como para el control de la calidad del (software).

De acuerdo a la definición del Instituto de Ingenieros Eléctricos y Electrónicos (IEEE, Std. 610-1990) “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”.

Hay factores para Pressman que determinan la Calidad del Software, tales como:

- Corrección. ¿Hace lo que quiero?
- Fiabilidad. ¿Lo hace de forma fiable todo el tiempo?
- Eficiencia. ¿Se ejecutará en mi hardware lo mejor que pueda?
- Seguridad (Integridad). ¿Es seguro?
- Facilidad de uso. ¿Está diseñado para ser usado?

La calidad del software no es más que la concordancia del software producido con los requerimientos explícitamente establecidos, con los estándares de desarrollo prefijados y con los requerimientos no establecidos formalmente, que desea el usuario (*Pressman, 2002*).

1.2 Control de la Calidad de Software.

El control de la calidad no es más que una serie de inspecciones, revisiones y pruebas realizadas a lo largo del ciclo de desarrollo del software para asegurar que el producto cumpla con los requisitos que le han sido asignados.

1.3 Conceptos de Auditorías y Revisiones.

Auditorías:

El origen etimológico de la palabra es el verbo latino "Audire", que significa "oír". La *Norma ISO 9000:2000 - Fundamentos y vocabulario* define como concepto de Auditoría: "El Proceso sistemático, independiente y documentado para obtener evidencias de la auditoría y evaluarlas de manera objetiva con el fin de determinar la extensión en que se cumplen los criterios de auditoría".

En resumen la auditoría no es más que un examen independiente de un producto de software, proceso del software, o sistema de procesos del software para determinar conformidad con especificaciones, estándares, acuerdos contractuales, u otros criterios.

Revisiones:

Las revisiones de software sirven para validar la calidad y/o el estado de un producto. Este producto a revisar puede ser un documento, un módulo, un prototipo, etc.

Las revisiones son un proceso o reunión durante el cual un producto de software se presenta a personal del proyecto, administradores, usuarios, clientes u otras partes interesadas para comentario o aprobación. (*IEEE Std 1028 - 1997 IEE*)

En resumen las revisiones no son más que actividades de control de calidad que permiten detectar defectos en los proyectos de software y ayudan a encontrar falencias, y a identificar riesgos que serían difíciles de encontrar de otra manera.

1.3.1 Objetivos de las Auditorías y Revisiones.

Los objetivos fundamentales de las **auditorías** son:

- Acumular y evaluar las evidencias.
- Determinar e informar sobre el grado de correspondencia existente entre la información cuantificable y los criterios de evaluación establecidos.

Los objetivos fundamentales de las **revisiones** son:

- Indicar la necesidad de mejoras en un producto.
- Indicar las partes que no es necesario mejorar.
- Evaluar conformidad con estándares y especificaciones técnicas.
- Descubrimiento de defectos e inconsistencias en el sistema.

1.3.2 Ventajas de auditorías y revisiones.

Las auditorías y revisiones proporcionan varias ventajas en el desarrollo de un software, como son:

- Asegura el cumplimiento de procedimientos y estándares establecidos.
- Descubrimiento temprano de defectos.
- Evaluación sistemática de las 4 P (Producto, Proceso, Proyecto y Personas)
- Busca la solución de los problemas.
-

1.3.3 Análisis del Swebok

Con el propósito de ser breve, las revisiones y auditorías son tratadas como un solo tópico en la Guía del Cuerpo del Conocimiento de la Ingeniería de Software (Swebok), en lugar de dos tópicos como se hace en (IEEE12207.0-96). El proceso de revisión y auditoría se define en

términos generales en (IEEE12207.0-96), y en más detalle en (IEEE1028-97). Cinco tipos de revisiones o auditorías se presentan en el estándar IEEE1028-97:

- Revisiones de gestión
- Revisiones técnicas
- Inspecciones
- Walk-throughs
- Auditorías

Revisiones de gestión

"El propósito de una revisión de gestión es supervisar el progreso, determinar el estado de los planes y programas, confirmar los requisitos y sus localizaciones en el sistema para alcanzar la idoneidad del objeto "[IEEE1028-97]. Ellos apoyan las decisiones acerca de los cambios y medidas correctivas que se requieren durante un proyecto de software. Las revisiones de gestión determinan la idoneidad de los planes, programas, y requerimientos y supervisan su progreso o inconsistencias. Estas revisiones podrán realizarse en productos tales como los informes de auditoría, informes de progreso, informes V & V, y proyectos de muchos tipos, incluyendo los de gestión de riesgo, gestión de proyectos, la gestión de configuración del software, seguridad del software, y la evaluación de riesgos, entre otros.

Revisiones técnicas

"El propósito de una revisión técnica es evaluar un producto de software para determinar si es idóneo para su correspondiente uso. El objetivo es identificar las discrepancias de las especificaciones aprobadas y estándares aprobados. Los resultados deben ofrecer gestión con evidencias que confirmen (o no) que el producto cumpla con las especificaciones y se adhiere a las normas, y que los cambios son controlados "(IEEE1028 -97).

Los roles específicos que se establecen en una revisión técnica son: el que toma las decisiones, un revisor líder, un registrador, y un equipo técnico para apoyar las actividades de revisión. Una revisión técnica requiere que las entradas obligatorias estén situadas en su lugar para proceder a:

- Planteamiento de objetivos
- Un producto de software específico
- El plan específico de gestión del proyecto
- La lista de aspectos asociados al producto
- El procedimiento de revisión técnica

El equipo sigue el procedimiento de revisión. Un individuo técnicamente calificado presenta una descripción general del producto, y el examen se realiza durante uno o más reuniones. La revisión técnica se completa una vez que todas las actividades catalogadas en el examen se han completado.

Inspecciones

"La finalidad de una inspección es detectar e identificar anomalías del producto de software "(IEEE1028-97). Existen dos importantes elementos diferenciadores entre inspección y revisión, que son los siguientes:

1. Un individuo, que mantiene una posición de dirección sobre cualquier miembro del equipo de inspección no puede participar en la inspección.
2. Una inspección, debe ser guiada por un facilitador imparcial quien está entrenado en las técnicas de inspección.

Las inspecciones de software siempre implican al autor del producto intermedio o final, mientras que otras revisiones puede que no. Las inspecciones también incluyen un líder de inspección, un registrador, un lector, y unos pocos inspectores (de 2 a 5). Los miembros de un equipo de inspección pueden tener diferentes niveles de especializaciones, tales como especialización en el dominio, especialización en el método de diseño, o especialización en el lenguaje. Las inspecciones son usualmente dirigidas a una pequeña parte del producto en un momento. Cualquier anomalía encontrada es documentada y es enviada al líder de la inspección. Durante la inspección, el responsable de la inspección conduce la sesión y verifica que todos estén preparados para la misma. Una herramienta comúnmente utilizada en las inspecciones es una

lista de comprobación, con las anomalías y las preguntas pertinentes para las cuestiones de interés. En la lista resultante a menudo se clasifican las anomalías (consulte IEEE1044-93 para más detalles) y es revisada para su completamiento y precisión por parte del equipo. La decisión sobre el final de la inspección se corresponde a uno de los tres criterios siguientes:

1. Aceptar sin re-trabajo o con el menor re-trabajo.
2. Aceptar con verificación del re-trabajo.
3. Re-inspección.

Las reuniones de inspección suelen durar algunas horas, mientras que las revisiones técnicas y las auditorías tienen por lo general un alcance más amplio y toman más tiempo.

Walk-throughs

"El propósito de un Walk-throughs es evaluar un producto software. Un Walk-throughs puede llevarse a cabo con el propósito de formar a una audiencia en relación con un producto software". (IEEE1028-97) Los principales objetivos son [IEEE1028-97]:

- Encontrar anomalías
- Mejorar el producto de software
- Considerar implementaciones alternativas
- Evaluar la conformidad con estándares y especificaciones

Un Walk-throughs es similar a una inspección, pero es llevado a cabo normalmente de forma menos formal. El Walk-throughs es organizado principalmente por el ingeniero de software para dar a su equipo la oportunidad de revisar su trabajo, como una técnica de aseguramiento

Auditorías

"El propósito de una auditoría de software es proporcionar una evaluación independiente a la conformidad del producto y los procesos de sus regulaciones aplicables, estándares, directrices, planes y procedimientos "[IEEE1028 - 97]. La auditoría es una actividad organizada formalmente,

con participantes que tienen funciones específicas, tales como el auditor principal, otro auditor, un registrador o un iniciador, e incluye un representante de la organización auditada. La auditoría identificará los casos de no conformidad y producirá un informe el cual requiere que tome medidas correctivas correspondientes. Si bien puede haber muchos nombres oficiales para las revisiones y las auditorías como los mencionados en la norma (IEEE1028 - 97), el punto importante es que las revisiones y auditorías pueden practicarse sobre casi cualquier producto en cualquier etapa del desarrollo o proceso de mantenimiento.

1.4 Software Educativo

Podemos definir al *software educativo* o *material educativo con TIC's* como todo aquel objeto o software desde el cual podemos aprovechar sus características para el logro de aprendizajes. Imágenes, sonidos, textos desde un PC, pueden ser utilizados para cualquier área de conocimiento, independiente de para que estos software hayan sido diseñados inicialmente. (Gerardo González García, 2007).

1.5 Definición de Multimedia

Una multimedia es la combinación de diferentes medios de comunicación, tales como sonido, gráficos, animación y vídeo, utilizados por un procesador en su relación con las personas. Combinación de varias tecnologías de presentación de información (imágenes, sonido, animación, video, texto) con la intención de captar tantos sentidos humanos como sea posible. Previamente a la existencia de la multimedia, el intercambio de información con las computadoras estaba limitado al texto. Luego, con el nacimiento de las interfaces de usuario gráficas y los desarrollos en video y sonido, la multimedia permitió convertir el modo de comunicación entre personas y dispositivos aumentando la variedad de información disponible.

Se puede concluir con que se denomina multimedia, a una colección de tecnologías basadas en la utilización de la computadora, que da al usuario la capacidad de acceder y procesar información en forma de texto, gráficas, imagen fija, imagen con movimiento y audio (*Elda Quiroga, 15 de noviembre del 2004*)

1.5.1 Definición de Multimedia Interactiva

El concepto de multimedia ha ido evolucionando a través del tiempo. Al principio se trataba de la simple sumatoria de medios distintos: textos, sonidos e imágenes - animaciones, videos, gráficos, ilustraciones, fotos, en todas sus posibles combinaciones. Sin embargo, hoy en día muchas veces se usa multimedia para referirse al, más acertadamente denominado, multimedia interactivo

Desde el punto de vista del usuario, interactividad es la cantidad de control que éste tiene sobre los contenidos. Esta definición alude a los grados de interactividad que puede tener el producto. El más bajo de estos niveles de interactividad es el agotado y agotador recurso de teclear una y otra vez enter, lo que algunos, con indudable gracia, han dado en llamar interactividad. La verdadera interactividad, en cambio, implica una modificación en la propia estructura del "relato" multimedia: se pasa de una presentación lineal, donde todo el control es del autor, a la multilinealidad donde el control es del usuario. El usuario es así el dueño de la secuencia y del tiempo dedicado al contenido. (*By multimediam - Posted on 27 Febrero 2006*).

1.5.2 Definición de Hipermedia.

Una hipermedia no es más que materiales con soporte informático, que se caracterizan por permitir enlazar de forma interactiva las diversas informaciones que contienen. También se caracteriza porque dicha información puede ser presentada a través de diferentes códigos simbólicos y no se navegaba de forma lineal.

1.5.3 Definición de Hipertexto.

El hipertexto puede considerarse como un subtipo de materiales hipermedia, puesto que el término parece implicar que los datos dentro del mismo son sólo textuales. (Informática y multimedia educativo, 14 Mayo, 2008).

1.6 Tendencias de tecnologías actuales

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 requerimiento, 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.

Existen varias herramientas de apoyo al desarrollo de los entornos virtuales de aprendizaje, algunas de estas herramientas se exponen a continuación.

- **Moodle:** es un sistema de gestión de cursos, de distribución libre, que ayuda a los educadores a crear comunidades de aprendizaje en línea. Este tipo de plataformas tecnológicas también se conocen como LMS (Learning Management System).

Moodle es una herramienta poderosa para diseñar y producir cursos en línea, basados en la utilización de Internet y páginas Web. Esta plataforma fue diseñada basándose en el enfoque del constructivismo social (*SERUNAM, Junio 2007*)

- **WebCT:** es una plataforma que facilita la creación de un sofisticado entorno educacional basado en la World Wide Web. Puede usarse para crear un curso completo *on-line*, o simplemente para publicar material suplementario a la docencia presencial. WebCT ofrece un conjunto de herramientas de contenidos, comunicación (foro, chat, correo), evaluación (autoevaluación, corrección automática, etc. cuyos resultados se salvaguardan en áreas protegidas mediante password para su posterior procesado estadístico), seguimiento y otras utilidades como un calendario. Aunque se echa en falta la integración de otras herramientas

(la videoconferencia), las sucesivas versiones incorporan mejoras como el manejo simultáneo de archivos (*WebCT 4.1 Manual, noviembre 2004*).

1.6.1 Herramientas para el desarrollo Multimedia

Existen varias herramientas que se utilizan para el desarrollo de multimedia. A continuación se mencionarán algunas de ellas:

- **Toolbook:** es un programa que permite realizar aplicaciones Windows. Adicionalmente, posee el lenguaje de programación OpenScript orientado a objetos que enriquece extraordinariamente sus posibilidades en la generación de aplicaciones multimedia. Los sistemas de autor como Toolbook permiten diseñar una amplia variedad de actividades de distinta índole, combinando textos, imágenes, sonidos, animaciones, vínculos a sitios Web y autoevaluaciones en distintos formatos.

Toolbook Instructor es considerada una de las herramientas de autoría e-learning líderes a nivel mundial. Más de 10.000 empresas e instituciones educativas, a lo largo de los cinco continentes, desarrollan sus contenidos e-learning con Toolbook.

Concluimos que Toolbook es una herramienta muy poderosa en la que se pueden desarrollar diversas actividades tales como entrenamientos, simulaciones y es especialmente útil en el campo educativo por su interactividad y amplio rango de posibilidades.

- **Authorware:** es un lenguaje de programación gráfico interpretado basado en diagramas de flujo. Authorware se usa para crear programas interactivos que pueden integrar sonidos, texto, gráficos, animaciones simples, y películas digitales. Authorware es una herramienta utilizada principalmente para crear los medios que aprenden los usos para la entrega en las redes corporativas, CD/DVD, y el Web.

Loa authorware se usan principalmente para crear productos multimedia interactivos e instructivos, pero también se pueden utilizar para el desarrollo de prototipos de productos multimedia.

- **Revolution:** Revolution es una herramienta de desarrollo que se destaca, sin lugar a dudas, porque permite crear aplicaciones con una interfaz de usuario para la mayoría de las plataformas existentes en nuestros días. La herramienta de autor Revolution es la hermana joven de MetaCard. Se desarrolla sobre MetaCard y es una respuesta a lo que en muchas ocasiones se ha lamentado de su "hermana": su austera interfaz. El entorno de trabajo deja de ser austero para pasar a ser totalmente visual en la disposición de las opciones y el modo de trabajo más cuidadoso desde el punto de vista estético. Esta aplicación está disponible en las plataformas soportadas por MetaCard y tiene un lenguaje de programación de alto nivel orientado a objetos llamado "Transcript". Esta herramienta permite proyectar y desarrollar aplicaciones fácil y rápidamente. (BELLONCH, 2006)
- **Adobe Director:** Es una aplicación en forma de estudio (que inspiró a Adobe Flash) destinado a la producción de películas ejecutables en Adobe Shockwave, usando para ello mapas de bits y programación Lingo. Es actualmente distribuido por Adobe Systems Incorporated

Este software permite generar presentaciones multimedia (en archivos ejecutables, por ejemplo) que pueden ser distribuidas a través de CD. Permite incorporar a las películas múltiples tipos de medios como imágenes (jpeg, bmp, png, gif, psd, tiff, etc.), vídeos (mov, avi, etc.), sonidos (wav, aiff, etc.) o animaciones Flash. Incluye editores básicos para texto, mapa de bits, vectores y sonido.

- **Flash 8:** es una potente herramienta creada por Macromedia que ha superado las mejores expectativas de sus creadores. Inicialmente, Macromedia Flash fue creado con el objeto de realizar animaciones vistosas para la web, así como para crear GIFS animados.

Flash 8 también posibilita mejoras en cuanto a facilidad de manejo, mayor potencia gráfica y de integración con programas de edición de imágenes. Facilidad para importar vídeo. Posibilidad de emular tus películas dirigidas a dispositivos móviles y, para los menos avanzados, se recupera el asistente de ActionScript que había desaparecido en la versión

anterior... A continuación se analizan estas ventajas y otras muchas con un poco más de detalle:

- **Diseños más atractivos:** Flash 8 permite el uso de efectos visuales que nos facilitarán la creación de animaciones, presentaciones y formularios más atractivos y profesionales. Así mismo se, ponen a nuestra disposición mecanismos para hacer este trabajo más cómodo y rápido, tales como la existencia de filtros y modos de mezcla.
- **Optimización de fuentes:** Incorpora también opciones de legibilidad para fuentes pequeñas, haciendo la lectura de textos más agradable y de alta legibilidad. Además de poder modificar la optimización, Flash permite también la selección de configuraciones preestablecidas para textos dinámicos y estáticos.
- **Bibliotecas integradas:** Ahora se puede buscar rápidamente cualquier objeto existente en películas, navegando por las bibliotecas de todos los archivos abiertos desde un único panel.
- **Mayor potencia de animación:** Flash 8 permite un mayor control de las interpolaciones, habilitando un modo de edición desde el que se podrá modificar independientemente la velocidad en la que se apliquen los diferentes cambios de rotación, forma, color, movimiento, etc., de las interpolaciones. +
- **Mayor potencia gráfica:** Evita la repetición innecesaria de la representación de objetos vectoriales, simplemente señalando un objeto como mapa de bits. Aunque el objeto se convierta al formato de mapa de bits, los datos vectoriales se mantienen tal cual, con el fin de que, en todo momento, el objeto pueda convertirse de nuevo al formato vectorial.
- **Mejoras en la importación de vídeo:** Para facilitar el resultado con formatos de vídeo, Flash 8 incluye un códec independiente de calidad superior, capaz de competir con los mejores códecs de vídeo actuales con un tamaño de archivo mucho más pequeño.
- **Compatibilidad Metadatos:** Incluye tus SWF en buscadores de Internet con la nueva característica de definición de archivo con un título, una descripción y/o palabras clave para que los motores de búsqueda reflejen con más precisión el contenido representado por el archivo.
- **Asistente de ActionScript:** El Asistente de ActionScript ha vuelto. Fue eliminado en la versión anterior, pero se ha vuelto a recuperar, y de forma mejorada, en esta última.

1.6.2 Herramienta a utilizar

Para la creación de la multimedia interactiva de auditorías y revisiones se escogió como herramienta de autor Macromedia Flash 8, debido a sus ventajas y características expuestas anteriormente, además de que Flash ha conseguido hacer posible lo que más se echa en falta en Internet: Dinamismo, y con dinamismo no sólo nos referimos a las animaciones, sino que Flash permite crear aplicaciones interactivas que permiten al usuario ver la Web como algo atractivo, no estático. Con Flash se pueden crear, de modo fácil y rápido, animaciones de todo tipo. Flash es fácil de aprender. Tiene un entorno amigable.

1.6.3 Metodología de desarrollo y lenguaje de modelado

Existen varias metodologías y varios lenguajes de modelado que se pueden utilizar para el desarrollo del producto. Algunas de ellas se muestran a continuación:

Metodologías:

- Metodología de Administración de Relaciones (RMM).
- Programación extrema (XP).
- Proceso Unificado de Software (RUP).

Lenguajes:

- Lenguaje unificado de modelado (UML).
- Lenguaje orientado a objetos para el modelado de aplicaciones multimedia (OMMA-L).

Como **metodología** para el desarrollo del producto se seleccionó **RUP**, ya que es un proceso unificado que guía a los equipos de proyecto en cómo administrar el desarrollo iterativo de un modo controlado, mientras se balancean los requisitos del negocio, el tiempo al mercado y los riesgos del proyecto. El proceso describe los diversos pasos involucrados en la captura de los requisitos y en el establecimiento de una guía arquitectónica lo más pronto, para diseñar y probar el sistema hecho de acuerdo a los requisitos y a la arquitectura. El proceso describe qué entregables producir, cómo desarrollarlos y también provee patrones. El proceso unificado es

soportado por herramientas que automatizan, entre otras cosas, el modelado visual, la administración de cambios y las pruebas.

El RUP (Proceso Unificado de Rational) se fundamenta en seis prácticas:

- El desarrollo iterativo
- La administración de requisitos
- La arquitectura basada en componentes
- El modelado visual
- La verificación continua de la calidad
- La administración del cambio

Estas seis prácticas orientan el modelo y con ellas se pretende solucionar muchos de los problemas asociados al software. Adicionalmente hay muchos aspectos de diseño que son bien conocidos, pero que en realidad han sido muy poco implementados en los proyectos de software; estos son: facilidad de uso, modularidad, encapsulamiento y facilidad de mantenimiento. Es necesario entonces definir una arquitectura sólida basada en componentes, para construir mejores y más flexibles soluciones de software para las necesidades organizacionales. Los cambios en un proyecto no pueden ser detenidos, dado que la evolución del entorno de cada organización es continua; pero sí pueden ser administrados de manera que su impacto pueda ser estimado, para determinar si dicho cambio se incluye o no y si el proyecto debe ser reajustado. Cada cambio en el proyecto debe tener especificado cuándo y cómo se va a realizar, quién lo va a hacer y qué productos se ven involucrados en ese cambio. En ese punto es donde el control de cambios y la trazabilidad de los componentes a través de los diversos modelos, adquieren una gran importancia.

Como **lenguaje de modelado** para desarrollar la multimedia se escogió el **Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA - L)**, el cual es una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos; y se escogió además MVC (Modelo Vista Controlador) para la interfaz de usuario, siendo éste un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes

distintos, de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos.

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

- **Vista Lógica:** modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias.
- **Vista de Presentación espacial:** modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagrama tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización (texto, gráfico, video, animación) e interacción (scrolls, barras de menú, botones, campos de entrada y salida, hipertextos con hipervínculos). Estos diagramas de presentación pueden ser divididos en capas virtuales de presentación donde en cada uno de ellas sólo se haga referencia a una clase específica de componentes (por ejemplo, una vista para los objetos de visualización y otra para los de interacción, u otro tipo de división para la representación de los intereses de los desarrolladores.).
- **Vista de Comportamiento temporal predefinido:** modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El Diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos; marcas de inicio y fin de ejecución que permite soportar su reusabilidad; marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media; activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.

- **Vista de Control Interactivo:** modelado a través del Diagrama de Estado. Extendido a partir del diagrama de estado de UML, sintácticamente igual a este último, mas con la diferencia semántica de que en el orden de unir los controles interactivos y predefinidos, no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados; queriendo ésto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

Extendiendo el paradigma MVC para multimedia a las peculiaridades de comportamiento estático y dinámico identificadas anteriormente, se obtiene el MVCMM, sobre el que se basa las especificaciones de OMMMA – L. Actualmente, OMMMA – L se evalúa en diferentes escenarios como proyectos industriales para la especificación de servicios de información multimedia, y se investigan características adicionales de sincronía para su especificación en el lenguaje y la formalización de un modelo para la composición dentro y entre los diferentes diagramas de comportamiento.

1.6.4 Lenguajes utilizados en la multimedia

XML (Lenguaje de Marcas Extensibles): siglas en inglés de “*Extensible Markup Language*” (“lenguaje de marcas ampliable»). Es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.

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

XML es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con unas posibilidades mucho mayores. Tiene un papel muy importante en

la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Ventajas del XML:

- Es extensible: Después de diseñado y puesto en producción, es posible extender XML con la adición de nuevas etiquetas, de modo que se pueda continuar utilizando sin complicación alguna.
- El analizador es un componente estándar. No es necesario crear un analizador específico para cada versión de lenguaje XML; ésto posibilita el empleo de cualquiera de los analizadores disponibles. De esta manera se evitan *bugs* y se acelera el desarrollo de aplicaciones.
- Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarla. Mejora la compatibilidad entre aplicaciones.

ActionScript: es un lenguaje de programación orientado a objetos (OOP), utilizado en especial en aplicaciones web animadas realizadas en el entorno Adobe Flash, la tecnología de Adobe para añadir dinamismo al panorama web. Fue lanzado con la versión 4 de Flash, y desde entonces hasta ahora, ha ido ampliándose poco a poco, hasta llegar a niveles de dinamismo y versatilidad muy altos en la versión 9 (Adobe Flash CS3) de Flash.

ActionScript es un lenguaje de script, esto es, no requiere la creación de un programa completo para que la aplicación alcance los objetivos. El lenguaje está basado en especificaciones de estándar de industria ECMA-262, un estándar para Javascript, de ahí que ActionScript se parezca tanto a Javascript.

La versión más extendida actualmente es ActionScript 3.0, la cual significó una mejora en el manejo de programación orientada a objetos al ajustarse mejor al estándar ECMA-262 y es utilizada en la última versión de Adobe Flash y Flex (recientemente comprada a Macromedia) y en anteriores versiones de Flex. Recientemente, se ha lanzado la beta pública de Flex 2, que incluye el nuevo ActionScript 3, con mejoras en el rendimiento y nuevas inclusiones como el uso de expresiones regulares y nuevas formas de empaquetar las clases. Incluye, además, Flash

Player 8.5, que mejora notablemente el rendimiento y disminuye el uso de recursos en las aplicaciones Macromedia Flash.

1.7 Análisis de las herramientas existentes

Durante la búsqueda del tema específico ,el autor no encontró herramientas que se dediquen al proceso de enseñanza del tema en cuestión.

Durante la búsqueda se encontró que en la actualidad se utilizan los métodos tradicionales para la preparación de personal en el tema de auditorías y revisiones de software, así como universidades donde este tema se da como una asignatura de algunas carreras como Ingeniería, Contabilidad, entre otras. Al concluir la búsqueda no se encontró ningún producto de multimedia que posibilite estudiar enseñar acerca del tema de auditorias y revisiones de software.

En la UCI, el grupo de calidad esté impartiendo un curso de postgrado de auditorías y revisiones de software, con el objetivo de fomentar más este tema en la UCI y lograr una mayor calidad en los productos de software. Este curso se encuentra publicado en el EVA, pero esto no es suficiente ya que no todos podemos cursarlo y es por eso que se necesita una herramienta que todos podamos utilizar para estudiar el tema de auditorías y revisiones de software.

Conclusiones

En este capítulo se ha realizado un estudio de las principales herramientas, metodologías, lenguajes y conceptos que son usados en el desarrollo de una multimedia. Después de haber analizado las ventajas y desventajas de cada uno de ellos, se seleccionó como herramienta de desarrollo Macromedia Flash 8 y se propuso como metodología de desarrollo a RUP. Como lenguaje de modelado a usar OMMMA-L, debido a la gran importancia que tiene la ingeniería en el desarrollo del software. Se seleccionó como lenguaje de programación ActionScript y el XML como lenguaje de marcas, para lograr un mejor desarrollo de la aplicación.

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Introducción

En el presente capítulo se propone describir la solución propuesta para dar solución al problema planteado, por lo que se realiza un análisis del producto y, para ello se define el dominio de la aplicación, sus conceptos asociados, así como el mapa de navegación. Se realiza el levantamiento de los requisitos del sistema. Se obtienen y describen los casos de uso que guiarán la solución del sistema.

2.1 Especificación del contenido

El contenido de esta multimedia está dividido en tres temas fundamentales, los cuales son:

Tema 1: Introducción a los conceptos y técnicas de las auditorías y revisiones.

Este tema va a tratar tres puntos fundamentales, donde el usuario conocerá los principales conceptos, los objetivos, ventajas y limitaciones de los procesos de auditorías y revisiones de software.

Tema 2: Proceso de auditorías y revisiones de calidad.

En el tema Proceso de auditorías y revisiones de calidad se abordarán tres puntos fundamentales. Un primer punto donde se va caracterizar el proceso de auditorías y revisiones de calidad teniendo en cuenta el alcance de. Un segundo punto que se basa en conocer las actividades de dicho proceso; y un tercer punto que aborda la las dificultades de las auditorías y revisiones.

Tema 3: Papel del auditor y el revisor.

En este tema se analizarán dos puntos fundamentales, donde se abordaran las características y definiciones de los trabajadores de las auditorías y revisiones de software, así como sus clientes.

2.1.1 Mapa conceptual.

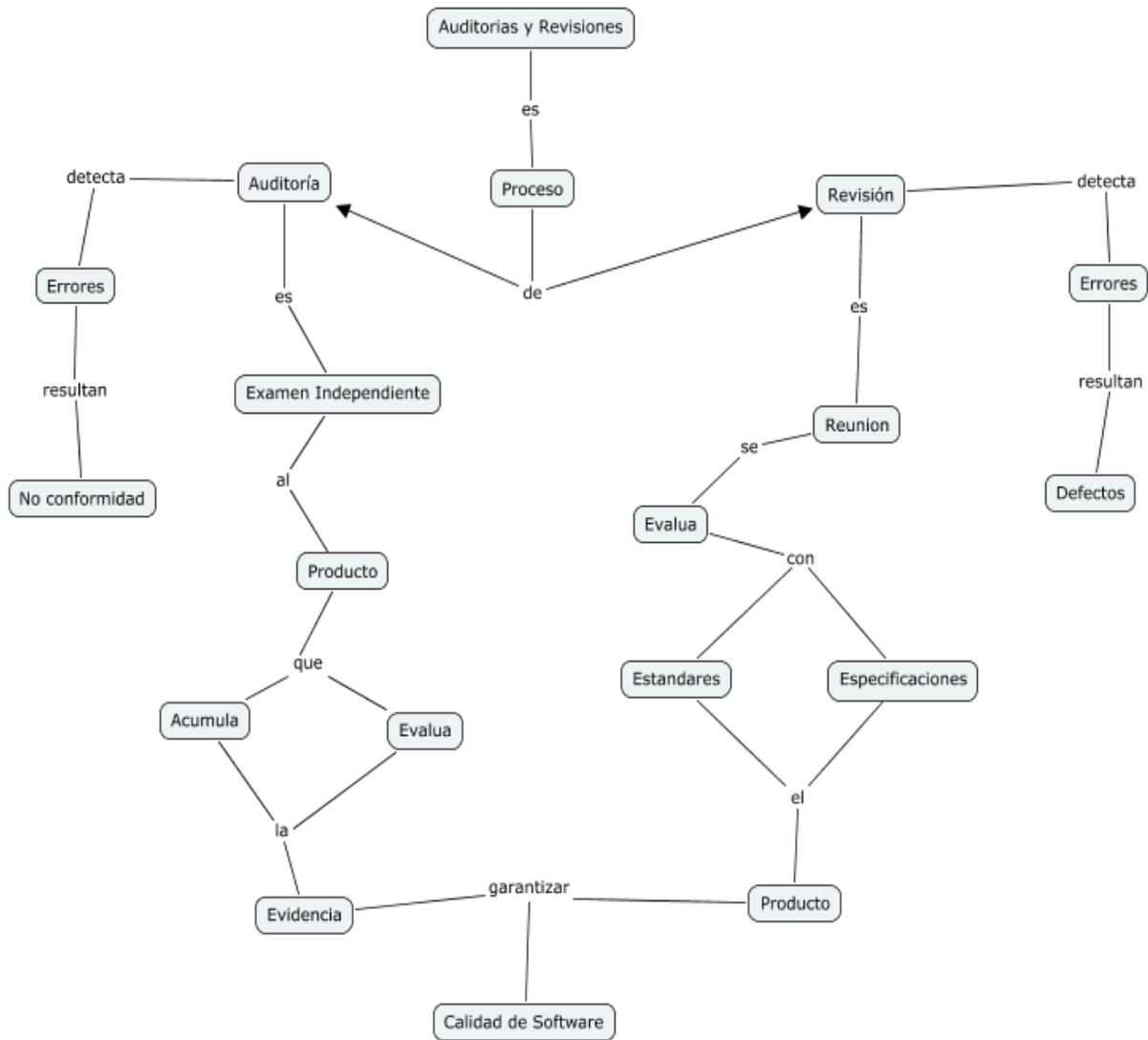


Fig.1 Mapa Conceptual.

2.2 Modelo de 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 2), 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 los conceptos y técnicas de las auditorías y revisiones, proceso de auditorías y revisiones de calidad, dificultades de las auditorías y revisiones; y papel del auditor y el revisor.

Mapas Conceptuales: Forma, de representar visualmente relaciones significativas entre conceptos.

Concepto: Los conceptos son las estructuras o esquemas mentales mediante los cuales se identifican las características esenciales para definir las cosas, hechos o eventos que dan significado a nuestra realidad inherente y circundante.

Videos: Representa un video que abarca el tema de auditorías y revisiones.

Las **Autoevaluaciones** representan un conjunto de ejercicios en los cuales se comprueban los conocimientos adquiridos en los temas.

2.2.1 Modelo de dominio.

El modelo del dominio muestra las principales clases conceptuales que intervienen en el sistema.

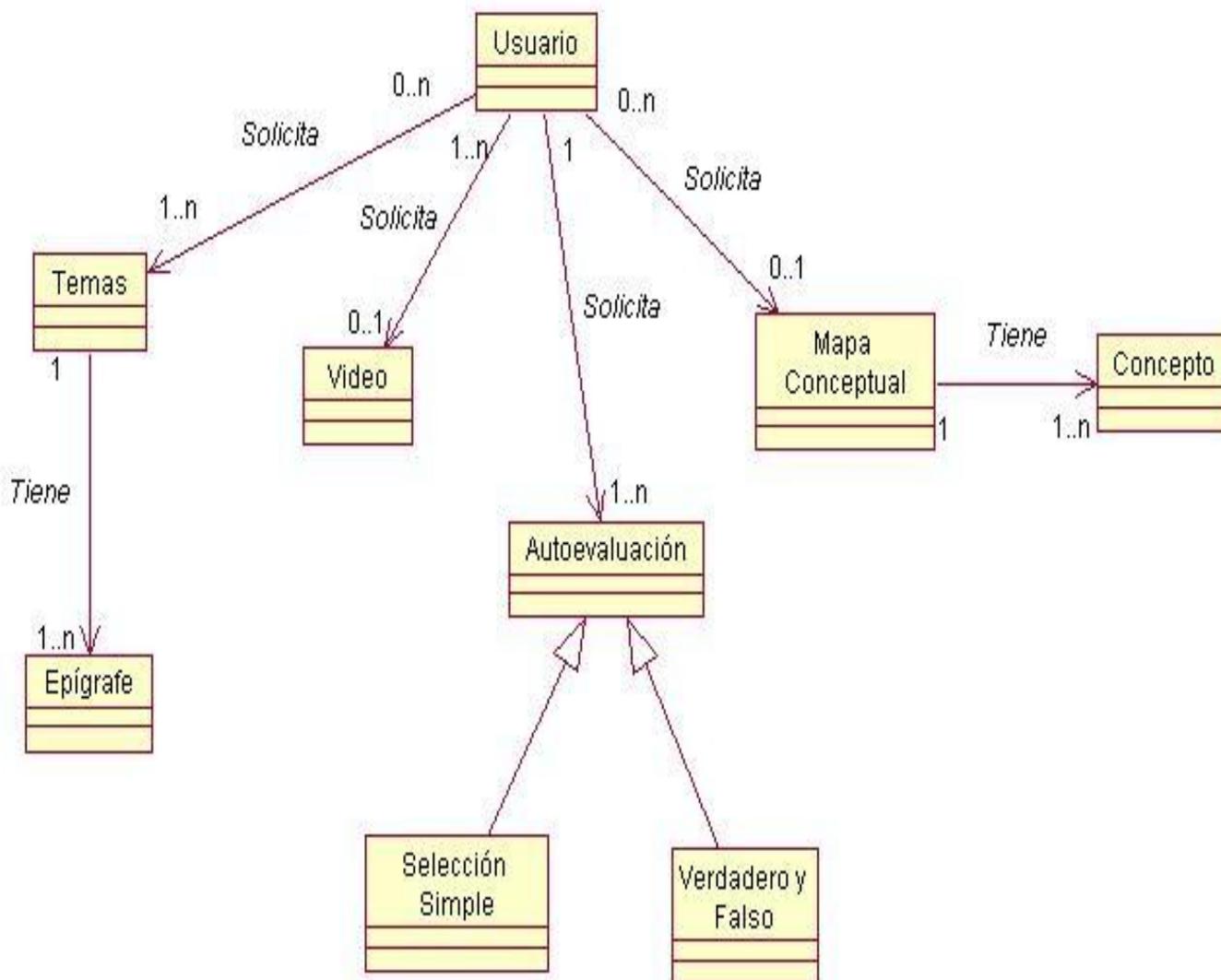


Fig.2 Modelo del Dominio.

2.3 Mapa de navegación.

El mapa de navegación es un diagrama que nos muestra cómo podemos movernos por los diferentes módulos de la aplicación.

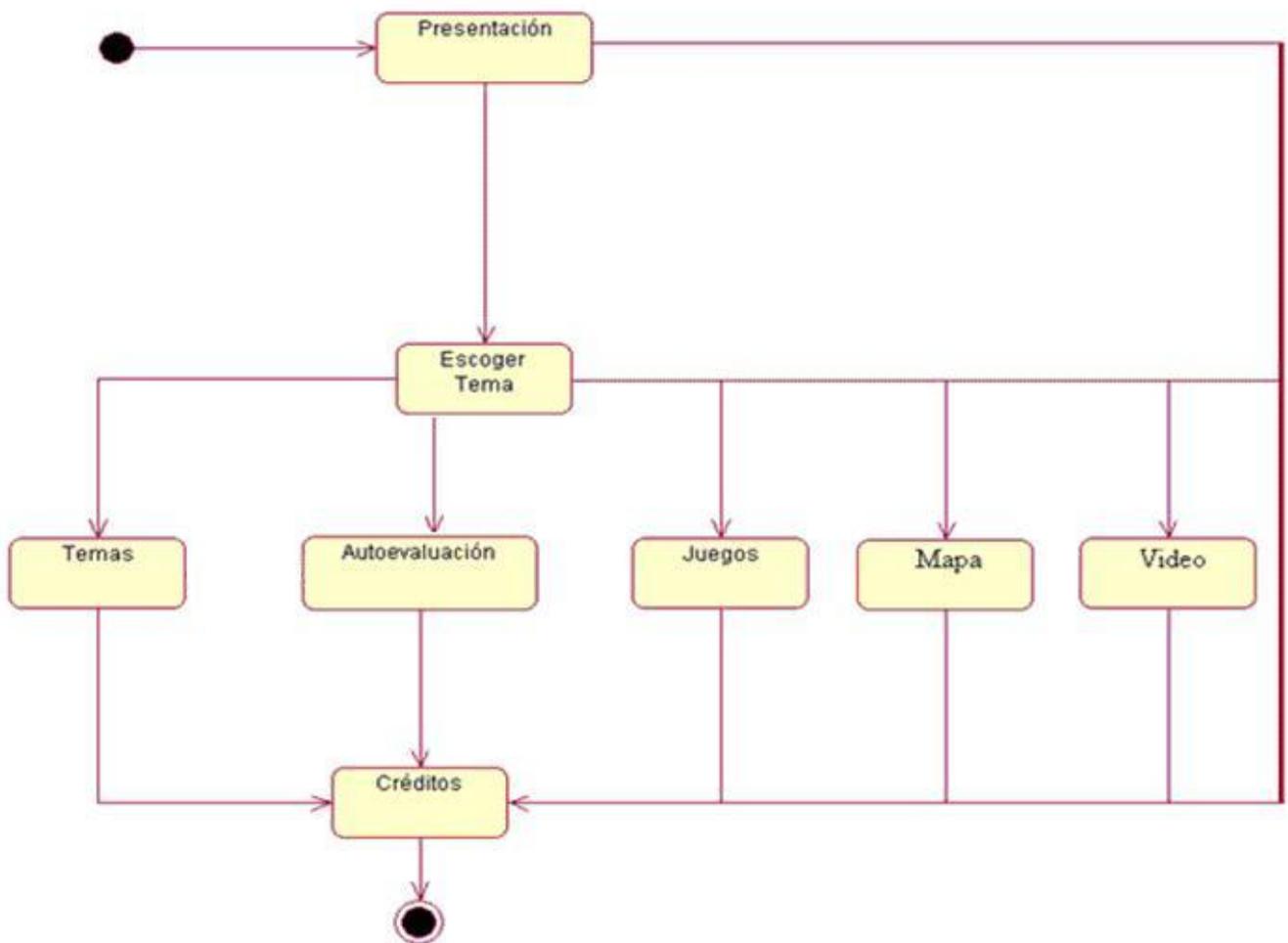


Fig.3 Mapa de Navegación.

DESCRIPCION DE LA SOLUCION PROPUESTA

2.4 Descripción de la funcionalidad.

2.4.1 Requisitos funcionales.

Tab.1 Requisitos Funcionales.

Referencia	Función
R1	Mostrar presentación de la aplicación.
R2	Reproducir música de fondo automática de forma predeterminada.
R3	Permitir al usuario, parar, pausar y reanudar música se fondo.
R4	Mostrar los temas que conforman la aplicación.
R5	Mostrar los contenidos de un tema.
R6	Permitir al usuario seleccionar un tema.
R7	Mostar la información del tema seleccionado.
R8	Permitir al usuario seleccionar los subtemas contenidos en un tema.
R9	Mostar la información del subtema seleccionado.
R10	Ejecutar juego disponible.
R11	Mostrar video.
R12	Controlar volumen y tamaño del video.
R13	Ejecutar ejercicios de autoevaluación
R14	Mostrar resultados obtenidos en la autoevaluación.
R15	Mostrar un mapa conceptual para cada tema del curso, posibilitando mostrar el concepto de un término del mapa.
R16	Permitir salir de la aplicación en cualquier momento deseado.
R17	Mostrar créditos al salir de la aplicación.

2.4.2 Requisitos no funcionales.

2.4.3 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 límite 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.

2.4.4 Requisitos de usabilidad

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

2.4.5 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).

DESCRIPCION DE LA SOLUCION PROPUESTA

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.

2.4.6 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.

2.4.7 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.

2.4.8 Requisitos de soporte

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

2.5 Modelo 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.

2.5.1 Actores del sistema.

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

2.5.2 Diagrama de casos de uso del sistema.

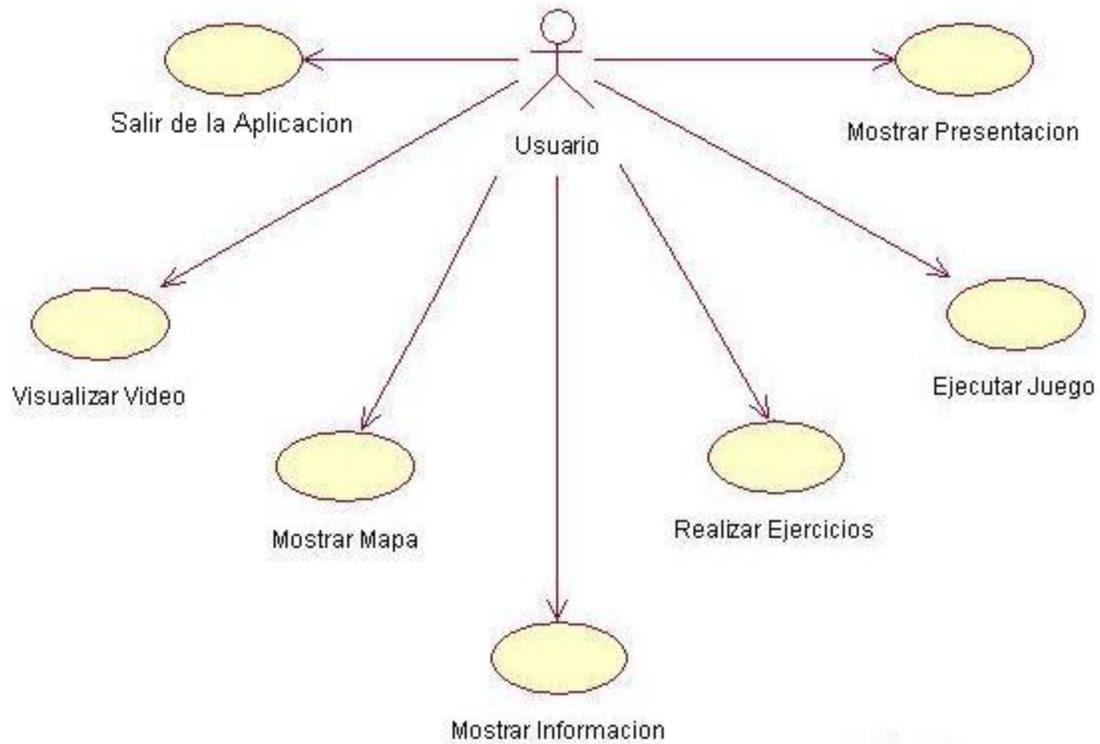


Fig.4 Diagrama de CUS.

DESCRIPCION DE LA SOLUCION PROPUESTA

2.5.3 Breve resumen de los casos de uso del sistema.

CU Mostrar Presentación.

Caso de Uso:	Mostrar Presentación
Actor(es):	Usuario
Propósito:	Mostrar la presentación de la multimedia.
Resumen:	Al iniciar la aplicación, se muestra la presentación del producto y acto seguido el menú principal.
Referencias:	R1

CU Mostrar información.

Caso de Uso:	Mostrar Información
Actor(es):	Usuario
Propósito:	Mostrar la información contenida en la multimedia, texto, glosario, etc.
Resumen:	El usuario de la multimedia puede seleccionar el tipo de información que desea ver. Subtemas dentro de cada tema, autoevaluaciones, palabras calientes, etc. Para visualizar la información deseada deberá hacer clic en los botones que contienen dicha información.
Referencias:	R2, R3, R4, R5, R6, R7, R8, R9.

DESCRIPCION DE LA SOLUCION PROPUESTA

CU Realizar Ejercicios.

Caso de Uso:	Realizar Ejercicios.
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 por ciento obtenido en la misma.
Referencias:	R13, R14.

CU Ejecutar Juegos.

Caso de Uso:	Ejecutar Juego
Actor(es):	Usuario
Propósito:	Ejecutar el juego disponible en la aplicación.
Resumen:	Una vez que el usuario decide que conoce los términos asociados a la disciplina que está estudiando puede dirigirse a la sección de juego. Se ejecuta el juego Sopa de Palabras. Cuando el usuario no desee seguir cerrará el juego e irá a cualquier otra parte de la multimedia que desee.
Referencias:	R10.

DESCRIPCION DE LA SOLUCION PROPUESTA

CU Visualizar Video.

Caso de Uso:	Visualizar Video.
Actor(es):	Usuario
Propósito:	Mostrar video contenido en la multimedia.
Resumen:	El usuario ejecutará el botón "Video" y mostrará el video contenido en la multimedia con las opciones de de parar, reproducir, subir y bajar el volumen del mismo.
Referencias:	R11, R12.

CU Mostrar Mapa.

Caso de Uso:	Mostrar Mapa
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:	R15.

DESCRIPCION DE LA SOLUCION PROPUESTA

CU 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 mostrarán los créditos.
Referencias:	R16, R17.

Conclusiones:

En este capítulo se hizo la descripción de la solución propuesta. Se determinaron los requisitos funcionales, los requisitos no funcionales y los casos de uso que dan solución a estos requisitos, quedando un total de 7 casos de uso para dar solución a este problema. Se construyó el mapa conceptual de la disciplina.

CAPÍTULO 3: SOLUCIÓN DE LA DESCRIPCIÓN PROPUESTA

Introducción

En el presente capítulo se desarrolla la construcción de la solución propuesta, apoyado en los flujos de trabajo de diseño e implementación. Se realizan los diagramas de presentación como parte del modelo de diseño; luego el diagrama de componentes y el de despliegue como elementos fundamentales del modelo de implementación. Se abordarán algunos aspectos de los principios de diseño, así como especificaciones de la estructura de los archivos XML utilizados.

3.1 Principios y normas de diseño.

La interfaz de una aplicación interactiva es su medio de expresión, el recurso por el que informa al usuario, le orienta, instruye, convence o emociona. Para que una interfaz sea eficiente debemos conocer cuál es el objetivo de la aplicación, cómo son las personas a las que se dirige, cómo funciona la mente del usuario, qué dispositivo actuará como soporte, y cuáles serán las condiciones de interacción. También resulta fundamental conocer las herramientas que permitirán comprobar que el diseño responde a los objetivos propuestos.

3.2 Descripción de archivos XML

La aplicación utiliza el lenguaje XML para almacenar todos los textos de la multimedia.

Descripción de archivos XML Tema

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

```
<tema>

<nombreTema><p align='center'><i><b>Nombre del tema</b></i>

</p></nombreTema>

<Subtema1><b><i>Nombre del Subtema1</i></b> </ Subtema1>

.

< Subtema_n><b><i>Nombre del epígrafe n</i></b> </ Subtema_n>

<texto_Subtema 1> Texto que pertenece al epígrafe 1 del tema</texto_Subtema 1>

.

.

<texto_Subtema _n> Texto que pertenece al epígrafe 1 del tema</texto_Subtema _n>

</tema>
```

Descripción de archivos XML Autoevaluación

```
<quiz>

<title> Nombre del título de la evaluación </title>

<items>

<item>

<question>Aquí estará la pregunta que se realizará en la autoevaluación que será al estilo Verdadero o Falso </question>

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

<answer>Verdadero</answer>

</item>

<item>

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

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

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

</item>
```

SOLUCION DE LA DESCRIPCION PROPUESTA

```
</items>
```

```
</quiz>
```

El XML está estructurado de la siguiente manera:

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

`</quiz>` cierre de nodo.

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

`</answer>` cierre de nodo.

3.3 Diagramas de Presentación del modelo de Análisis.

Estos diagramas tienen 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).

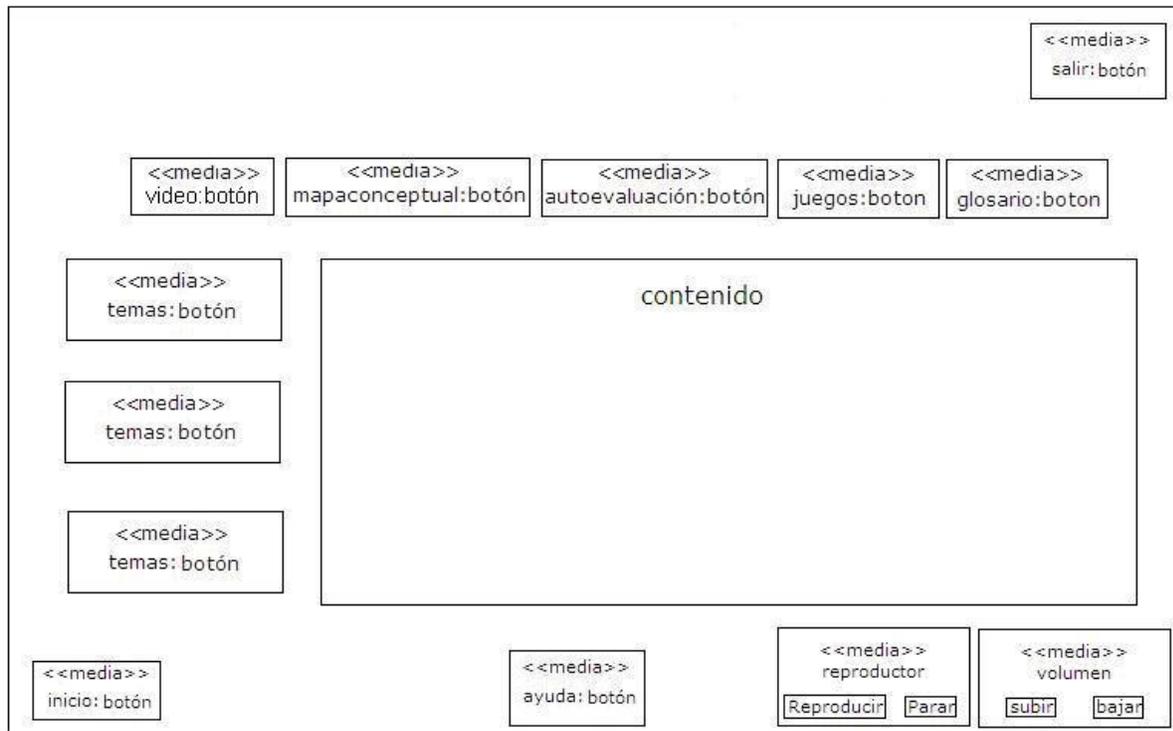


Fig.5 Diagrama de presentación general.

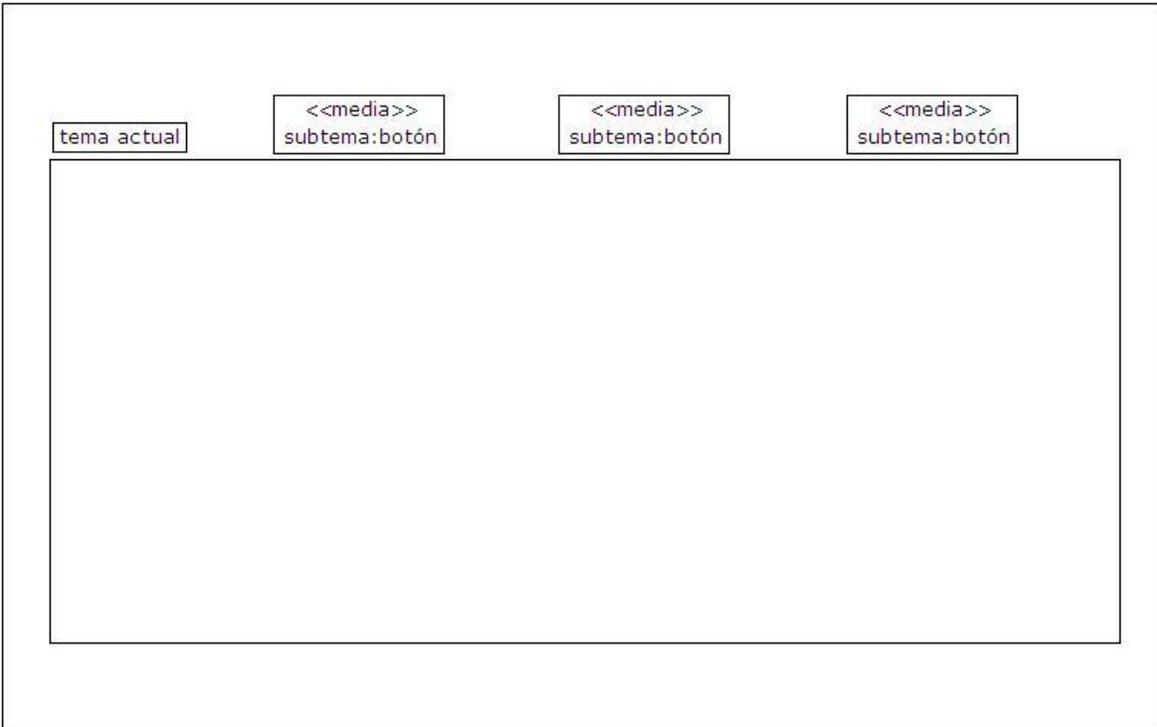


Fig.6 Diagrama de presentación de los temas.

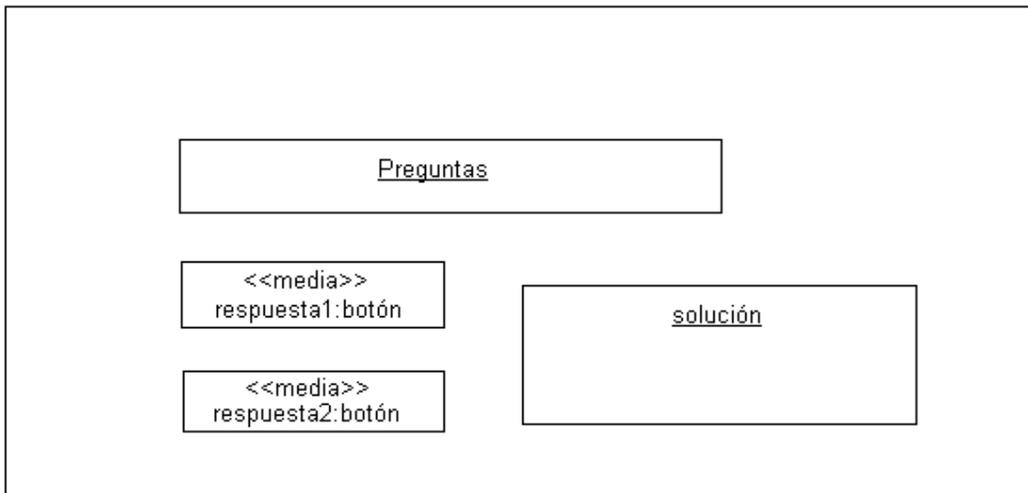


Fig.7 Diagrama de presentación de Autoevaluación.

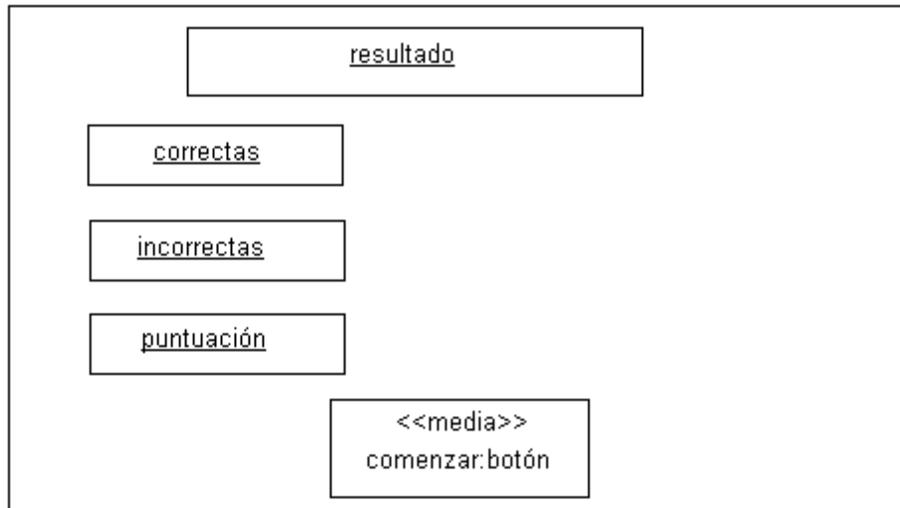


Fig.8 Diagrama de presentación de Resultado de Evaluación.

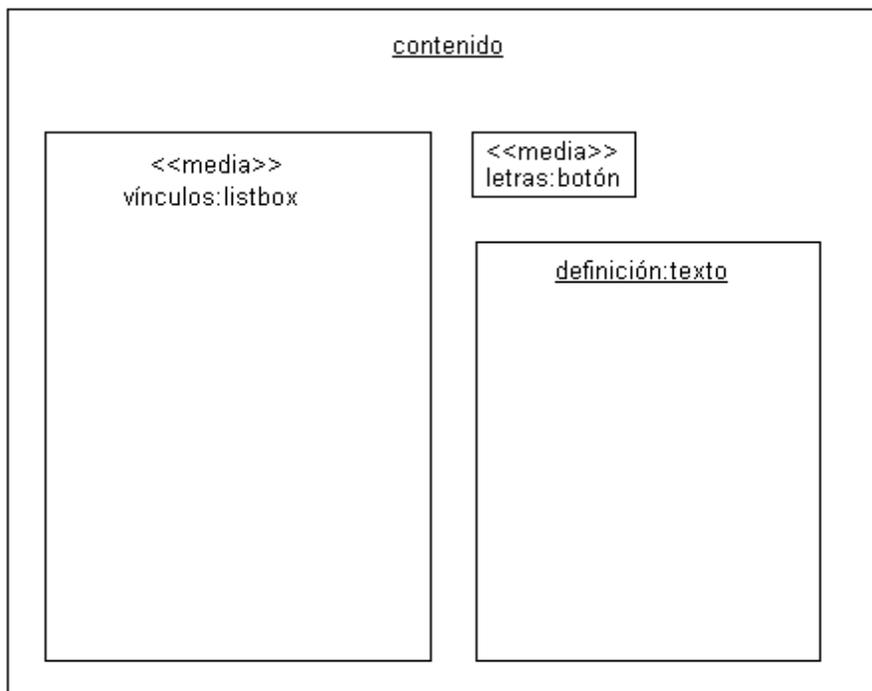


Fig.9 Diagrama de Presentación del Glosario de términos.

3.4 Modelo de Despliegue

El modelo de despliegue describe la distribución física del sistema en nodos de información; mostrando cómo están distribuidos los componentes de software entre los distintos nodos. Incorpora los elementos establecidos en la arquitectura para completar la descripción física de la aplicación; permite comprender la correspondencia entre la arquitectura software y la arquitectura hardware (Figura 10).



Fig.10 Modelo de Despliegue.

3.6 Diagrama de Componentes

Un diagrama de componentes representa la separación de un sistema de software en componentes físicos y muestra las dependencias entre estos componentes (Figura 11).

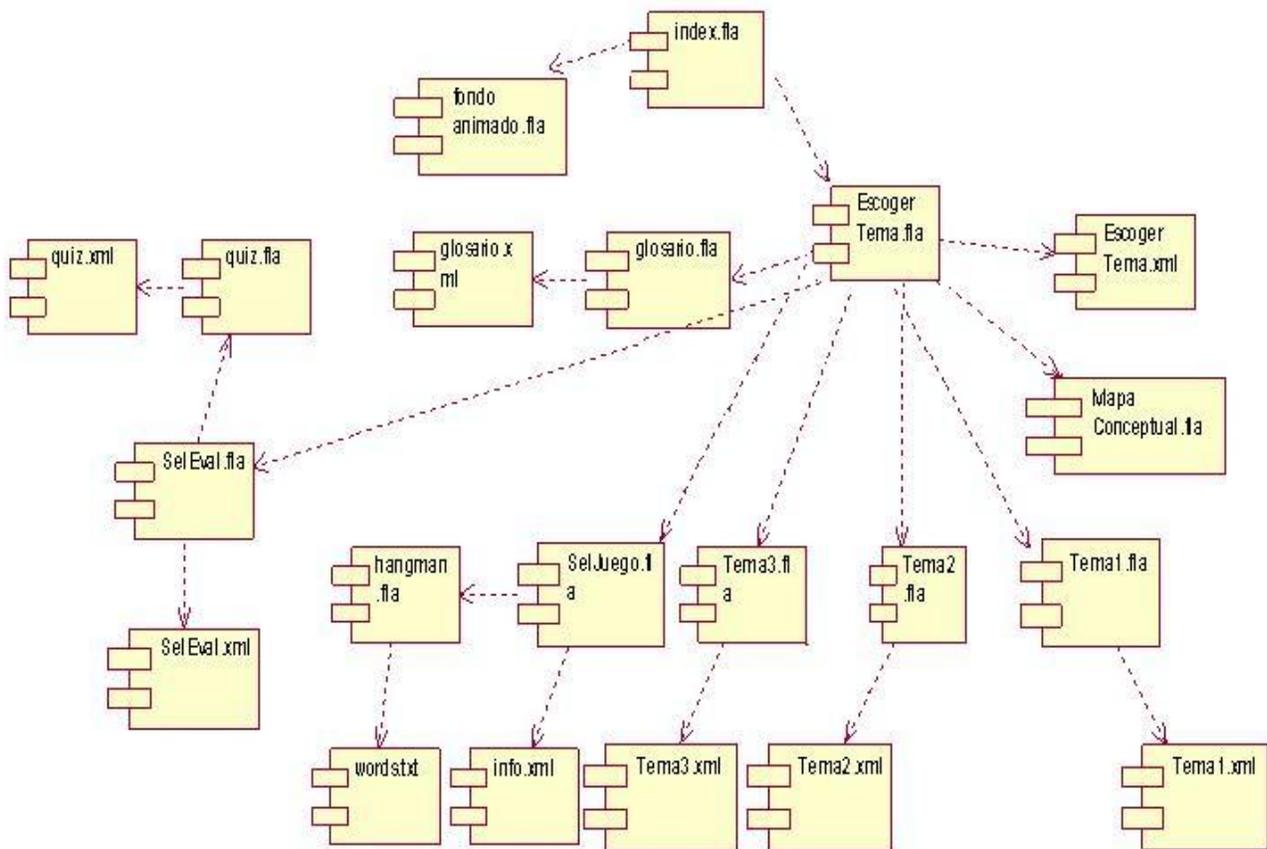


Fig.11 Diagrama de Componentes.

Conclusiones

En este capítulo se hizo un análisis de los principios del diseño que se utilizó en la multimedia, así como las estructuras XML usadas en la misma. Además se dio una breve panorámica de la multimedia a través de las pantallas de presentación, así como los artefactos de la implementación.

CONCLUSIONES GENERALES

Una vez concluido este trabajo se puede afirmar:

- Se desarrolló con éxito la multimedia de Auditorías y Revisiones de software.
- Se confeccionaron todos los temas concebidos para el curso auditorías y revisiones de software y, como valor agregado, se construyó el mapa conceptual que reúne los principales conceptos asociados al tema de auditorías y revisiones de software.
- La multimedia tiene una interfaz amigable y fácil de usar, 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 realización de esta aplicación tiene un enfoque educacional, ya que expresa el aumento del conocimiento sobre el tema de auditorías y revisiones, apoyándose en mapas conceptuales que agrupan los principales conceptos de la disciplina.
- Todas las tareas a desarrollar fueron cumplidas con éxito, especialmente la de diseño e implementación de la multimedia; por lo que se alcanzó el objetivo propuesto.

RECOMENDACIONES

RECOMENDACIONES

Se recomienda:

- Para futuras versiones se recomienda la integración de la multimedia a un conjunto que agrupe las multimedias de las demás disciplinas.

- Ampliar el contenido de los temas en la multimedia.

REFERENCIAS BIBLIOGRAFICAS

Referencias Bibliográficas:

By multimediam - Posted .2006. *¿Qué es multimedia interactiva?*

<http://portal.educar.org/multimediam/blog/queesmultimediainteractiva>

Carlos del Angel Castellanos. 2007

<http://moodletecnoinf.blogspot.com/2007/10/que-es-el-moodle.html>

Ciberaula. 2006. *Lo nuevo en Flash 8*

http://flash.ciberaula.com/articulo/flash_8/

Enrique Hernández Orallo. El Lenguaje Unificado de Modelado (UML)

<http://www.disca.upv.es/enheror/pdf/ActaUML.PDF>

Febles.2005. *La gestión y configuración y el desarrollo de software en las universidades una experiencia practica.2005, Revista Cubana de Educación superior, No1/2005.*

http://www.informaticahabana.com/evento_virtual/files/MUL069.pdf.

Gerardo González García. 2007. *Tema: Reflexiones.*

<http://profesorinteractivo.blogia.com/2007/041701--que-es-software-educativo-.phpmultimedia>

Informática y multimedia educativo. 2008. *Diseño y desarrollo de materiales.*

<http://alcoyanita.wordpress.com/2008/05/14/informatica-y-multimedia-educativo-diseno-y-desarrollo-de-materiales/>

Ing.Elda Quiroga. 2004. *Introducción a la Ingeniería (ISI), Investigación.*

REFERENCIAS BIBLIOGRAFICAS

<http://homepages.mty.itesm.mx/al598180/introduccion/investigacion.doc>.

Ing. Yaimí Trujillo. 2006. *Evaluación teórica de la adopción del enfoque de Factorías de Software en la Universidad de Ciencias Informáticas.*

<http://www.intempres.pco.cu/Intempres2006/Intempres2006/Evaluacion%20de%20trabajos/Yaim%ED%20Trujillo%20Casa%F1ola%20P.pdf>

Jesús Salinas Ibáñez. *Hipertexto e hipermedia en la enseñanza universitaria*

<http://tecnologiaedu.us.es/bibliovir/pdf/gte5.pdf>

Luciamultimediasjose. 2009. *¿Que es la Multimedia?*

<http://luciamultimediasjose.blogspot.es/>

Toolbook. Guides and tutoriales

[http://www.toolbook.com/learn_guides.php?from=menuDemo:](http://www.toolbook.com/learn_guides.php?from=menuDemo)

<http://www.toolbook.com/overviewdemo.html>

SERUNAM. 2007. *Tutorial de uso de la plataforma Moodle para el Diplomado Integración de medios en formatos CD, DVD, Web” en su primera emisión.*

<http://www.ser.unam.mx/oferta/diplomados/integracionmedios/>

Standish Group. 2007. *There’s Less Development Chaos Today*

<http://www.standishgroup.com/>

WebCT 4.1 Manual. 2004. *Manual_8_Nov04*

http://csie.unavarra.es/manuales/WebCT4.1_Manual_8_Nov04.pdf

Wikipedia. 2009. *MAcromedia Authorware*

http://es.wikipedia.org/wiki/Macromedia_Authorware

REFERENCIAS BIBLIOGRAFICAS

Wikipedia. 2009. *Software_educativo.*

http://es.wikipedia.org/wiki/Software_educativo 20 feb 2009

Yancy Martínez Pérez, Alexey Díaz Domínguez, Abel Ernesto Lorente Rodríguez. 2006.
Plantilla para el montaje dinámico de los productos de la colección multisaber. Universidad de las Ciencias Informáticas, Dirección de Software Educativo.

http://www.informaticahabana.com/evento_virtual/files/MUL069.pdf.

REFERENCIAS BIBLIOGRAFICAS

Bibliografía

IEEE, Std. 610-1990. *Standard Glossary of Software Engineering Terminology*

IEEE Std 1028 - 1997 IEEE. *Standard for Software Reviews.*

JACOBSON, Booch Grady y Rumbaugh, James. 2000. *El proceso unificado de desarrollo de software (RUP).* s.l. : Pearson Educación S.A, 2000.

NORMA-ISO 9000_2000 ISO 9000_2000

NORMA-19011

NORMA -ISO 9001_2000 ISO 19011_2002

NORMA -ISO-9001-2000_Requisitos

NORMA -ISO 9001_2000

NORMA -ISO 19011_2002

PRESSMAN, Roger. 2001. *Ingeniería de Software. Un enfoque Práctico.* 2001

SWEBOK. 2004. *Guide to the Software Engineering Body of Knowledge .* 2004.

Anexos

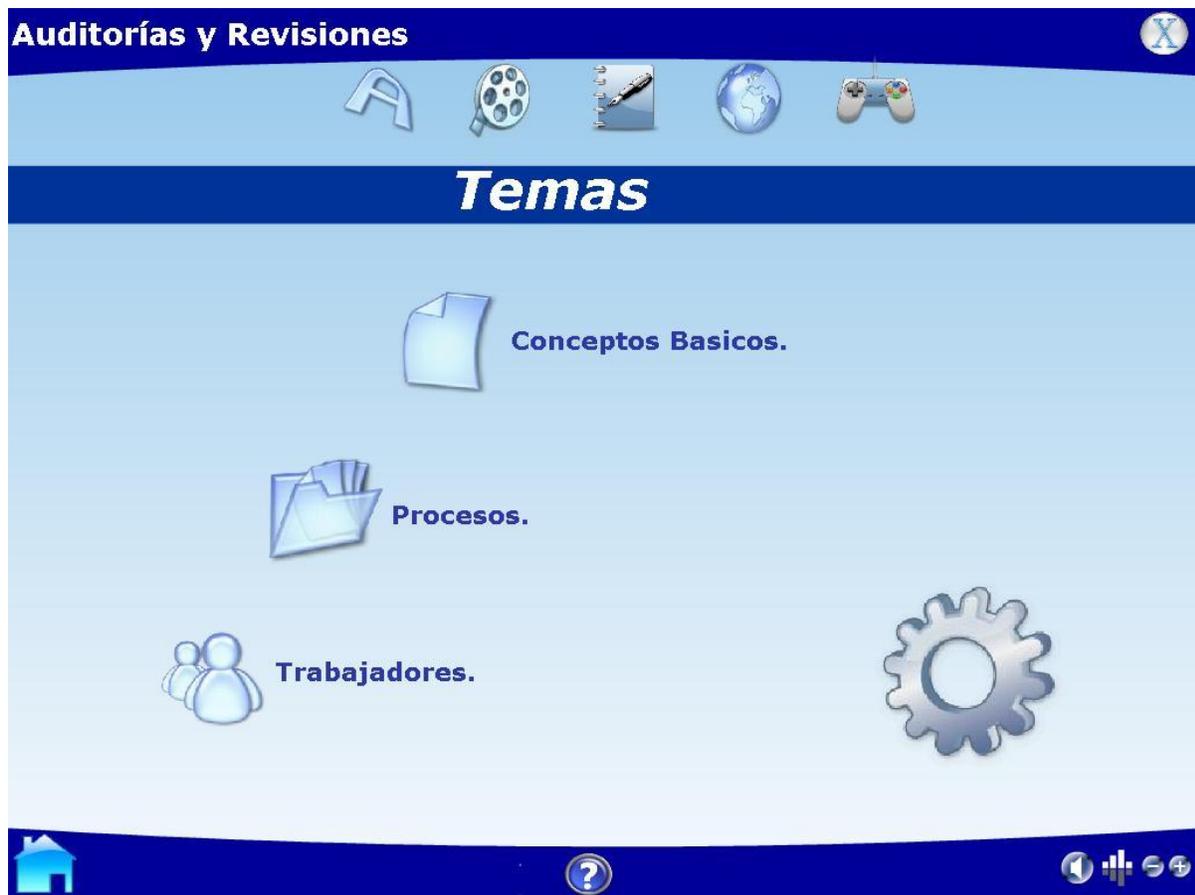


Fig.12 Pantalla Escoger Temas de la Multimedia

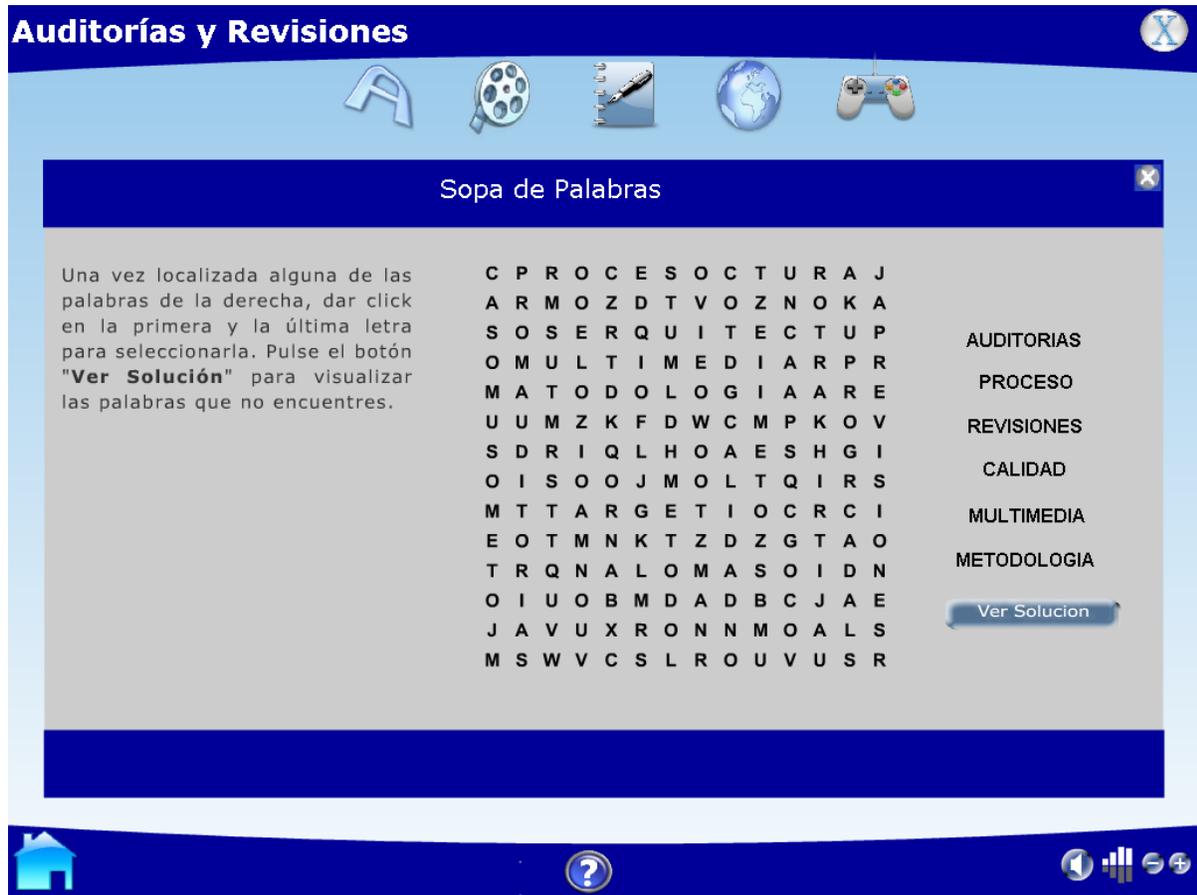


Fig.13 Pantalla de Juegos.

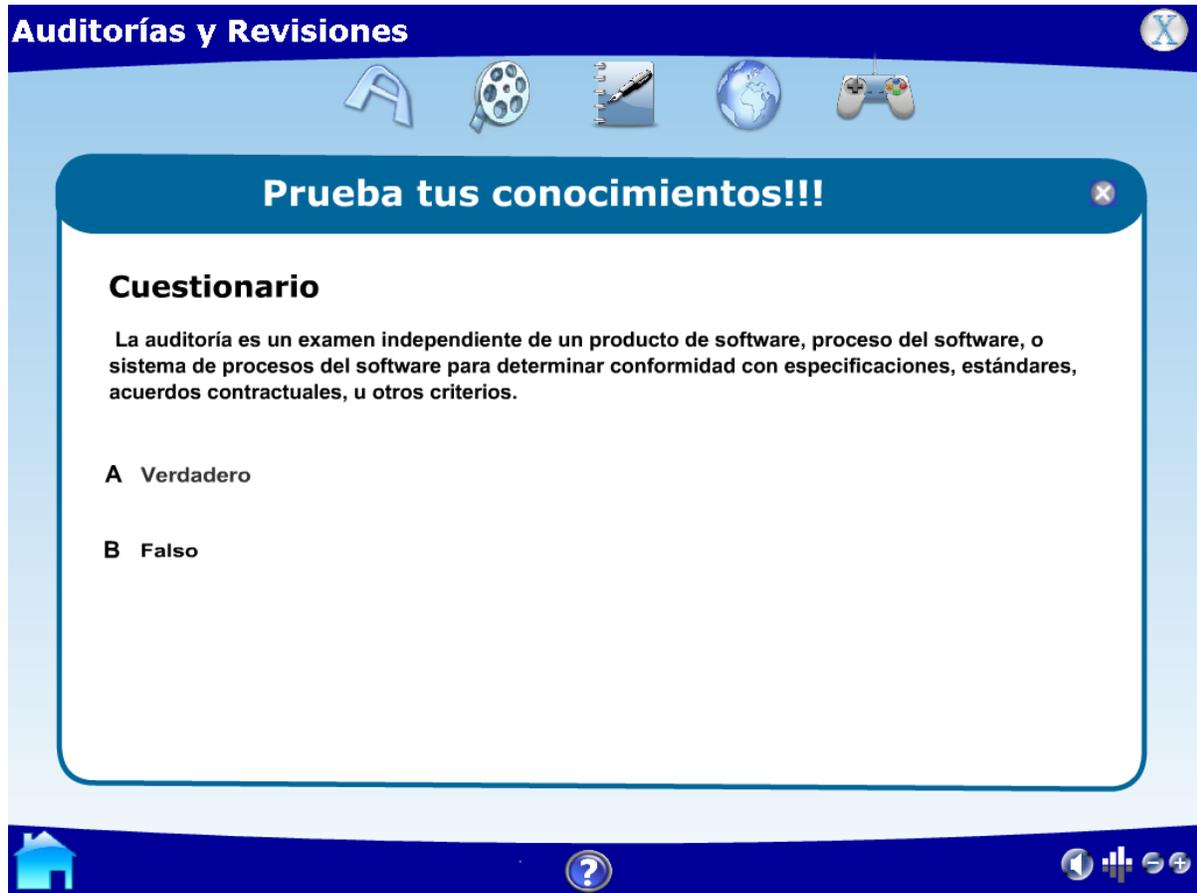


Fig.14 Pantalla de Autoevaluación de los Temas

GLOSARIO DE TÉRMINOS

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

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

RUP: el Proceso Unificado Racional o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).

UML: es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. El UML ofrece un estándar para escribir un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables, es un "lenguaje" para especificar y no un método o un proceso. El UML se usa para definir un sistema de software; para detallar los artefactos en el sistema, para documentar y construir. El UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no especifica en sí mismo qué metodología o proceso usar.

OMMMA-L: el Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia es una extensión de UML especializado en aplicaciones multimedia. (ver explicación ampliada en el CAPÍTULO 1: Lenguajes de Modelado).

GLOSARIO DE TERMINOS

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

XML: es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones.

.Fla: Una extensión de archivo con la cual se puede trabajar en Macromedia Flash, es decir, es el código fuente de una animación.

Herramientas o software de autor: también llamadas “lenguajes de autor” permiten a los profesores construir programas del tipo tutoriales, especialmente a profesores que no disponen de grandes conocimientos de programación e informática, ya que usando muy pocas instrucciones, se pueden crear muy buenas aplicaciones hipermediales.