

Universidad de las Ciencias Informáticas

Facultad # 5



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

**Título: Multimedia Interactiva para el aprendizaje
de gestión de la configuración de software.**

Autora: Yurelis Pérez Reinoso.

Tutores: MSc. Yamilis Fernández Pérez.

Ing. Miguel Ángel del Pino Zincke

Junio, 2009

Tutor: MSc. Yamilis Fernández Pérez

Correo electrónico: yamilisf@uci.cu

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

Tutor: Ing. Miguel Ángel del Pino Zincke

Correo Electrónico: madelpino@uci.cu

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

Quiero agradecer antes que nada a Nuestro Señor por darme la fuerza suficiente y la fe necesaria para seguir adelante.

A mi familia y en especial a mis padres por ayudarme, por ser pacientes, y apoyarme siempre que lo necesité. A mi mamá por tanto amor y dedicación durante tantos años. A mi tía querida por aconsejarme y reanimarme cada vez que flaqueaba en momentos difíciles y creía no poder. A Dailyn por todo el apoyo que siempre me ha brindado. A mi prima Irelis, a todos mis tíos que me han ayudado mucho. A mis abuelos que aunque no estén conmigo en estos momentos, siempre fueron muy especiales para mí, me apoyaron y ayudaron en todo momento.

A Lorenzo, por toda su ayuda, por su apoyo incondicional y comprensión en aquellos momentos en que se hacía difícil estar junto a mí. Sin su ayuda este trabajo no hubiese sido posible, a ti te agradezco gran parte de lo que hoy soy.

A todos mis amigos, en especial a Yunier por todo su inmenso cariño y estar ahí siempre conmigo. A Carlos Mario, Jorgito, a mis compañeros de grupo, a las muchachitas de mi apto, a todas las personas que siempre estuvieron dispuestos a ayudarme.

A mis tutores, por su ayuda y talento profesional.

Y para terminar quiero agradecerles a todos los que no mencioné pero que de cualquier forma siempre estuvieron conmigo.

A todos. Muchas Gracias.

A mis padres por ser las personas que más amo en la vida. A mi abuela por todo su infinito amor. A mi tía por ser como una madre para mí. A toda mi familia.

El actual momento histórico que vive la humanidad, con la consecuente explosión de la denominada “era digital”, ha propiciado la incorporación de los entornos virtuales y multimediales al proceso de enseñanza aprendizaje, perfeccionándolo y apoyándolo a un mismo tiempo. El software educativo toma cada vez mayor importancia como herramienta para la transmisión del conocimiento, y constituye una vía alternativa al método presencial, muy efectiva para el aprendizaje, debido a sus ventajosas y atractivas prestaciones. La Universidad de las Ciencias Informáticas (UCI) constituye una verdadera potencia en este sentido dentro del terreno nacional, debido al alto número de software educativo que produce para el apoyo al proceso docente. No obstante, existen aún materias que no cuentan con este soporte, las que además resultan imprescindibles para el desarrollo de software con calidad, como es el caso de la Gestión de Configuración de Software. El presente trabajo de diploma tiene como objetivo la producción de una multimedia, que sirva de apoyo a la mencionada materia. Para ello se realizó un estudio de las principales tendencias actuales en la tecnología de aplicaciones de este tipo, finalizando con el desarrollo de un producto. Se utilizaron como herramientas informáticas Adobe Flash CS3 Profesional y se trabajó a través de la metodología de desarrollo de software RUP, se utilizó como lenguaje de modelado OMMMA-L, el cual es una extensión del UML para el diseño de software multimedia. Además se utilizó el lenguaje de programación ActionScript y XML para presentar de forma dinámica la información en la multimedia. Este trabajo contiene la fundamentación teórica que sustenta el producto, además de todo el análisis, diseño e implementación del mismo.

Palabras Clave:

Gestión de Configuración de Software, multimedia, producto multimedia, aprendizaje, calidad

Introducción	1
Capítulo 1: "Fundamentación Teórica"	6
1.1 SWEBOK y SEEK	6
1.2 Gestión de Configuración de Software	8
1.2.1 GESTIÓN DE LOS PROCESOS DE GCS	9
1.2.1.1 PLANIFICACIÓN DEL PROCESO DE GCS	10
1.2.2 IDENTIFICACIÓN DE LA CONFIGURACIÓN DE SOFTWARE	12
1.2.3 CONTROL DE LA CONFIGURACIÓN DEL SOFTWARE	14
1.2.4 CONTABILIDAD DEL ESTADO DE LA CONFIGURACIÓN DEL SOFTWARE	15
1.2.5 AUDITORÍA DE LA CONFIGURACIÓN DEL SOFTWARE	16
1.2.6 GESTIÓN DEL RELEASE Y ENTREGA DEL SOFTWARE	17
1.2.7 LAS HERRAMIENTAS DE GESTIÓN DE CONFIGURACIÓN DE SOFTWARE	18
1.3 Enseñanza de Gestión de Configuración de Software en la Actualidad	19
1.4 Multimedia, Características y Conceptos relacionados	20
1.5 Tendencias Tecnológicas Actuales	22
1.5.1 HERRAMIENTAS PARA LA CREACIÓN DE SOFTWARE MULTIMEDIA	22
1.5.2 LENGUAJES UTILIZADOS EN LA MULTIMEDIA	28
1.5.3 METODOLOGÍAS A CONSIDERAR PARA EL DESARROLLO DE SOFTWARE	30
1.5.4 LENGUAJES DE MODELADO	31
1.5.5 FUNDAMENTACIÓN DE LA HERRAMIENTA Y METODOLOGÍA A UTILIZAR	33
Capítulo 2: "Descripción de la Solución Propuesta"	35
2.1 Identificación de la audiencia	35
2.2.2 MAPA CONCEPTUAL	37
2.3 Modelo de Dominio	38
2.3.1 DIAGRAMA DE CLASES DEL MODELO DE DOMINIO	39
2.3.2 ANÁLISIS DE LOS CONCEPTOS DE DOMINIO	39
2.4 Descripción de la Funcionalidad	40
2.4.1 REQUISITOS FUNCIONALES	41
2.4.2 REQUISITOS NO FUNCIONALES	42
2.5 Modelo de Casos de Uso del Sistema	44
2.5.1 DIAGRAMA DE CASOS DE USO DEL SISTEMA	44
2.5.2 DETERMINACIÓN Y JUSTIFICACIÓN DE LOS ACTORES DEL SISTEMA	45

2.5.3 DESCRIPCIÓN DE CASOS DE USO	46
2.6 Mapa de Navegación	50
Capítulo 3: “Construcción de la solución propuesta”	52
3.1 Principios de Diseño	52
3.2 Diagramas de Presentación	53
3.3 Descripción de archivos XML	57
3.4 Diagrama de Componentes	60
Conclusiones	62
Recomendaciones	63
Referencias Bibliográficas	64
Bibliografía	66
Anexos	67
Glosario de Términos	70

ÍNDICE DE FIGURAS

FIG. 1 GUIÓN	37
FIG. 2 MAPA CONCEPTUAL	38
FIG. 3 DIAGRAMA DE CLASES DEL MODELO DEL DOMINIO	39
FIG. 4 DIAGRAMA DE CASOS DE USO DEL SISTEMA.	45
FIG. 5 MAPA DE NAVEGACIÓN GENERAL	51
FIG. 6 DIAGRAMA DE PRESENTACIÓN PANTALLA PRINCIPAL	54
FIG. 7 DIAGRAMA DE PRESENTACIÓN CONTENIDOS	55
FIG. 8 DIAGRAMA DE PRESENTACIÓN VIDEO	56
FIG. 9 DIAGRAMA DE PRESENTACIÓN SALIR	57
FIG. 10 DIAGRAMA DE COMPONENTES	61
FIG. 11 PANTALLA PRINCIPAL	67
FIG. 12 PANTALLA CONTENIDOS	68
FIG. 13 PANTALLA AYUDA	69
FIG. 14 PANTALLA GLOSARIO	70

<i>FIG. 15 PANTALLA MAPA CONCEPTUAL.....</i>	<i>71</i>
<i>FIG. 16 PANTALLA JUEGO.....</i>	<i>72</i>
<i>FIG. 17 PANTALLA VIDEO.....</i>	<i>73</i>
<i>FIG. 18 PANTALLA SALIR.....</i>	<i>74</i>

ÍNDICE DE TABLAS

<i>TABLA 1. ÁREA DE CONOCIMIENTO GCS SEGÚN SWEBOK.....</i>	<i>7</i>
<i>TABLA 2. ÁREA DE CONOCIMIENTO GCS SEGÚN SEEK.....</i>	<i>8</i>
<i>TABLA 3. COMPARACIÓN ENTRE HERRAMIENTAS PARA DESARROLLAR SOFTWARE MULTIMEDIA.....</i>	<i>26</i>
<i>TABLA 4. REQUISITOS FUNCIONALES.....</i>	<i>41</i>
<i>TABLA 5. JUSTIFICACIÓN DE LOS ACTORES DEL SISTEMA.....</i>	<i>45</i>
<i>TABLA 6. MOSTRAR PRESENTACIÓN.....</i>	<i>46</i>
<i>TABLA 7. MOSTRAR INFORMACIÓN.....</i>	<i>46</i>
<i>TABLA 8. REALIZAR EJERCICIO.....</i>	<i>47</i>
<i>TABLA 9. MOSTRAR MAPA.....</i>	<i>47</i>
<i>TABLA 10. CONTROLAR SONIDO.....</i>	<i>48</i>
<i>TABLA 11. EJECUTAR JUEGO.....</i>	<i>48</i>
<i>TABLA 12. VISUALIZAR VIDEO.....</i>	<i>49</i>
<i>TABLA 13. MOSTRAR GLOSARIO.....</i>	<i>49</i>
<i>TABLA 14. MOSTRAR AYUDA.....</i>	<i>50</i>
<i>TABLA 15. SALIR APLICACIÓN.....</i>	<i>50</i>

INTRODUCCIÓN

La industria de software en el mundo se está desarrollando a gran velocidad y cuenta con la introducción permanente de importantes innovaciones. De esta manera, se vislumbra un mercado internacional de grandes proporciones para las próximas décadas y en los diferentes campos que esta industria ofrece. A pesar de ello, dentro de todo el crecimiento vertiginoso de esta industria, se pueden ver con frecuencia fracasos de software a nivel mundial. Se aprecia que aún esta industria está afectada por la mala calidad de algunos aspectos que determinan la competitividad de una empresa.

Es un hecho que la mayoría de los software que se producen en el mundo chocan, durante el tiempo que dura su realización, con una serie de problemas que afectan de manera significativa el resultado final. Cuba no escapa a esta realidad. Los programas se encarecen, alteran, detienen, etc., por lo que resulta en extremo complejo el período de desarrollo de estos proyectos.

Según estudios realizados por “The Chaos Report” [Standish Group], más del 70% de los proyectos de Software “fracasan”. Se observa como el 29% de los proyectos se completan en tiempo, coste y con la funcionalidad requerida. El 53% tienen problemas de plazos, presupuestos o bien no entregan toda la funcionalidad requerida, y el 18% se cancelan. En media, un proyecto de Software se retrasará de 6 a 12 meses y costará entre un 50% y un 100% más. (Gaitero, 2008)

De forma general puede afirmarse que esto ocurre debido a una mala planificación en el plan de desarrollo de software. Los protocolos a seguir para el desarrollo de los proyectos apenas se cumplen y la mayoría de las veces los mismos son producto de un proceso desordenado e irregular. Esto sucede fundamentalmente por la falta de preparación de los profesionales informáticos en materia de calidad de software.

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. Ello 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, 2005a]

La calidad de software adquiere cada día mayor importancia en el mercado mundial, debido a las exigencias de los clientes y la competitividad entre las empresas. La garantía de la calidad es un aspecto fundamental a tener en cuenta para cualquier institución productora de software. Cuba se encuentra en un momento importante de desarrollo de su infraestructura tecnológica, dentro de lo que sin dudas destaca la industria del software, con un papel fundamental en dicho desarrollo.

“La Industria Cubana del Software (InCuSoft) está llamada a convertirse en una significativa fuente de ingresos para el país, como resultado del correcto aprovechamiento de las ventajas del alto capital humano disponible. 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. El continuar la producción sostenida de software de alta calidad en prestaciones, imagen y soporte, para satisfacer las necesidades nacionales en estos sectores, tendrá una positiva repercusión en el incremento de la exportación” [MINREX, 2004].

Es precisamente por ello que el país posee cada vez mayor interés en insertarse dentro del mercado mundial de software. No obstante, aún debemos superar algunos obstáculos en el desarrollo de los mismos con tal de alcanzar dicho objetivo. Así lo demuestran encuestas relacionadas con la calidad en los procesos de desarrollo y el producto final, realizadas a 31 empresas de la Industria Cubana de Software donde se refleja como muchos de los problemas están dados por la no aplicación de actividades de Gestión de Configuración, disciplina que se ha considerado, en los últimos años, como la columna vertebral en el desarrollo de software. En estas encuestas se destaca que los desarrolladores y los líderes, en un 61%, no conocen que es GCS y en un 79% no se aplican ninguno de los procedimientos asociados a este proceso. Esta misma encuesta fue aplicada a estudiantes de tercer año que forman parte de equipos de desarrollo de software en la CUJAE y los resultados fueron similares. [Febles 2005a]

La Universidad de las Ciencias Informáticas es una universidad de nuevo tipo que une formación y producción. En ella hay un incremento de la actividad productiva y se hace imprescindible enseñar y comprometer a los estudiantes con la calidad del software. Con este fin se ha incluido dentro del plan de estudio de la UCI la disciplina de Ingeniería y Gestión de Software y se ha desarrollado además el diseño de un segundo perfil de calidad de software. A pesar de todo esto, la Universidad no queda exenta de los

problemas antes mencionados sobre la no aplicación de técnicas de Gestión de Configuración de Software.

Dentro de la disciplina de Ingeniería de Software se encuentra presente la enseñanza de Gestión de Configuración como un tema incluido en ésta, donde se definen los conceptos fundamentales relacionados con la Gestión de Configuración, se identifican los principales elementos de Gestión de Configuración de un sistema, y determinan los rasgos esenciales de las herramientas utilizadas para la Gestión de Configuración. Además este tema constituye una de las materias dadas en el curso de Calidad de Software incluido dentro de uno de los cursos del Perfil. A pesar de ello existen problemas con el desarrollo del proceso de Gestión de Configuración de Software en los proyectos de la universidad.

A partir de las entrevistas realizadas por consultores del SIE Center de Monterrey, México, en la UCI, en noviembre del 2008, se detectaron los siguientes problemas relacionados con la disciplina de Gestión de la Configuración en los diferentes proyectos productivos de la Universidad:

- ✓ No se realiza un entrenamiento del personal para el proceso de Gestión de Configuración.
- ✓ No se realizan las evaluaciones que determinen si se están efectuando como corresponde el establecimiento de líneas base, los procedimientos, si se supervisan y controlan los cambios, entre otras actividades.
- ✓ El establecimiento de las líneas base se realiza parcialmente.
- ✓ El control de cambio se realiza de forma parcial.
- ✓ No se revisan las actividades, estado y resultado del proceso de gestión de configuración con la alta gerencia y no se resuelven problemas.
- ✓ Las auditorías de configuración no siempre se realizan, es decir hay problemas con la documentación de las líneas base, no siempre son precisas.
- ✓ La identificación de los ítems de configuración se realiza parcialmente.

Analizando la situación antes detallada, se hace necesario elevar la capacitación de las personas involucradas en procesos de desarrollo de software, por lo que identificamos el siguiente **problema científico** a resolver:

¿Cómo sustentar la inclusión de buenas prácticas durante el proceso de enseñanza-aprendizaje de Gestión de Configuración de Software, a través de los medios interactivos?

El **Objeto de Estudio** de este trabajo es:

Los medios interactivos en el proceso de enseñanza-aprendizaje de Gestión de la Configuración de Software en la UCI.

De ello se deriva el **campo de acción**: El proceso de desarrollo de software multimedia interactivo que apoye la enseñanza-aprendizaje de Gestión de la Configuración de Software.

El **objetivo general** es:

Implementar una multimedia interactiva como medio de sustento de la inclusión de buenas prácticas durante el proceso de enseñanza aprendizaje de Gestión de Configuración de Software en la UCI.

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

- ✓ Recopilar y analizar toda la información necesaria sobre gestión de la configuración del software para la elaboración del contenido del producto a desarrollar.
- ✓ Confeccionar un mapa conceptual que contenga los conceptos necesarios para la comprensión del tema propuesto.
- ✓ Analizar críticamente 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).
- ✓ Analizar las metodologías existentes para el desarrollo de aplicaciones con tecnología multimedia.
- ✓ Seleccionar la metodología de desarrollo de software que mejor se ajuste a las características del producto a desarrollar.

- ✓ Analizar y diseñar una multimedia que contenga un mapa conceptual que ayude a la comprensión del tema, utilizando la herramienta y el lenguaje de modelado seleccionado.

Para lograr el objetivo de esta investigación se definen los siguientes métodos teóricos a utilizar. Analítico-sintético, para la extracción de los elementos más importantes de la información recopilada durante la investigación. En general para el procesamiento de la información que será necesaria para el conocimiento de los diferentes temas en que se fundamenta este trabajo, y como parte muy importante de éste, en específico para la información referida al estudio de la Gestión de Configuración de Software para entender mejor cada uno de los conceptos que se refieren a este tema. Inductivo-Deductivo, para realizar un análisis de los elementos y llegar a conclusiones. El método empírico que se utilizará es la entrevista a diferentes profesores y especialistas en el tema.

El presente trabajo está conformado por tres capítulos que abarcan todo el contenido de la investigación:

Capítulo 1: “Fundamentación teórica.”

Resume todos los conceptos relacionados con el objetivo de la investigación para lograr comprender bien el problema en cuestión. Se describen además los lenguajes de programación, las metodologías y sistemas para realizar el análisis y diseño de la aplicación, así como las tendencias y tecnologías actuales de este tipo de aplicación.

Capítulo 2: “Descripción de la solución propuesta”

En este capítulo se realiza un análisis de las principales funcionalidades que debe tener el sistema a desarrollar, definiendo el dominio de la aplicación, mapa de navegación y el levantamiento de los requerimientos funcionales y no funcionales. También se realiza una descripción de los casos de uso correspondientes, modelándose el sistema.

Capítulo 3: “Construcción de la solución propuesta.”

En este capítulo se desarrolla la construcción del sistema, incluyéndose los diagramas de presentación. También se realiza el diagrama de componentes como parte del modelo de implementación.

CAPÍTULO 1: "FUNDAMENTACIÓN TEÓRICA"

INTRODUCCIÓN

Durante el desarrollo de este capítulo se expondrán los principales conceptos que nos ayudaran a comprender qué es Gestión de Configuración de Software, se describe cómo se realiza este proceso y la importancia que tiene, todo este estudio fue realizado a través de la determinación de los puntos de contacto entre dos libros muy importantes: La guía para el cuerpo del conocimiento de la Ingeniería de software (SWEBOK)[SWEBOK, 2004] y Orientaciones curriculares para programas de estudio de pregrado en Ingeniería del software (SEEK)[SEEK, 2004]. También se dan un conjunto de definiciones relacionadas con Multimedia, y se desarrolla un estudio sobre las metodologías y herramientas más usadas en la creación de Software multimedia, fundamentándose luego cuál utilizar para ello y los diferentes lenguajes, tanto de programación como de modelado usados en el desarrollo de la aplicación.

1.1 SWEBOK Y SEEK

La Gestión de Configuración de Software es un proceso imprescindible en el ciclo de desarrollo de un producto software. No se puede desarrollar el diseño de un curso de GC sin consultar dos documentos claves, ellos son SWEBOK y SEEK. El SWEBOK fue un proyecto de la IEEE Computer Society con el objetivo de proporcionar una caracterización validada por consenso de los aspectos relacionados a la disciplina Ingeniería de Software y ofrecer un acceso por tópicos al cuerpo del conocimiento que soporta esa disciplina. En este libro son identificadas 10 áreas del conocimiento y una de ellas es la Gestión de Configuración de Software. En la Tabla. 1 se muestra esta Área de Conocimiento con sus subareas definidas las cuales caracterizan todo el proceso de Gestión de la Configuración.

Tabla 1. Área de Conocimiento GCS según SWEBOK

Gestión de Configuración de Software	
1. Gestión de los procesos de GCS	
	Contexto Organizacional para la GCS
	Restricciones y Guías para la GCS
	Planificación para la GCS
	Plan de la GCS
	Seguimiento en la GCS
2. Identificación de la Configuración de Software	
	Identificación de los elementos a ser controlados
	Biblioteca de Software
3. Control de la Configuración de Software	
	Solicitud, Evaluación y Aprobación de Cambios del Software
	Implementación de los Cambios en el Software
	Desviaciones y Renuncias
4. Contabilidad del Estado de la Configuración del Software	
	Información del Estado de la Configuración del Software
	Reportes del Estado de la Configuración del Software
5. Auditoría de la Configuración del Software	
	Auditoría Funcional de la Configuración del Software
	Auditoría Física de la Configuración de Software
	Auditoría In-process de la Línea Base de Software
6. Gestión del release y entrega del Software	

Construcción del Software
Gestión del release del Software

Además se realiza un estudio del SEEK, en éste libro también está presente la Gestión de Configuración como área de conocimiento y se define un conjunto de unidades y tópicos que no pueden dejar de ser impartidos. En la Tabla 2 se muestra dicha área de conocimientos.

Tabla 2. Área de Conocimiento GCS según SEEK

Gestión de Configuración de Software
1. Control de las Revisiones
2. Gestión del Release
3. Herramientas de Soporte
4. Construcción
5. Proceso de Gestión de la Configuración de Software
6. Problemas de mantenimiento

Luego de esta investigación se determina hacer un estudio para incluir en la multimedia los siguientes temas: el Proceso de Gestión de la Configuración de Software detallado en la tabla 1 y además las Herramientas de Soporte.

1.2 GESTIÓN DE CONFIGURACIÓN DE SOFTWARE

En el SWEEBOK se destacan conceptos importantes tales como el de sistema, configuración de un sistema, gestión de configuración. Sistema según la IEEE 610.12-90 puede ser definido como una colección de componentes organizados para cumplir una función específica o un grupo de funciones. La configuración de un sistema son las características funcionales y/o físicas de hardware, firmware o software, o una

combinación de las mismas, como se establece en la documentación técnica y puesto de manifiesto en un producto. También puede ser una colección de versiones específicas de hardware, firmware o elementos de software combinados, elaborados para un propósito específico. Gestión de Configuración, entonces, es la disciplina que se encarga de la identificación de la configuración de un sistema en distintos puntos en un período de tiempo, con el propósito de controlar sistemáticamente los cambios en la configuración, y mantener la integridad y traceabilidad de la configuración a través del ciclo de vida del sistema. La Gestión de la Configuración es formalmente definida en (IEEE610.12-90) como: "Una disciplina que aplica direcciones técnicas y administrativas y seguimiento para: identificar y documentar las características físicas y funcionales de un elemento de configuración, controlar los cambios de esas características, almacenar y reportar cambios procesando e implementando estados, y verifica el cumplimiento con requisitos específicos".

La GCS es un proceso de soporte del ciclo de vida del software el cual beneficia las actividades de administración, desarrollo y mantenimiento del proyecto, y finalmente para los usuarios y clientes a los cuales está dirigido el producto.

Las actividades de GCS son: planificación y gestión de los procesos de GCS, identificación de la Configuración de software, Control de la Configuración de Software, Contabilidad del estado de la configuración de Software, auditoria de la Configuración de Software y gestión del release del software y entrega. A continuación se explicará cómo se realiza el Proceso de Gestión de Configuración de acuerdo a estas actividades definidas en el SWEBOK.

1.2.1 GESTIÓN DE LOS PROCESOS DE GCS

La GCS controla la evolución y la integridad de un producto, identificando sus elementos, gestionando y controlando cambios, y verificando, almacenando y produciendo informes sobre la información de configuración.

Para lograr una exitosa implementación de la GCS es necesario una cuidadosa planificación y gestión. Esto, a su vez, requiere una comprensión del contexto organizacional para los requisitos impuestos sobre el diseño e implementación del proceso de GCS. Es entonces cuando comienza a jugar un papel importante el contexto

organizacional. Es necesario entender las relaciones entre los elementos organizacionales pues la GCS interactúa con muchos de estos elementos.

Normalmente es un elemento definido de la organización o un individuo especialmente designado quien tiene la responsabilidad total de la GCS.

La GCS puede interactuar con las actividades de aseguramiento de la calidad de software de la organización en lo que se refiere a temas tales como la gestión de registro y los puntos de no-conformidad.

Un aspecto importante definido dentro de esta actividad de gestión y planificación de los procesos de de GCS son las restricciones y guías para este proceso. Estas vienen de un gran número de fuentes. Las políticas y los procedimientos que se establecen en el nivel corporativo u otros niveles organizacionales, pueden influenciar o prescribir el diseño y la implementación de un proceso de GCS para un proyecto dado. Además, el contrato entre el cliente y el suministrador puede contener provisiones afectando el proceso de GCS. Finalmente, el proceso del ciclo de vida del software elegido para un proyecto de software en particular y las herramientas seleccionadas para implementar el software, afectan el diseño y la implementación del proceso de GCS.

1.2.1.1 PLANIFICACIÓN DEL PROCESO DE GCS

La planificación de un proceso de GCS para un proyecto dado debe ser consistente con el contexto organizacional, con restricciones aplicables, las guías comúnmente aceptadas, y naturaleza del proyecto. Las actividades más importantes cubiertas por la planificación son: Identificación de la Configuración, Control de la Configuración de Software, Contabilidad del estado de la Configuración de software, Auditoría de la Configuración y Gestión del release del software y entrega. Adicionalmente a lo expuesto anteriormente, asuntos tales como organización y responsabilidades, recursos y planes de trabajo, selección de herramientas e implementación, control del vendedor y el subcontratista, y control de interfaz son típicamente considerados. Los resultados de la actividad de planificación son almacenados en un Plan de GCS (PGCS), el cual está típicamente sujeto a auditoría y revisiones de Aseguramiento de la Calidad de Software.

Para prevenir confusión acerca de quién realiza una actividad o tarea dada de GCS, la organización involucrada en el proceso de GCS necesita ser identificada con claridad. Las responsabilidades específicas para actividades dadas de GCS necesitan ser asignadas a entidades organizacionales, o por título o por elemento organizacional.

Dentro de la planificación de la GCS se identifica el personal y las herramientas involucradas en llevar a cabo las actividades y tareas de GCS. Se abordan cuestiones de planificación estableciendo las secuencias necesarias de tareas de GCS y la identificación de sus relaciones con los planes del proyecto e hitos establecidos en la etapa de planificación de la gestión del proyecto. También son especificados algún requerimiento de entrenamiento necesario para la implementación de los planes y entrenamiento de nuevos miembros del personal.

Los diferentes tipos de capacidades de las herramientas y los procedimientos para su uso apoyan las actividades de GCS. El apoyo de herramientas automáticas comienza a ser cada vez más importante y difícil de establecer mientras los proyectos crecen en tamaño y el entorno del proyecto se hace más complejo.

En la planificación, los ingenieros de software eligen las herramientas de GCS apropiadas para el trabajo y se consideran problemas que pueden surgir en la implementación de estas herramientas.

Otro elemento a tener en cuenta en la planificación de la GCS, como se había mencionado antes, es el control del vendedor y el subcontratista. Un proyecto de software puede adquirir o hacer uso de productos de software comprados, tales como compiladores u otras herramientas. La planificación de la GCS considera cómo estos elementos serán tomados bajo control de la configuración y cómo los cambios y las actualizaciones serán evaluadas y gestionadas.

También se necesitan establecer los requisitos de GCS a ser impuestos en el proceso de GCS del subcontratado como parte del subcontrato y los medios para supervisar el cumplimiento.

Cuando un elemento de software interacciona con otro elemento de software o hardware, un cambio a cualquiera de los dos elementos puede afectar al otro. La planificación para

los procesos de GCS también considera cómo se identificarán los elementos que interaccionan y cómo se gestionarán y comunicarán los cambios a dichos elementos.

Los resultados de la planificación de la GCS para un proyecto dado se guardan en un Plan de Gestión de la Configuración del Software (PGCS), que es un “documento vivo” que sirve como referencia para los procesos de la GCS. El documento se mantiene (o sea, se actualiza y aprueba) según vaya siendo necesario durante el ciclo de vida del software. Después de que el proceso de la GCS ha sido implementado, puede ser necesario un cierto grado de seguimiento para asegurarse de que las provisiones del PGCS se llevan a cabo correctamente.

Se pueden diseñar medidas en la GCS para proporcionar información específica acerca de la evolución del producto o una visión interna de cómo funcionan los procesos de la GCS. Las mediciones de procesos de la GCS proporcionan un buen medio para monitorizar la efectividad de las actividades de la GCS de una manera continuada.

1.2.2 IDENTIFICACIÓN DE LA CONFIGURACIÓN DE SOFTWARE

La actividad de identificación de la configuración de software identifica elementos que se han de controlar; establece esquemas de identificación para los elementos y sus versiones, las herramientas y técnicas que se usarán para adquirir y gestionar los elementos controlados. Estas actividades proporcionan la base para las otras actividades de la GCS.

Un primer paso para controlar cambios es identificar los elementos de software a ser controlados. Esto requiere comprender la configuración del software en el contexto de la configuración del sistema, seleccionando elementos de configuración de software, desarrollando una estrategia para etiquetar elementos de software y describir las relaciones entre ellos e identificar las líneas base que se usarán, además de los procedimientos de adquisición de elementos para una línea base.

Un elemento de configuración de software (ECS) es una agregación de software, designada para la gestión de configuración y es tratada como una sola entidad en el proceso de la GCS (IEEE610.12-90). Los planes, documentación de especificaciones y diseño, material de pruebas, herramientas de software, código fuente y ejecutable,

bibliotecas de código, datos y diccionarios de datos y documentación para la instalación, mantenimiento, operación y uso del software, están entre los elementos de software con potencial para convertirse en ECSs.

La selección de ECSs es un proceso importante en el que se ha de conseguir un equilibrio entre proporcionar una visibilidad adecuada para el control del proyecto y proporcionar un número manejable de elementos a controlar.

Los elementos de software evolucionan al mismo tiempo que el proyecto de software avanza. Una versión de un elemento de software es un elemento identificado y especificado particularmente. Se puede pensar en ella como el estado de un elemento que evoluciona. Se define revisión como una versión que se construye sobre otra versión anterior. Una variante es una nueva versión de un elemento que se añadirá la configuración sin reemplazar la versión anterior.

Una línea base de un software es un conjunto de elementos de configuración del software formalmente designados y fijados en un momento determinado durante el ciclo de vida del software. El término se usa también para referirse a una versión en particular de un elemento de la configuración del software acordada previamente. En cualquiera de los casos, la línea base solo se puede cambiar por medio de procedimientos formales de control de cambios. Una línea base representa, junto con todos los cambios aprobados para la línea base, la configuración actual aprobada.

Las líneas base a usar en un proyecto determinado, junto con los niveles de autoridad asociados necesarios para la aprobación de cambios, se identifican normalmente en el Plan de Gestión de la Configuración del Software.

Los elementos de configuración del software se ponen bajo el control de la GCS en diferentes momentos; lo que significa que se añaden a una línea base en particular en momentos específicos del ciclo de vida del software. El evento que da comienzo al proceso es la terminación de alguna tarea formal de aceptación, como una revisión formal.

Seguidamente de la adquisición de un ECS, los cambios a dicho elemento se deben aprobar formalmente de la manera apropiada para el elemento y la línea base

involucrados, como se define en el PGCS. Después de la aprobación, el elemento se incorpora en la línea base del software siguiendo el procedimiento apropiado.

Otro elemento muy importante son las Bibliotecas de Software, definidas por la IEEE610.12-90 como “una colección controlada de software y documentación relacionada al mismo, diseñada para ayudar en el uso, el desarrollo y el mantenimiento del software”. El número de bibliotecas corresponde con el nivel de madurez de los ECS que son colocados en las mismas. Las bibliotecas también son usadas durante las actividades de gestión del release y entrega de software. La seguridad, en términos de control de acceso y medios de copia de seguridad, es un aspecto clave en la gestión de bibliotecas. La(s) herramienta(s) que se usan en cada biblioteca deben soportar el control de la GCS que sea necesario para dicha biblioteca, en términos de control de los ECSs y de acceso a la biblioteca.

1.2.3 CONTROL DE LA CONFIGURACIÓN DEL SOFTWARE

EL control de la configuración del software se refiere a la gestión de cambios durante el ciclo de vida del software. Cubre los procesos que determinan qué cambios hacer, la autoridad para aprobar ciertos cambios, el soporte para la implementación de dichos cambios y el concepto de desviación formal de los requisitos del proyecto, además de las cancelaciones de éstos. La información derivada de estas actividades es útil para medir el tráfico de cambios, ruptura y aspectos por rehacer.

El primer paso en la gestión de cambios a elementos controlados, es determinar qué cambios hacer. El proceso de solicitud de cambio del software proporciona procedimientos formales para recoger y registrar solicitudes de cambios, evaluando el coste e impacto potencial de un cambio propuesto y aceptar, modificar o rechazar dicho cambio. Las solicitudes de cambios a elementos de la configuración del software las puede originar cualquiera durante cualquier momento del ciclo de vida del software y puede incluir una solución propuesta y una prioridad de solicitud. El tipo de cambio (por ejemplo, un defecto o mejora) se registra normalmente en la Solicitud de Cambios de Software (SCS), sin importar la fuente.

Una vez que se ha recibido una SCS, se realiza una evaluación técnica (también conocida como análisis del impacto) para determinar el tamaño de las modificaciones necesarias en

caso de que se aceptara la solicitud de cambio. Finalmente, la evaluación de los aspectos técnicos y de gestión de la solicitud de cambios, será realizada por una autoridad establecida, de acuerdo con la línea base afectada, el ECS involucrado y la naturaleza del cambio y entonces se aceptará, modificará, rechazará o pospondrá el cambio propuesto.

La autoridad para aceptar o rechazar los cambios propuestos, es normalmente la responsabilidad de una entidad conocida como Comité de Control de la Configuración y Cambios (CCCC). Las actividades del CCCC están sujetas normalmente a auditorías de la calidad de software o revisiones.

Un proceso efectivo de solicitud de cambio del software (SCS) requiere el uso de herramientas de soporte y procedimientos, desde formularios de papel y un procedimiento documentado hasta la herramienta electrónica para generar solicitudes de cambios. Un enlace entre las capacidades de esta herramienta y el sistema de reporte de errores puede facilitar el seguimiento de soluciones para los informes de errores.

Las SCSs aprobadas se implementan utilizando los procedimientos de software definidos, de acuerdo con los requisitos de planes aplicables. Como se podría implementar simultáneamente un número de SCSs, es necesario proporcionar un medio para el seguimiento de cuáles SCSs son incorporadas dentro de una versión y línea base particular del software. Como parte de la finalización del proceso de cambios, los cambios completados podrían someterse a auditorías de configuración y verificación de la calidad del software. Esto incluye asegurarse de que solo se han realizado los cambios aprobados.

La implementación real de un cambio está soportada por las capacidades de la herramienta de la biblioteca, que proporciona gestión de versiones y soporte para el repositorio de código.

1.2.4 CONTABILIDAD DEL ESTADO DE LA CONFIGURACIÓN DEL SOFTWARE

La contabilidad del estado de la configuración del software (CECS) es la actividad de registrar y proporcionar la información necesaria para una gestión efectiva de la configuración del software.

La actividad de la CECS diseña y opera un sistema para la captura y generación de los informes necesarios durante el ciclo de vida. Como en cualquier sistema de información, se debe identificar, recoger y mantener la información del estado de la configuración que se ha de gestionar según las configuraciones evolucionan.

Los reportes de información pueden ser usados por varios elementos de la organización y del proyecto, incluyendo el equipo de desarrollo, el equipo de mantenimiento, la gestión del proyecto y las actividades de calidad de software. Los informes pueden tener la forma de respuestas inmediatas a preguntas específicas o ser informes prediseñados producidos periódicamente. Alguna de la información producida por las actividades de contabilidad del estado durante el curso del ciclo de vida podría acabar siendo registros de la garantía de la calidad.

Además de informar el estado actual de la configuración, la información obtenida por la CECS puede usarse como base para varias mediciones útiles para la gestión, desarrollo y la GCS. Un ejemplo incluye el número de cambios solicitados por ECS y el tiempo medio necesario para implementar una solicitud de cambio.

1.2.5 AUDITORÍA DE LA CONFIGURACIÓN DEL SOFTWARE

La auditoría de software es una actividad que se realiza para evaluar independientemente la conformidad de productos de software y procesos con regulaciones, estándares, guías, planes y procedimientos.

La actividad de auditoría de la configuración del software determina el grado en que un elemento satisface las características funcionales y físicas requeridas. Se pueden realizar auditorías informales de este tipo en momentos clave del ciclo de vida. Hay dos tipos de auditorías formales que podrían ser requeridas por el contrato: la Auditoría funcional de la Configuración y la Auditoría Física de la Configuración. El completar con éxito estas auditorías puede ser un prerrequisito para establecer la línea base del producto.

El propósito de la Auditoría Funcional de la Configuración de Software es asegurarse de que el elemento de software que se audita es consistente con las especificaciones que lo rigen. Los resultados de las actividades de verificación y validación del software son la clave de entrada para esta auditoría. Mientras que la Auditoría Física de la Configuración

de Software tiene el propósito de asegurar que el diseño y la documentación de referencia son consistentes con el producto de software tal y como se ha construido.

Durante el proceso de ingeniería del software se pueden llevar a cabo auditorías para investigar el estado actual de elementos específicos de la configuración o para valorar la implementación del proceso de la GCS. Las auditorías in-process de la GCS proveen un mecanismo más formal para monitorizar aspectos seleccionados del proceso, siendo un elemento importante dentro de todo el seguimiento que proporciona la GCS.

Se podría aplicar una auditoría a elementos seleccionados de la línea base para asegurarse de que el rendimiento es consistente con las especificaciones o para asegurarse de que la documentación continúa siendo consistente con el elemento de la línea base que se está desarrollando.

1.2.6 GESTIÓN DEL RELEASE Y ENTREGA DEL SOFTWARE

El término “release” se usa en este contexto para referirse a la distribución de un elemento de la configuración de software fuera de la actividad de desarrollo. Esto incluye tanto lanzamientos internos como la distribución a clientes. Cuando diferentes versiones de un elemento de software están disponibles para ser entregadas, como versiones para diferentes plataformas o versiones con diferentes capacidades, es normalmente necesario preparar una versión específica y empaquetar los materiales adecuados para distribuirla.

La construcción del software es la actividad de combinar las versiones correctas de elementos de configuración del software, usando la configuración de datos apropiada, dentro de un programa ejecutable para su distribución a los clientes u otros receptores, como la actividad de pruebas. Las instrucciones de construcción se aseguran de que se toman los pasos de construcción adecuados y en la secuencia correcta. Además de construir software para un nuevo lanzamiento, la GCS normalmente necesita ser capaz de reproducir releases previos para recuperación, pruebas, mantenimiento u otros propósitos de lanzamiento adicionales.

El software se construye usando versiones particulares de la herramientas de soporte como compiladores. Podría ser necesario reconstruir una copia exacta de un elemento de configuración que se haya construido previamente. En ese caso, las herramientas de

soporte y las instrucciones de construcción asociadas deben estar bajo el control de la GCS, para asegurarse de la disponibilidad de las versiones correctas de las herramientas.

Las capacidades de las herramientas son útiles para seleccionar la versión correcta de elementos de software, para un determinado objetivo de entorno y para automatizar el proceso de construcción del software con las versiones seleccionadas y los datos de configuración apropiados. En proyectos grandes con desarrollo en paralelo o en entornos de desarrollo distribuido, estas capacidades de las herramientas son necesarias.

La gestión de lanzamiento del software conlleva la identificación, empaquetamiento y distribución de los elementos de un producto, por ejemplo, programas ejecutables, documentación, notas del release y datos de configuración. Dado que los cambios del producto pueden ocurrir constantemente, una de las preocupaciones en la gestión del release es determinar cuándo realizar un lanzamiento. La tarea de empaquetamiento debe identificar qué elementos del producto se deben distribuir y por tanto seleccionar las variantes correctas de dichos elementos, dada la aplicación que se le quiere dar al producto. La información que documenta el contenido físico del release se conoce como documento de descripción de la versión. Las notas del release normalmente describen nuevas capacidades, problemas conocidos y requisitos necesarios de la plataforma para la operación adecuada del producto. El paquete a ser lanzado también contiene instrucciones de instalación o actualización. Finalmente, en algunos casos, se podría requerir la actividad de gestión del release para el seguimiento de la distribución del producto a varios clientes o sistemas objetivo.

1.2.7 LAS HERRAMIENTAS DE GESTIÓN DE CONFIGURACIÓN DE SOFTWARE

Otro tema importante tratado por el SEEK y presente también en el SWEBOOK es lo relacionado a las herramientas para la gestión de la configuración que han sido divididas, de acuerdo al SWEBOOK, en tres categorías: seguimiento, gestión de versiones, y herramientas de liberación.

- ✓ Herramientas relacionadas con los defectos, mejoras, problemas, y seguimiento de problemas: Estas herramientas son usadas en la conexión de los problemas de seguimiento asociados con un producto de software en particular.

- ✓ Herramientas de gestión de versiones: Estas herramientas están implicadas en la gestión de múltiples versiones de un producto.
- ✓ Herramientas de liberación y construcción: Estas herramientas son usadas para la gestión de las tareas de liberación y construcción de software. La categoría incluye las herramientas de instalación que han sido ampliamente usadas para configurar la instalación de productos de software.

1.3 ENSEÑANZA DE GESTIÓN DE CONFIGURACIÓN DE SOFTWARE EN LA ACTUALIDAD

La preparación del personal dedicado a la Gestión de la Configuración de Software debe ser cada vez mayor, debido a la importancia que tiene para el proceso de desarrollo de software. En la actualidad no existe una herramienta para la enseñanza de Gestión de Configuración, aunque este tema se imparte en algunas universidades como parte del plan de estudio del centro. Es decir, que a pesar de lo urgente que resulta la preparación de personal en esta temática, y a pesar de que la misma se imparte en varias universidades importantes, el apoyo a la enseñanza mediante algún software educativo es prácticamente nulo.

Ejemplo de ello es la Universidad Privada Ricardo Palma, de Chile, donde la facultad de Ingeniería Informática imparte un curso llamado Evolución de Software. Las unidades de aprendizaje de esta asignatura son: Gestión de Configuración de Software, Planeamiento y Organización de la GCS; Identificación de la GCS; Administración y Control de Control de Cambios; Contabilidad del Estado; Auditoría de la GCS; Administración de la Biblioteca de Software y Lanzamientos y Mantenimiento de Software, temas expuestos por el profesor a través de conferencias.

También el IT Institute (Instituto de Tecnología en la Información), centro de formación avanzada en temas selectos de tecnologías de información con acreditación, imparte un curso presencial que introduce a los participantes en la disciplina de Gestión de Configuración de Software. Dicho curso resulta ideal para todos aquellos administradores de la configuración que busquen aprender qué información es crítica para ser incluida en el plan de administración de la configuración.

Por su parte, la compañía SQS S.A (acrónimo de las palabras en inglés Software Quality Systems), líder en servicios de Consultoría de Calidad de Software y Prueba, ofrece un curso presencial de Gestión de Configuración de Software, en Bilbao, Madrid y Barcelona.

En la UCI la enseñanza de Gestión de Configuración se incluye dentro de la disciplina de Ingeniería de Software II como parte de los flujos de trabajo de la metodología RUP. Este tema es impartido en el aula por los profesores y además está disponible en el EVA (Entorno Virtual de Aprendizaje) de la Universidad. En el EVA se encuentran varios materiales para el estudio individual de cada estudiante y también pruebas online a las que los estudiantes podrán acceder para autoevaluarse y de esta forma el profesor también tener un criterio más de evaluación de cada uno de ellos.

Además, la Gestión de la Configuración de Software, constituye una asignatura independiente que forma parte de uno de los Cursos del Perfil: Calidad de Software. En ella se imparten los principales conceptos, habilidades y herramientas necesarias para poder llevar a cabo esta actividad durante el desarrollo de un proyecto software, disponible también en el EVA.

Este trabajo de diploma tiene el objetivo de apoyar el proceso de enseñanza de la Gestión de la Configuración, proporcionando un producto multimedia sobre el tema a todos los estudiantes que se relacionen de una forma u otra con éste.

1.4 MULTIMEDIA, CARACTERÍSTICAS Y CONCEPTOS RELACIONADOS

Desde su mismo surgimiento, el software multimedia supuso un replanteamiento de la manera de entender los programas informáticos hasta ese entonces. La combinación de lenguajes y las utilísimas maneras de enlazar información, hicieron de la multimedia un producto rápidamente incorporado a casi todas las esferas de la vida. La educación se nutrió muchísimo de las mismas, pues encontraron en ellas una forma alternativa y amena de transmitir conocimientos.

El término Multimedia en el mundo de la computación es la forma de presentar información que emplea una combinación de texto, sonido, imágenes, vídeo y animación. La mayoría de las aplicaciones multimedia incluyen asociaciones predefinidas conocidas como hipervínculos o enlaces, que permiten a los usuarios moverse por la información de

modo intuitivo. La conectividad que proporcionan los hipertextos hace que los programas multimedia no sean presentaciones estáticas con imágenes y sonido, sino una experiencia interactiva infinitamente variada e informativa. (Menjivar, 2001)

Asociados al término multimedia se encuentran conceptos que comúnmente logran confusión, estos son hipermedia e hipertexto, es importante comprender bien ambos conceptos.

Hipertexto son datos que contienen enlaces a otros datos. Pueden encontrarse en cualquier lugar dentro de un documento. Cada enlace está marcado de alguna forma que lo destaca. Una palabra que tiene un enlace puede estar resaltada o subrayada, o bien identificada por un número. En el lenguaje de Web un documento de hipertexto es algo que contiene datos y posiblemente, enlace a otros documentos. (CICEI 1995).

El termino Hipermedia, combinación de los conceptos Hipertexto y Multimedia, hace referencia a una tecnología de construcción de (híper) documentos que permite a los lectores encontrar fácilmente la información que realmente necesitan, de la manera que ellos decidan, a través de enlaces establecidos por el autor entre los diferentes elementos de información Multimedia (texto, sonido, imagen, vídeo, etc.) que conforman el documento. (BARROS 2004).

La enseñanza interactiva le permite al estudiante un alto grado de control del proceso de aprendizaje, con lo que se estará muy cerca de conseguir la tan ansiada individualización de la enseñanza. Además puede favorecer también los procesos de enseñanza y aprendizaje grupales. La auténtica interactividad implica poner el control del proceso de aprendizaje en manos de los aprendices.

Multimedia interactiva implica un diálogo entre el usuario y los contenidos. El grado de esta actividad es lo que va a hacer aprender y retener más rápido aquello que se propone el usuario.

1.5 TENDENCIAS TECNOLÓGICAS ACTUALES

1.5.1 HERRAMIENTAS PARA LA CREACIÓN DE SOFTWARE MULTIMEDIA

Debido al gran avance tecnológico que existe, actualmente se ha desarrollado un conjunto de herramientas para el desarrollo de software multimedia. Cada una de ellas cuenta con características diferentes. Para elaborar la aplicación, se ha realizado un estudio de las principales herramientas que la autora propone para el desarrollo de un software multimedia. Además, se muestra una tabla con una comparación entre estas herramientas.

TOOLBOOK

ToolBook es una herramienta que principalmente está destinada a la creación de CBTs(Computer Based Training o formación basada en ordenador). Para ello hace uso de la metáfora de un libro y utiliza un lenguaje de programación propio: OpenScript. Con el asistente Book Specialist se va guiando paso a paso al usuario por la creación de una nueva aplicación. Este asistente contempla los múltiples parámetros a tener en cuenta a la hora de realizar un CTB. Una vez creado el proyecto utilizando un prototipo, se entra en el editor principal, desde el cuál se controlará todo el proceso de creación. Uno de los aspectos más importantes es el lenguaje de programación. ToolBook es una herramienta muy flexible, pudiendo adaptarse de forma sencilla (mediante programación) a todo aquello que deseemos. La clave de toda la programación radica en la orientación a objetos que ToolBook utiliza como arquitectura principal. Esto permite compartir guiones entre distintos objetos, sin necesidad de tener que reescribir el código fuente (también soportado por la nueva versión de Director, aunque salvando las distancias). La consecuencia inmediata es que el mismo producto incorpora una gran cantidad de guiones prescritos, de forma que únicamente es necesario buscar el guión adecuado y asignarlo al objeto correspondiente. (Merino, 2008)

AUTHORWARE

El Authorware es una herramienta que permite integrar múltiples medios. Es una herramienta basada en íconos. Esto quiere decir que hace uso de ellos para insertar o integrar diferentes tipos de medios. Los medios han sido previamente creados, editados o

preparados. Es posible que estos medios dependiendo del formato, cuenten con alguna información adicional, la cual puede ser aprovechada o interpretada por el Authorware.

Macromedia Authorware integra gráficos, sonido, animación, texto y video dentro de experiencias de aprendizaje cautivantes y ricas en medios. La conectividad con bases de datos y el registro/seguimiento de datos integrado, permiten registrar y monitorear más fácilmente el desempeño y resultados del estudiante, lo cual es importante para evaluar la efectividad de la capacitación y el retorno sobre la inversión. Las aplicaciones de Authorware pueden distribuirse fácilmente a través de la Web, redes corporativas y CD-ROMs, ofreciendo resultados consistentes y de alta calidad en todas las plataformas. (SOLENZAL y DÍAZ, 2006)

MACROMEDIA DIRECTOR

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

Macromedia Director es uno de los entornos de autor multimedia más potentes para crear aplicaciones y contenido interactivo, dinámico y acabados de alta calidad desplegados en CD/DVD-ROM, kioscos, y la Web. Macromedia Director MX incluye integración completa y compatible con Macromedia Flash TM, adopción de la interfaz de usuario Macromedia MX, soporte Mac OS X y nuevos flujos de trabajo más eficaces. A pesar de las funcionalidades y ventajas anteriormente señaladas, es importante destacar que los archivos creados con esta poderosa herramienta no pueden ser visualizados en entornos GNU/Linux. (BIRKDAR, 2003)

ADOBE FLASH CS3 PROFESSIONAL

Adobe Flash (previamente llamado Macromedia Flash) es una plataforma multimedia originalmente adquirida por Macromedia y actualmente desarrollada y distribuida por Adobe.

Desde su introducción en 1996, flash se convirtió en un popular método para crear animaciones e interactividad para la web. Por la flexibilidad y facilidad que brinda para combinar elementos gráficos y medias, los desarrolladores de multimedia adoptaron esta poderosa herramienta para el desarrollo sus productos. La primera versión fue lanzada bajo el nombre de Macromedia Flash 1, así fueron desarrollándose un grupo de versiones todas con sus propias innovaciones, además de continuar con las características de la versión anterior: Macromedia Flash 2, Macromedia Flash 3, Macromedia Flash 4, Macromedia Flash 5, Macromedia Flash MX, Macromedia Flash MX 2004, Macromedia Flash MX Profesional 2004, luego aparece el tan popular Macromedia Flash 8, con sus versiones Macromedia Flash Basic 8 y Macromedia Flash Professional 8. Posteriormente a esto, es lanzado el Adobe Flash CS3 Professional como la versión 9 del flash, publicado el 16 de abril de 2007. Es la primera versión de Flash publicada bajo el nombre de Adobe para luego lanzar la última versión Adobe Flash CS4 Professional.

Flash puede manipular vectores e imágenes de formatos clásicos como jpg, bmp entre otras, además soporta flujo bidireccional para audio y video. La interfaz de programación de Flash está basada en JavaScript, pero con base en este lenguaje, fue creado ActionScript, el lenguaje más potente para la creación de script

Diversos software y dispositivos son capaces de crear y reproducir contenidos flash, incluyendo Adobe Flash Player, el cual está disponible para la mayoría de navegadores webs y sistemas operativos. Las extensiones que utilizan sus archivos son swf, tradicionalmente llamados "ShockWaveFlash" y van desde la primera versión Macromedia Flash 1 hasta la actual Adobe Flash CS4 Professional.

Este estudio se centra en el Adobe Flash CS3 Professional que, a diferencia de su sucesor Adobe Flash CS4 Professional, requiere una estación de trabajo con menos prestaciones.

En resumen podemos enumerar las mejoras del Flash CS3 con respecto a versiones anteriores que son de utilidad para la realización de este producto:

- ✓ **La Interfaz CS3:** La interfaz de usuario de Flash se ha actualizado para compartir una interfaz común con otros productos de Adobe Creative Suite CS3. El aspecto similar de todas las aplicaciones de Adobe facilita el trabajo de los usuarios con diferentes aplicaciones. Esta característica es muy importante, pues para el desarrollo de la aplicación es necesario usar otras herramientas para la edición de videos, de imágenes, y dentro de esta Suite se encuentran herramientas para ello.
- ✓ **El Cuadro de diálogo “Elemento de biblioteca de símbolos de mapa de bits”:** Este cuadro de diálogo se ha ampliado para ofrecer una mayor previsualización del mapa de bits.
- ✓ **Recuadros de delimitación multicolores:** Se puede cambiar el color de selección de tipos de elementos específicos para identificar cada elemento más fácilmente.
- ✓ **Detecciones de contenido activo:** Para que los usuarios no necesiten activar primero Flash Player para interactuar con el contenido de Flash, Flash publica las plantillas HTML que se pueden utilizar para incorporar los archivos SWF de Flash. Con estas plantillas, los archivos SWF incorporados se activan uniformemente sin necesidad de hacer clic en el ratón u otra activación del usuario.
- ✓ **Previsualización en el escenario de 9 divisiones:** La escala de 9 divisiones ofrece previsualización en el escenario, por lo que se pueden ver los cambios y ajustes realizados a clips de película en el escenario.
- ✓ **Operación de copiado y pegado de filtros:** Ahora se pueden copiar y pegar configuraciones de filtros gráficos de una instancia a otra.
- ✓ **Copiar y pegar movimiento:** Copiar y pegar movimiento permite copiar una interpolación de movimiento y pegar (o aplicar) los fotogramas, interpolaciones e información de símbolos en otro objeto.
- ✓ **Mejoras en la herramienta Pluma:** Se han introducido mejoras a la herramienta Pluma.
- ✓ **Importación de Adobe Photoshop:** Ahora puede importar archivos PSD de Adobe Photoshop directamente a los documentos de Flash.

- ✓ **Importación de Adobe Illustrator:** Ahora se pueden importar los archivos AI de Adobe Illustrator directamente a los documentos de Flash.
- ✓ **Herramientas de dibujo Rectángulo simple y Óvalo simple:** Las nuevas herramientas de dibujo Rectángulo simple y Óvalo simple permiten crear rectángulos y óvalos con propiedades editables en cualquier momento desde el inspector de propiedades. (Adobe, 2008)

COMPARACIÓN ENTRE HERRAMIENTAS

A continuación se muestra una tabla donde se aprecia una comparación entre las principales herramientas profesionales para desarrollar software multimedia, en la cual se evalúan un grupo de características de las mismas.

Tabla 3. Comparación entre herramientas para desarrollar Software Multimedia

Características	Toolbook	Authorware	Macromedia Director	Adobe Flash CS3
Fabricante	Asymetrix	Macromedia	Macromedia	Adobe
Idioma	Inglés	Inglés	Inglés	Inglés-Español
Drag and Drop	Sí	Sí	Sí	Sí
Fuentes Antidentadas	No	No	Sí	Sí
Soporte de DirectX	No	No	No	Sí
Contenedor ActiveX	Sí	Sí	Sí	Sí
Exportación HTML	Sí	No	No	Sí
Exportación Java	Sí	No	No	No
Optimización transmisión de recursos	Sí	Sí	Sí	Sí
Soporte de Audio	Sí	Sí	Sí	Sí
Empaquetador de proyectos	Sí	No	No	Sí

Capítulo 1: Fundamentación Teórica

Acceso al registro de Windows	Sí	No	No	No
Soporte para 32 Bits	Sí	Sí	Sí	Sí
Lenguaje de Programación	OpenScript	No	Lingo	ActionScript
Compatibilidad Macintosh	No	Sí	Sí	Sí
Formatos de imagen	AI, BMP, DIB, CGM, DRW, EPS, JPG, GIF, PCT, PCX, TIF, WMF	TIF, BMP, DIB, RLE, GIF, JPG, LRG, PNG, PICT, TGA, WMF, EMF, PSD	BMP, DIB, EPS, GIF, JPG, LRG, MacPaint, PIC, PNG, PNT, PSD, SWA, TARGA, TIF, WMF, PhotoCD	BMP, DIB, EPS, GIF, JPG, LRG, MacPaint, PIC, PNG, PNT, PSD, SWA, TARGA, TIF, WMF, PSD, PCT, QTIF, SGI
Formatos de animación	FLC, FLI	FLC, FLI, CEL	FLC, FLI	SWF
Formatos de audio	WAV, MID, RMI	AIF, WAV	AIF, AIFF-IMA, WAV	MP3, WAV, AIF
Formatos de vídeo	AVI, MOV, PIC	AVI, MOV, MPEG*	AVI, MOV	FLV, MOV, AVI, MPG, MPEG, DV, DVI, ASF
Entorno	Excelente	Bien	Muy Bien	Excelente
Trabajo con Recursos	Excelente	Bien	Muy Bien	Excelente
Flexibilidad	Excelente	Regular	Regular	Excelente
Compatibilidad	Mal	Muy Bien	Muy Bien	Excelente
Facilidad de uso	Bien	Muy Bien	Excelente	Muy bien
Lo Mejor	Muy flexible	Fácil de programar y compatibilida	Compatibilidad multiplataforma y sencillez	Muy flexible, compatibilidad multiplataforma,

		d con Macintosh	de uso	Lenguaje de programación orientado a objetos
Lo Peor	Poca compatibilida d y mucha programació n	Poco flexible	Requiere conocer Lingo	Necesario una estación de trabajo con buenas prestaciones

1.5.2 LENGUAJES UTILIZADOS EN LA MULTIMEDIA

EL LENGUAJE DE MARCAS XML

XML (Extensible Markup Language) es un metalenguaje que fue diseñado básicamente para estructurar, almacenar y para intercambiar datos entre diferentes aplicaciones. Se convirtió en un estándar, ya que es extensible y puede ser utilizado por cualquier aplicación independientemente de la plataforma.

XML es un lenguaje de marcas, por ello, sirve para definir de qué manera se debe estructurar la información para que después pueda ser interpretada. (Sherekan, 2008)

Entre sus principales ventajas se destacan:

- ✓ 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. (Wikipedia, 2008)

LENGUAJE DE PROGRAMACIÓN. ACTIONSCRIPT 2.0

ActionScript es el lenguaje de programación para crear scripts en Flash, no requiere la creación de un programa completo para que la aplicación alcance los objetivos. Una película de Flash está formada por una serie de fotogramas secuenciales llamada “línea de tiempo”. Utilizando ActionScript en ciertos fotogramas clave se puede crear elementos interactivos como:

- ✓ Botones clásicos de navegación que reaccionan a clics del ratón y envían la cabeza lectora de la película a un fotograma concreto.
- ✓ Contenido que se anima basándose en movimientos del ratón.
- ✓ Objetos que pueden ser movidos por el ratón o el teclado.
- ✓ Campos de texto que permiten a los usuarios entrar datos a la película como en un formulario.
- ✓ Controlar el contenido audiovisual.

ActionScript puede ser utilizado para examinar o modificar las propiedades de los elementos de una película. Por ejemplo, posibilita: (CIBERAULA, 2005)

- ✓ Cambiar el color y la localización de un objeto.
- ✓ Reducir el volumen de un sonido.
- ✓ Especificar la tipografía de un bloque de texto.
- ✓ Campos de texto que permiten a los usuarios entrar datos a la película como en un formulario.
- ✓ Modificar las propiedades repetidamente, produciendo comportamientos únicos como son los movimientos basados en la física y la detección de colisiones. (CIBERAULA, 2005)

1.5.3 METODOLOGÍAS A CONSIDERAR PARA EL DESARROLLO DE SOFTWARE

RUP

El proceso unificado de desarrollo (RUP) es una metodología para la ingeniería de software que va más allá del mero análisis y diseño orientado a objetos para proporcionar una familia de técnicas que soportan el ciclo completo de desarrollo de software. El resultado es un proceso basado en componentes, dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental.

RUP puede adaptarse a cualquier tipo de proyecto y empresa ya que basa su desarrollo en ciclos que, al concluir, originan una versión del producto. Cada ciclo consta de cuatro fases: Inicio, Elaboración, Construcción y Transición, llevando a cabo el cumplimiento de los objetivos propuestos y en su culminación el alcance de un hito siendo respectivamente: objetivos del ciclo de vida, arquitectura del ciclo de vida, funcionalidad operativa inicial y la versión del producto. Cada fase a su vez consta de varias iteraciones que definen los tiempos de ejecución, tratan los riesgos más importantes y se suceden de forma incremental a una etapa superior de producción garantizando el coste de riesgo del producto al de una iteración, manejando de forma más visible y progresiva los resultados, e identificando de manera creciente las necesidades y requisitos totales del usuario y el sistema (PRESSMAN 2001).

PROGRAMACIÓN EXTREMA (XP)

La Programación Extrema es una metodología ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Esta metodología fue desarrollada por Kent Beck. Surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos. XP se plantea como una metodología a emplear en proyectos de riesgo, y aumenta la productividad. (Fernández, 2002)

El ciclo de desarrollo consiste (a grandes rasgos) en los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.

3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

En todas las iteraciones de este ciclo, tanto el cliente como el programador, aprenden. No se debe presionar al programador a realizar más trabajo que el estimado, ya que se perderá calidad en el software o no se cumplirán los plazos. De la misma forma el cliente tiene la obligación de manejar el ámbito de entrega del producto, para asegurarse que el sistema tenga el mayor valor de negocio posible con cada iteración. (Alicante, 2003)

METODOLOGÍA DE ADMINISTRACIÓN DE RELACIONES (RMM)

El modelo RMM se define como un proceso de análisis, diseño y desarrollo de aplicaciones multimedia. Los elementos principales de RMM son el modelo E-R (Entidad-Relación) y el modelo RMDM (Relationship Management Data Model), basado en el modelo HDM. El modelo RMDM propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipertexto de la aplicación. Según sus autores, está orientada a problemas con datos dinámicos que cambian con mucha frecuencia, más que a entornos estáticos. (Izakowitz, 1995)

1.5.4 LENGUAJES DE MODELADO

UML

El lenguaje unificado de Modelado (UML) es un lenguaje de modelado visual utilizado para especificar, visualizar, construir y documentar artefactos de un sistema de software. Se usa para entender, diseñar, configurar, mantener y controlar la información sobre los sistemas a construir.

UML capta la información sobre la estructura estática y el comportamiento dinámico de un sistema. Un sistema se modela como una colección de objetos discretos que interactúan para realizar un trabajo que finalmente beneficia a un usuario externo. Pues ayuda al usuario a entender la realidad de la tecnología y la posibilidad de que reflexione antes de

invertir y gastar grandes cantidades en proyectos que no estén seguros en su desarrollo, reduciendo el costo y el tiempo empleado en la construcción de las piezas que constituirán el modelo. (Booch, 2000)

OMMMA-L

OMMMA – L, no es un nuevo lenguaje, sino una extensión del ya conocido UML, por lo que no es necesario aprenderlo, sino interpretar las características extendidas, centrados a la lógica de funcionamiento de una multimedia, que es por lo general, sencilla. Muestra análisis similares a otras metodologías y no se especializa en una clasificación de producto, sino que generaliza a través del uso de la semántica original de UML. Es robusto y altamente descriptivo, refleja el proceso en todas sus etapas.

Las características de OMMMA-L se pueden resumir en lo siguiente:

- ✓ Soporta el modelado de los aspectos estructurales, funcionales y dinámicos de un sistema interactivo y su interfaz de usuario.
- ✓ Se concentra en la funcionalidad desde la perspectiva del sistema de software.
- ✓ Su sintaxis es definida explícitamente.
- ✓ Tiene una semántica informal e intuitiva. (Monrose, 2004)

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 media. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.

- ✓ 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 diagramas tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de

visualización (texto, gráfico, video, animación) e interacción (scrolls, barras de menú, botones, campos de entrada y salida, hipertextos con hipervínculos).

- ✓ Vista de Comportamiento temporal predefinido: modelada por el Diagrama de Secuencia de OMMMAL, 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.
- ✓ 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, más 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 esto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

1.5.5 FUNDAMENTACIÓN DE LA HERRAMIENTA Y METODOLOGÍA A UTILIZAR

Para el desarrollo del software multimedia se ha seleccionado la herramienta Adobe Flash CS3 Profesional, pues ésta permite crear aplicaciones con tecnología multimedia de forma interactiva, posee un entorno muy potente para crear animaciones dinámicas, soportando video, entre otros elementos gráficos. Es una de las herramientas más utilizadas para la producción de software multimedia en la actualidad y de la que más tiene conocimiento la autora. Utiliza un lenguaje de creación de scripts con posibilidades enormes (ActionScript 2.0) y también orientado a objeto. Es compatible totalmente entre plataformas, sus aplicaciones pueden ser ejecutadas en Windows, Unix, en la web, etc., además posee una gran facilidad de uso al tener el aspecto similar de todas las aplicaciones de Adobe que se utilizan para el desarrollo del producto.

A partir del estudio de las principales metodologías que existen para el desarrollo de software, se ha decidido utilizar RUP, pues ésta presenta un conjunto de artefactos adecuados para software multimedia a través de la extensión de UML para multimedia, conocida por OMMMA – L, como son los diagramas de presentación y mapas de

navegación. Esta metodología proporciona una gran interacción con el usuario, posibilitando lograr una mejor calidad en el proceso de desarrollo.

CAPITULO 2: “DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA”

INTRODUCCIÓN

En el presente capítulo se realiza la descripción de la solución propuesta para cumplir el objetivo de esta investigación. Se define el dominio de la aplicación y los conceptos asociados a éste, realizándose el modelo conceptual. Se llevará a cabo el levantamiento de requisitos. Se identificará la audiencia a la cual va dirigida el producto multimedia. Se definen los mapas de navegación para obtener una visión general del sistema en forma de nodos. También se realizan y describen los casos de usos del sistema, logrando una mejor comprensión del flujo de eventos de las acciones que guían la solución propuesta.

2.1 IDENTIFICACIÓN DE LA AUDIENCIA

La identificación de la audiencia resulta un aspecto muy importante dentro del desarrollo de este trabajo, pues es necesario para determinar el usuario que va a interactuar finalmente con la multimedia, identificando sus necesidades para poder responder a éste de la mejor forma.

El software “Multimedia para el aprendizaje de Gestión de la Configuración” va dirigido principalmente a los estudiantes que cursan el tercer año en esta universidad, donde se imparte la asignatura de Ingeniería de Software, y el tema de Gestión de Configuración constituye uno de los Flujos de Trabajo definidos por la metodología RUP desarrollados en esta asignatura. Esta multimedia contribuye a mejorar la preparación de estos estudiantes. También está dirigido a las personas que imparten el curso de Gestión de Configuración del Perfil de Calidad de Software. En general esta aplicación está dirigida a todas las personas de la Universidad que participen en el proceso de desarrollo de Software y que por lo tanto necesiten capacitación sobre Gestión de Configuración para poder aplicar los procedimientos asociados a ese proceso.

2.2 ESPECIFICACIÓN DEL CONTENIDO

El contenido que se abordará en la multimedia está dividido en 3 temas fundamentales y cada uno de éstos en varios epígrafes.

- ✓ Tema 1: Conceptos Fundamentales

Este tema aborda los principales conceptos relacionados con la disciplina en cuestión para poder comprender bien todo el proceso de Gestión de Configuración de software. Está dividido en 3 epígrafes, tratando definiciones tales como: Línea Base; Gestión de Configuración de Software; Elemento de Configuración de Software; Control de Versiones; Control de Cambios, Repositorio, herramientas de soporte.

- ✓ Tema 2: Proceso de Gestión de Configuración de Software

En este tema se describe cómo se realiza el Proceso de Gestión de Configuración de Software, dividiéndolo en 7 epígrafes que constituyen las actividades que conforman este proceso: Identificación de la Configuración de Software; Administración de los procesos de GCS; Control de la Configuración de Software; Contabilidad del estado de la Configuración de Software; Auditoría de la Configuración de Software; Gestión del release del software y entrega.

- ✓ Tema 3: Herramientas utilizadas en el proceso de GCS

Este tema está compuesto por las herramientas que son usadas en el proceso de GCS, mostrando documentación sobre cada una de éstas, finalmente quedarían 6 epígrafes que describen las siguientes herramientas: ClearCase; SourceSafe; Subversion; Concurrent Versions System; Git; ClearQuest.

2.2.1 GUIÓN

La figura que se presenta a continuación, representa un guión donde se muestra en forma de nodos los contenidos que se abordan en la multimedia, para una mejor comprensión de cómo quedaría estructurado el contenido que se mostrará en ésta.

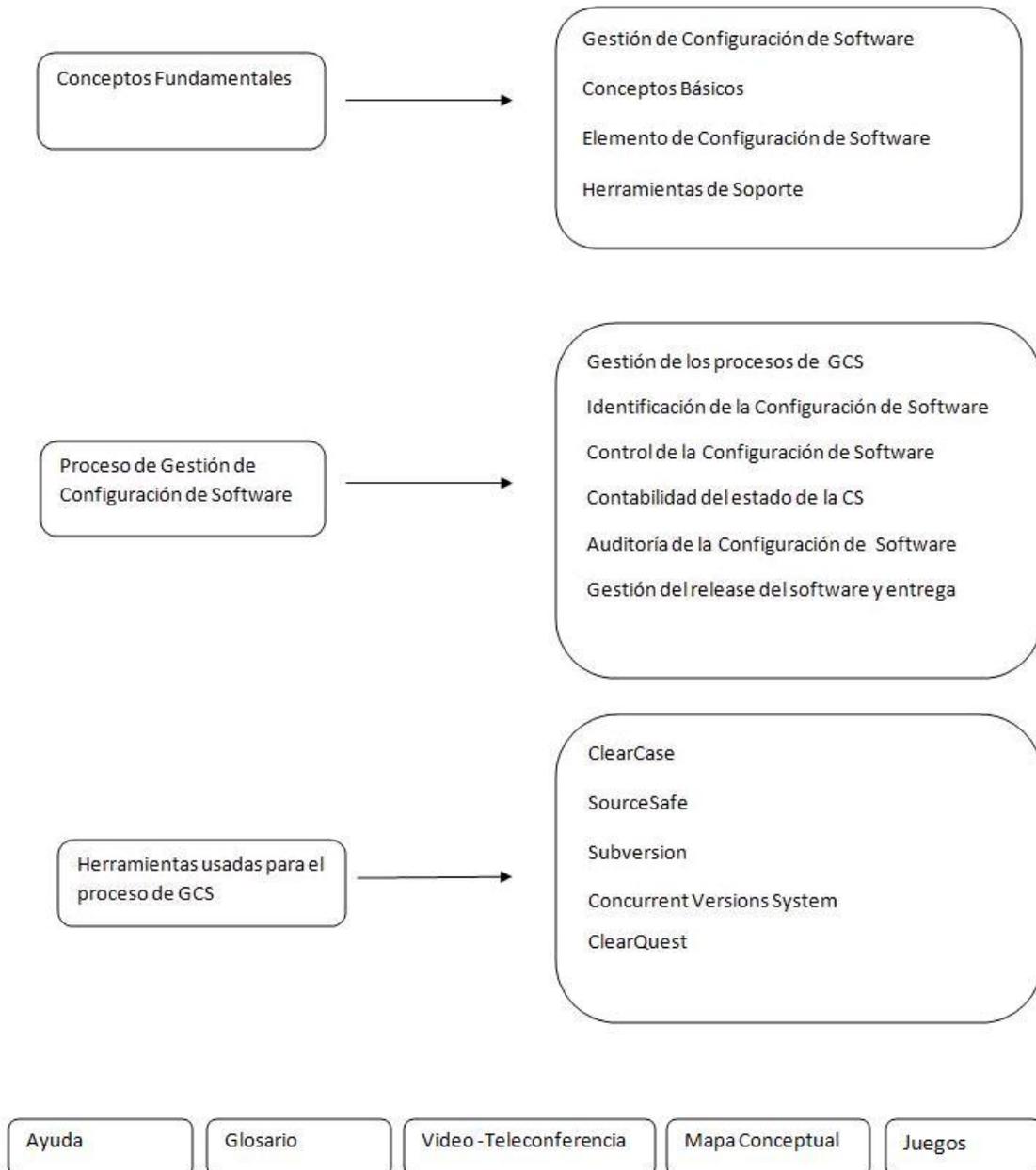


Fig. 1 Guión

2.2.2 MAPA CONCEPTUAL

Los mapas conceptuales son un medio para visualizar conceptos y las relaciones entre ellos. En el que se muestra en la fig. 2, se representan conceptos significativos que ayudan a comprender mejor el tema de Gestión de Configuración.

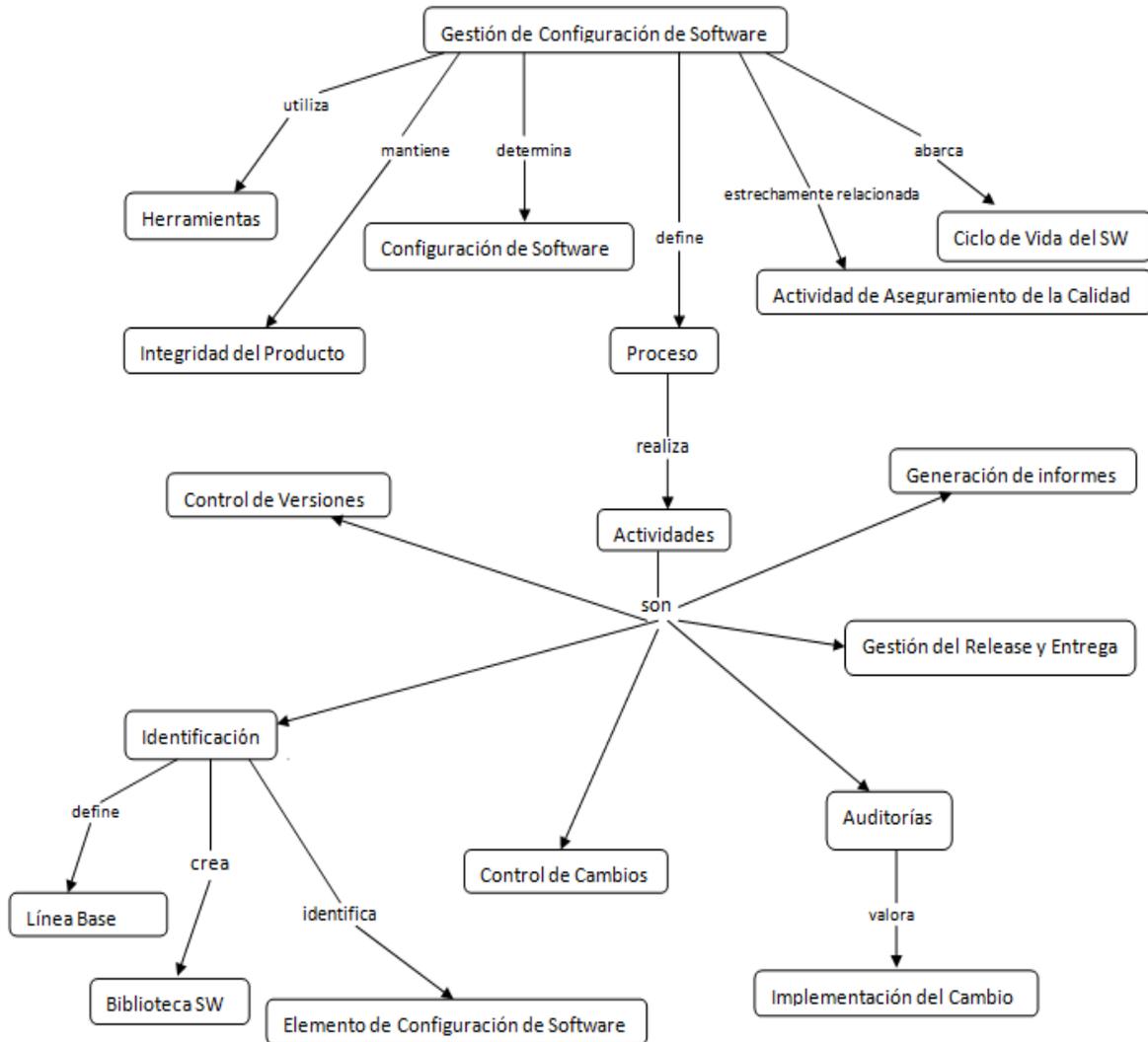


Fig. 2 Mapa Conceptual

2.3 MODELO DE DOMINIO

Mediante los conceptos más significativos que se definen en nuestro sistema, se pretende lograr una mejor comprensión del mismo al no estar bien definidos y estructurados los procesos de negocios; decidiéndose entonces realizar un modelo de dominio, donde se

definen las principales clases conceptuales que intervienen, representando los objetos significativos en el dominio del problema.

2.3.1 DIAGRAMA DE CLASES DEL MODELO DE DOMINIO

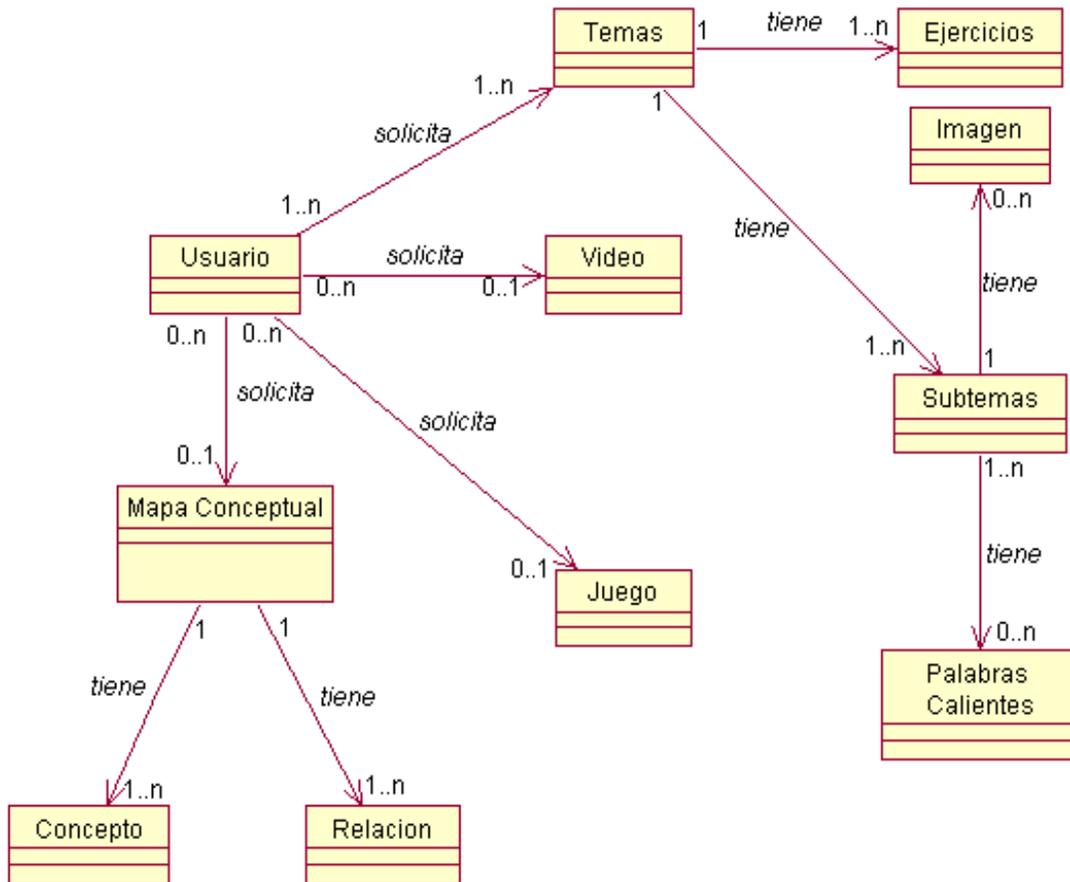


Fig. 3 Diagrama de Clases del Modelo del Dominio

2.3.2 ANÁLISIS DE LOS CONCEPTOS DE DOMINIO

A continuación se muestra el significado de los conceptos representados en el diagrama de clases del Modelo de Dominio.

Usuario: Es toda aquella persona que interactúe con el sistema.

Capítulo 2: Descripción de la Solución Propuesta

Temas: Distintos tópicos en los que está dividido el contenido, en este caso serían: Conceptos Fundamentales, Proceso de Gestión de Configuración de Software y Herramientas usadas en el proceso de GCS.

Ejercicios: Este objeto representa un conjunto de pequeñas pruebas para que el estudiante ejercite sus conocimientos y pueda tener un criterio sobre su aprendizaje en cada tema estudiado.

Subtemas: Distintos contenidos a estudiar en cada tema existente.

Imagen: Gráficos, figuras o dibujo utilizado en un epígrafe para aprender algún tema determinado.

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

Video: Representa una Video Conferencia sobre Gestión de Configuración de Software.

Juego: Representan objetos que ejercitan de una manera didáctica los conocimientos adquiridos durante el estudio del curso.

Mapa Conceptual: Representación estructural de conceptos mediante sus relaciones significativas.

Concepto: Significado de una determinada unidad de conocimiento dentro del mapa conceptual.

Relación: Tipo de relación existente entre un concepto y otro.

2.4 DESCRIPCIÓN DE LA FUNCIONALIDAD

Mediante la descripción de la Funcionalidad se especificará qué debe hacer el sistema a través de los requisitos Funcionales y No funcionales.

2.4.1 REQUISITOS FUNCIONALES

Los requisitos funcionales son capacidades o condiciones que el sistema debe cumplir. En la tabla 1 se muestra un listado de estos requisitos a los que dará respuesta el sistema de acuerdo a las necesidades del cliente.

Tabla 4. Requisitos Funcionales

Referencia	Función
R1	Mostrar presentación del producto.
R2	Permitir saltar la presentación del Producto.
R3	Mostrar pantalla principal del programa.
R4	Reproducir música de fondo automática de forma predeterminada en la pantalla principal.
R5	Permitir subir volumen, bajar volumen, parar y reanudar sonido de fondo.
R6	Permitir la navegación por el contenido del producto.
R7	Permitir que siempre estén visibles todos los temas contenidos en el producto.
R8	Mostrar los temas que conforman la aplicación.
R9	Mostrar menús desplegables al seleccionar un tema.
R10	Permitir escoger un tema seleccionado.
R11	Permitir al usuario seleccionar los subtemas existentes dentro de un tema.
R12	Mostrar información contenida dentro del subtema seleccionado por el usuario.
R13	Visualizar información referente a palabras calientes consultadas por el usuario.
R14	Permitir al usuario desplazar el scroll de texto para una visualización completa de la información en aquellas pantallas que requieren el uso del mismo.
R15	Permitir al usuario realizar ejercicios sobre los contenidos estudiados en cada tema.

R16	Mostrar mensaje de respuesta correcta e incorrecta
R17	Mostrar iconos de video conferencia, mapa conceptual y juego del curso.
R18	Permitir al usuario ejecutar el juego que se encuentra disponible.
R19	Permitir ver solución del juego.
R20	Visualizar video conferencia.
R21	Permitir subir volumen, bajar volumen, parar, reanudar, pausar, desplazar, ver siguiente, ver anterior, poner a pantalla completa y restablecer la pantalla del video.
R22	Visualizar Mapa Conceptual.
R23	Mostrar concepto asociado al nodo seleccionado por el usuario en el mapa conceptual.
R24	Permitir mover de lugar el mapa conceptual.
R25	Permitir ampliar tamaño de mapa conceptual.
R26	Mostrar Glosario de Términos.
R27	Mostrar Ayuda para el Producto.
R28	Permitir salir del sistema posterior a una confirmación.
R29	Mostrar créditos al salir de la aplicación.

2.4.2 REQUISITOS NO FUNCIONALES

Los requisitos no funcionales son propiedades o cualidades que el producto debe tener.

REQUISITOS DE HARDWARE

Los requisitos mínimos para la ejecución de la aplicación son:

- ✓ Procesador Pentium 3 a 800 MHz de velocidad de procesamiento o superior.

- ✓ 512 MB de espacio en disco duro disponible.

- ✓ 128 MB de RAM.

REQUISITOS DE SOFTWARE

La aplicación podrá ser ejecutada sobre cualquier sistema operativo, ya que para visualizar la multimedia solo es necesario tener instalado los plug-ins de Flash, y esto es posible en todo sistema operativo.

REQUISITOS DE APARIENCIA

- ✓ Utilizar botones que expresen su función ya sea con tooltip, que se intuya, o expresados con texto.
- ✓ El sistema debe brindar una interfaz amigable al usuario, siendo lo más agradable y cómoda posible.
- ✓ Todos los objetos interactivos del producto tendrán tooltips con el fin de mostrar su utilidad.
- ✓ El idioma de la aplicación será el español, con la excepción de las palabras que lo requieran por ser técnicas dentro del contexto donde aparezcan.
- ✓ El color de la letra debe estar en contraste con el fondo del área de texto para facilitar su lectura.
- ✓ La opción Salir debe estar disponible desde cualquier parte de la aplicación, con la excepción de la presentación.

REQUISITOS DE USABILIDAD

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

REQUISITOS DE DISEÑO E IMPLEMENTACIÓN

- ✓ Las herramientas a usar en el diseño gráfico e implementación de la aplicación serán: Adobe Flash CS3 Profesional, Adobe Photoshop Extended para el trabajo con las imágenes, Adobe Premiere Pro CS3 y Adobe Flash CS3 Video Encoder para la edición y codificación de los videos.
- ✓ El contenido será cargado desde archivos XML y el lenguaje de programación será ActionScript 2.0.
- ✓ Se utilizará la herramienta CASE Rational Rose Enterprise Edition 2003 para la modelación del sistema.

REQUISITOS DE SOPORTE

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

2.5 MODELO DE CASOS DE USO DEL SISTEMA

Un Modelo de Casos de Uso describe los requisitos funcionales de un actor en términos de las interacciones que éste ejecuta con el sistema. Presenta el sistema desde la perspectiva de su uso y esquematiza cómo proporcionará valor a sus usuarios. Cada caso de uso tiene una descripción que describe la funcionalidad que se construirá en el sistema propuesto.

2.5.1 DIAGRAMA DE CASOS DE USO DEL SISTEMA

La Fig. 5 muestra el Diagrama de casos de Uso del Sistema, donde se representa gráficamente el actor, los casos de usos y sus relaciones, identificados para dar solución al sistema propuesto.

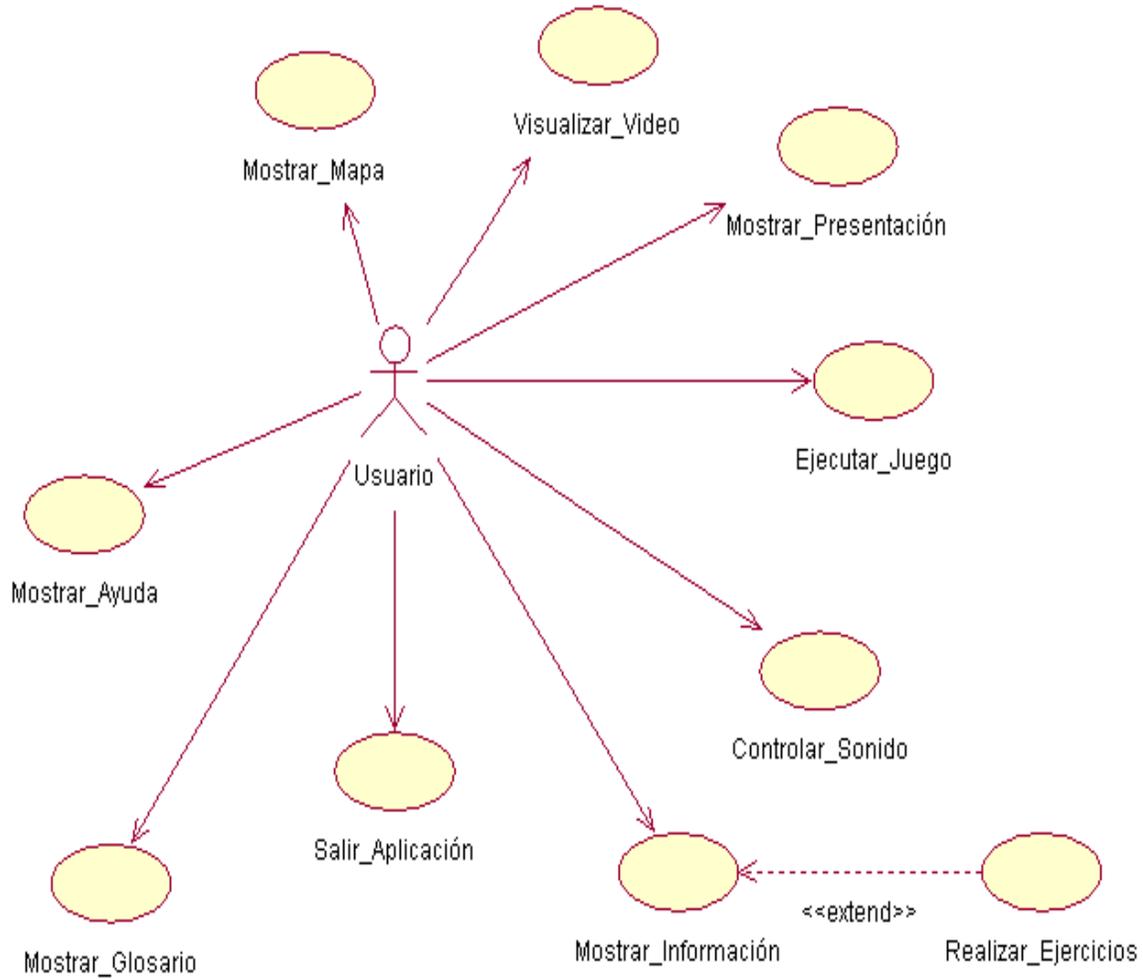


Fig. 4 Diagrama de Casos de Uso del Sistema.

2.5.2 DETERMINACIÓN Y JUSTIFICACIÓN DE LOS ACTORES DEL SISTEMA

Tabla 5. Justificación de los actores del sistema

Actor	Justificación
Usuario	Es la persona que va a interactuar con el sistema para buscar información sobre Gestión de Configuración de Software

2.5.3 DESCRIPCIÓN DE CASOS DE USO

Para lograr una mejor comprensión de los casos de uso del sistema, se realiza una breve descripción de cada uno de ellos mostrada en las tablas que aparecen a continuación.

Tabla 6. Mostrar Presentación

Caso de Uso:	Mostrar Presentación
Actor:	Usuario
Propósito:	Mostrar la presentación de la Multimedia
Resumen:	Al ejecutar la aplicación, el sistema carga la presentación del producto, siendo opcional su visualización, ya que el usuario puede pasar de forma directa a la pantalla principal del producto dando clic en el botón Saltar Presentación. Al concluir la presentación, se pasa de manera automática a la pantalla principal del producto.
Referencia:	R1, R2

Tabla 7. Mostrar Información

Caso de Uso:	Mostrar Información
Actor:	Usuario
Propósito:	Mostrar la información que solicite el usuario dentro de la aplicación, incluyendo mapa conceptual y ejercicios.
Resumen:	El caso de uso se inicia cuando el usuario selecciona la información que desea ver dentro de alguno de los subtemas incluidos en cada tema. También puede solicitar ver el significado de las palabras calientes que se encuentran en cada uno de éstos. El usuario hace clic sobre el botón perteneciente a la información deseada y el sistema se encargará de mostrar la misma.

Referencia:	R3, R6, R7, R14, R8, R9, R10, R11, R12, R13, R17
-------------	--

Tabla 8. Realizar Ejercicio

Caso de Uso:	Realizar Ejercicio
Actor:	Usuario
Propósito:	Permitir que el usuario ejercite cada uno de los temas estudiados y mida sus conocimientos.
Resumen:	El caso de uso se inicia cuando el usuario, después de haberse estudiado un tema seleccionado, presiona el botón referente a los ejercicios; visualizándose una pantalla en la cual podrá responder a las preguntas que se muestran del tipo Verdadero o Falso. Luego, cuando el usuario lo solicite, se le dará a conocer si la respuesta seleccionada es la correcta, y finalmente se le muestra las estadísticas, por ejemplo, cantidad de preguntas realizadas, la cantidad de respuestas correctas e incorrectas, finalizándose de esta forma el caso de uso.
Referencia:	R15, R16

Tabla 9. Mostrar Mapa

Caso de Uso:	Mostrar Mapa
Actor:	Usuario
Propósito:	Visualizar Mapa Conceptual que contiene los conceptos fundamentales del tema Gestión de Configuración de Software
Resumen:	El caso de uso se inicia cuando el usuario da clic sobre el botón del Mapa conceptual; visualizándose entonces una pantalla donde se muestra el mismo; dando la posibilidad de mostrar el concepto asociado al nodo seleccionado por el usuario mediante una ventana flotante y también mover de lugar el mapa conceptual, ampliar su tamaño.

Referencia:	R22, R23, R24, R25

Tabla 10. Controlar Sonido

Caso de Uso:	Controlar Sonido
Actor:	Usuario
Propósito:	Controlar la reproducción del sonido de fondo de la multimedia
Resumen:	Cuando se carga la pantalla principal de la multimedia, automáticamente comienza a reproducirse el sonido de fondo, en cualquier momento el usuario podrá controlar su reproducción, si desea subir y bajar el volumen, o parar y reanudar el sonido.
Referencia:	R4, R5

Tabla 11. Ejecutar Juego

Caso de Uso:	Ejecutar Juego
Actor:	Usuario
Propósito:	Ejecutar juego disponible en la aplicación
Resumen:	El caso de uso se inicia cuando el usuario seleccione el icono del juego contenido en el menú principal; mostrándose una pantalla donde se pueden ver un conjunto de preguntas a las que el usuario dará respuesta y de esta forma llenar el crucigrama donde, a partir de un conjunto de letras, en común se relacionan las palabras que darán respuestas a las preguntas dadas. El usuario podrá comprobar si su respuesta es correcta cada vez que entre una palabra; y además puede ver la solución del juego si lo solicita.
Referencia:	R18, R19

Tabla 12. Visualizar Video

Caso de Uso:	Visualizar Video
Actor:	Usuario
Propósito:	Visualizar la video- conferencia que contiene la multimedia
Resumen:	El usuario podrá visualizar la video-conferencia al dar clic sobre el botón que hace referencia a la misma; saliendo entonces una pantalla donde se muestra este video, y el usuario podrá parar, pausar, reproducir, subir o bajar el volumen del mismo.
Referencia:	R20, R21

Tabla 13. Mostrar Glosario

Caso de Uso:	Mostrar Glosario
Actor:	Usuario
Propósito:	Visualizar el Glosario de Términos
Resumen:	El caso de uso inicia cuando el usuario seleccione la opción referente al Glosario; mostrándose una pantalla donde hay un grupo de letras que constituyen el abecedario y, cuando el usuario seleccione la letra que desee, se visualizan los vocablos que comiencen con dicha letra, Las mismas coinciden con las palabras calientes, que serán las de difícil entendimiento.
Referencia:	R26

Tabla 14. Mostrar Ayuda

Caso de Uso:	Mostrar Ayuda
Actor:	Usuario
Propósito:	Brindarle al usuario una ayuda al utilizar el producto
Resumen:	El caso de uso se inicia cuando el usuario seleccione el icono de la Ayuda; entonces se le proporcionará una información general sobre la multimedia; además de la información relacionada con la navegación y las secciones que contiene la multimedia.
Referencia:	R27

Tabla 15. Salir Aplicación

Caso de Uso:	Salir Aplicación
Actor:	Usuario
Propósito:	Permitir salir de la aplicación.
Resumen:	El caso de uso se inicia cuando el usuario selecciona el botón salir ubicado en la parte superior derecha de la aplicación. Luego se visualiza una pantalla de confirmación. Si el usuario confirma se pasa a mostrar los créditos; en caso contrario, vuelve a la pantalla donde estaba anteriormente.
Referencia:	R28, R29

2.6 MAPA DE NAVEGACIÓN

El mapa de navegación es un diagrama que especifica cómo desplazarse dentro de los módulos de la aplicación; proporcionando una estructura del sistema compuesta por nodos; representando las diferentes pantallas de la aplicación. En la figura mostrada a continuación se presenta el Diagrama de Navegación General.

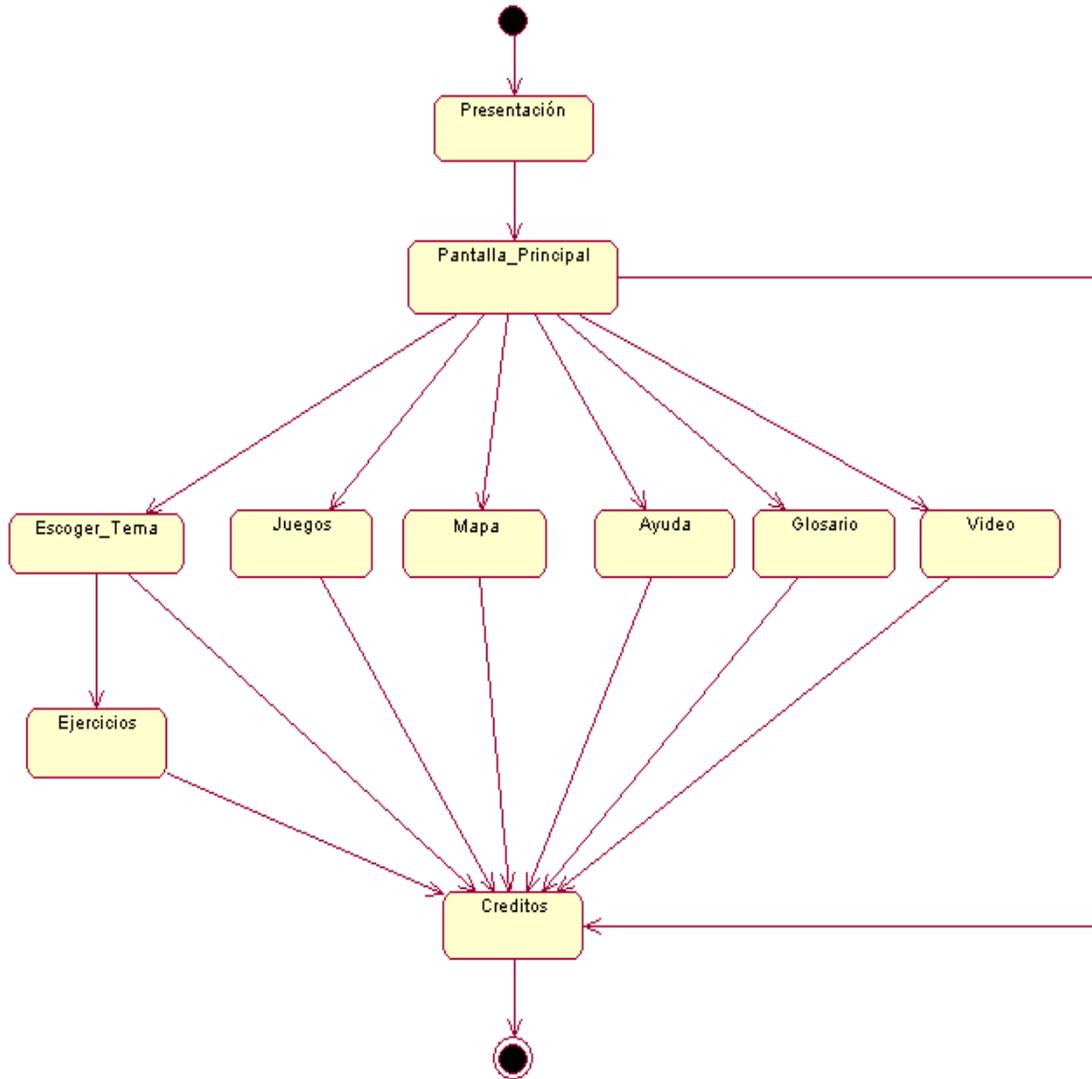


Fig. 5 Mapa de Navegación General

CAPÍTULO 3: “CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA”

INTRODUCCIÓN

En este capítulo se realiza la construcción de la solución propuesta, haciendo énfasis entonces en los flujos de trabajo del RUP de diseño e implementación. Como parte de éstos se realizan los diagramas de presentación de OMMMA-L, dando una panorámica de la visualización de las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área; y se realiza además el diagrama de componentes. Otros elementos importantes considerados, son los principios de diseño gráfico utilizados para la creación de las interfaces de la multimedia y la descripción de la estructura de los archivos XML usados para mostrar la información contenida en la multimedia de forma dinámica.

3.1 PRINCIPIOS DE DISEÑO

Actualmente las grandes y pequeñas compañías que desarrollan software multimedia, definen varias líneas de trabajo en dependencia del proyecto a desarrollar. Una de estas líneas de trabajo es el diseño gráfico del producto multimedia. Algunos de los parámetros que se analizan son: tipo de multimedia, si es de tipo educativa, de ocio, etc.; el contenido que almacenará, a quién va dirigida; entre otros, todo en función de crear un diseño que sea funcional, agradable en su uso, atractivo, dinámico, ameno, interesante y de fácil aprendizaje, pues la mejor interfaz de usuario exige que el esfuerzo de aprendizaje sea menor.

Para la creación de la interfaz gráfica es necesario seguir los siguientes principios de diseño:

- ✓ **Sistematicidad:** Consiste en mantener una uniformidad en su funcionamiento, para que el usuario memorice los aspectos generales con los que debe interactuar. Por ejemplo, que el modo de acceder a cada tema sea siempre el mismo. Lo mismo sucede con otros elementos de la interfaz como el sentido de la navegación, el color en algunas zonas, la estructuración de la pantalla, el uso de íconos y cursores y salida de las ventanas de diálogo.

- ✓ **Retroalimentación Funcional:** Es la respuesta del software a una acción ejecutada por el usuario como señal de haber entendido y procesado la necesidad de ese usuario. Si ésta no existe, puede causar frustración en el usuario y éste reduce la efectividad del producto. Es entonces que el diseño juega un papel importante al crear recursos gráficos visuales que hagan evidente la respuesta a cualquier acción del usuario.

Existen además otros puntos a tener en cuenta para lograr un buen diseño visual. Uno es el modo de organizar la información para percibirla como una forma en su conjunto y luego las partes. Esto teniendo en cuenta el contraste de formas por tamaño, por proximidad, las composiciones horizontales o verticales. Todo esto debe tributar a un menor esfuerzo de la vista para leer y de la mano para navegar.

De acuerdo a estos principios y elementos del diseño, finalmente se realiza el diseño de la multimedia predominando el color azul, pues es un color agradable a la vista, serio, se asocia al conocimiento y además es el color que identifica a nuestra universidad.

Con el objetivo de que haya mayor espacio para el contenido, se realiza un diseño con botones desplegables, mostrando los diferentes epígrafes que contienen la información que será visualizada, utilizando una tipografía Arial, caracterizada por poseer un espacio entre las letras que permite hacerla más apta para ser leída en una pantalla a varias resoluciones. El color de la letra está diseñado de manera tal que permita un buen contraste con respecto al color del fondo en el que se encuentre, ya sea una pantalla o un botón.

Las diferentes opciones que se pueden seleccionar además de los temas, se mantienen también siempre visibles y en el mismo lugar en cualquiera de las pantallas en la que se encuentra, posibilitando una mejor navegación dentro del producto.

3.2 DIAGRAMAS DE PRESENTACIÓN

Los diagramas de presentación son un nuevo artefacto dentro del lenguaje UML, incorporado a éste a partir de la extensión del mismo planteada por OMMMA-L. Estos son realizados a continuación para describir la distribución espacial de objetos visuales de la interfaz de usuario, divididos en objetos de visualización (texto, gráfico, video, animación)

e interacción (scrolls, barras de menú, botones, campos de entrada y salida, hipertextos con hipervínculos).

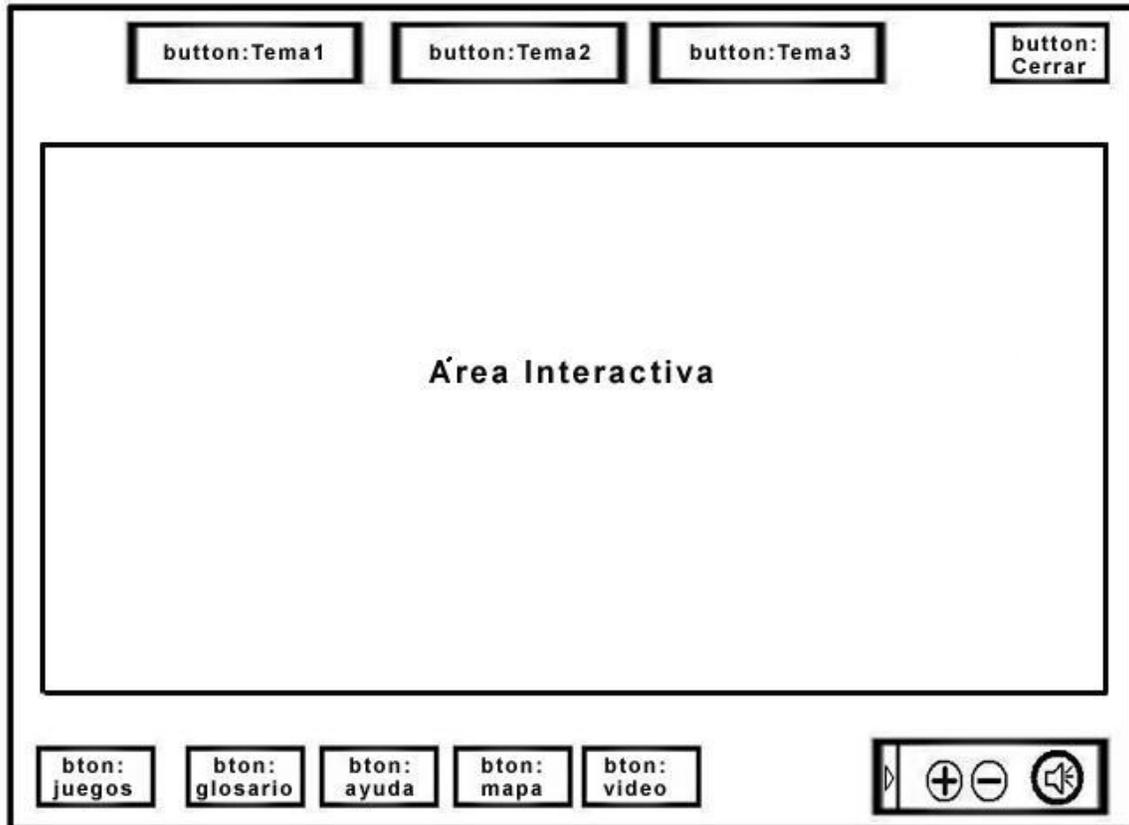


Fig. 6 Diagrama de Presentación Pantalla Principal

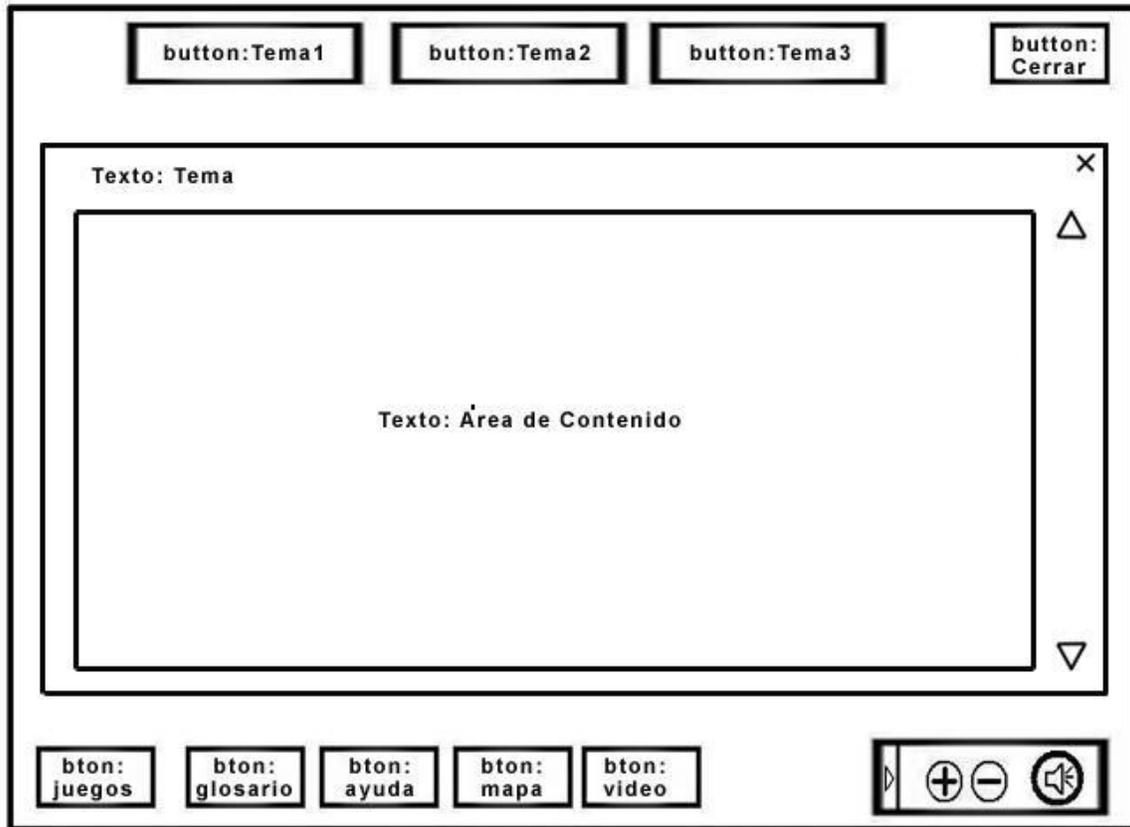


Fig. 7 Diagrama de Presentación Contenidos

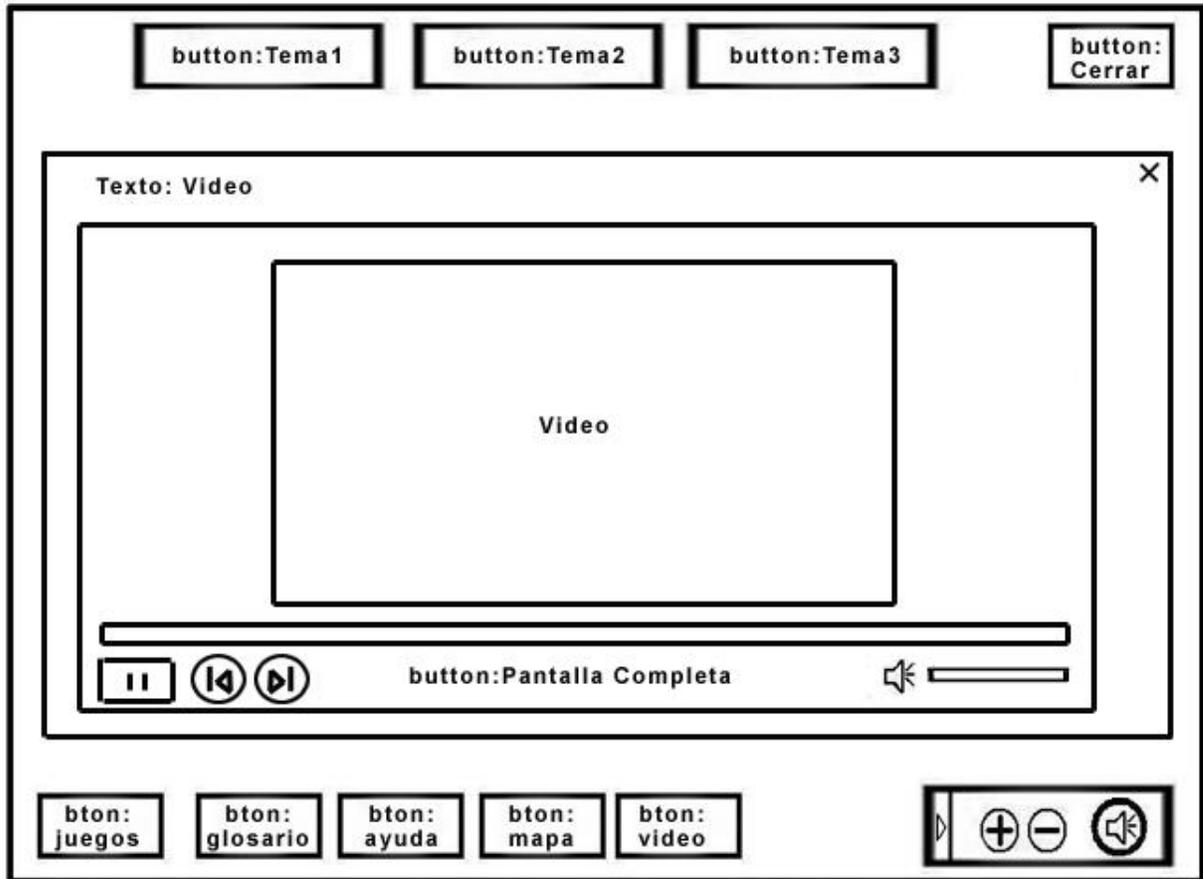


Fig. 8 Diagrama de Presentación Video

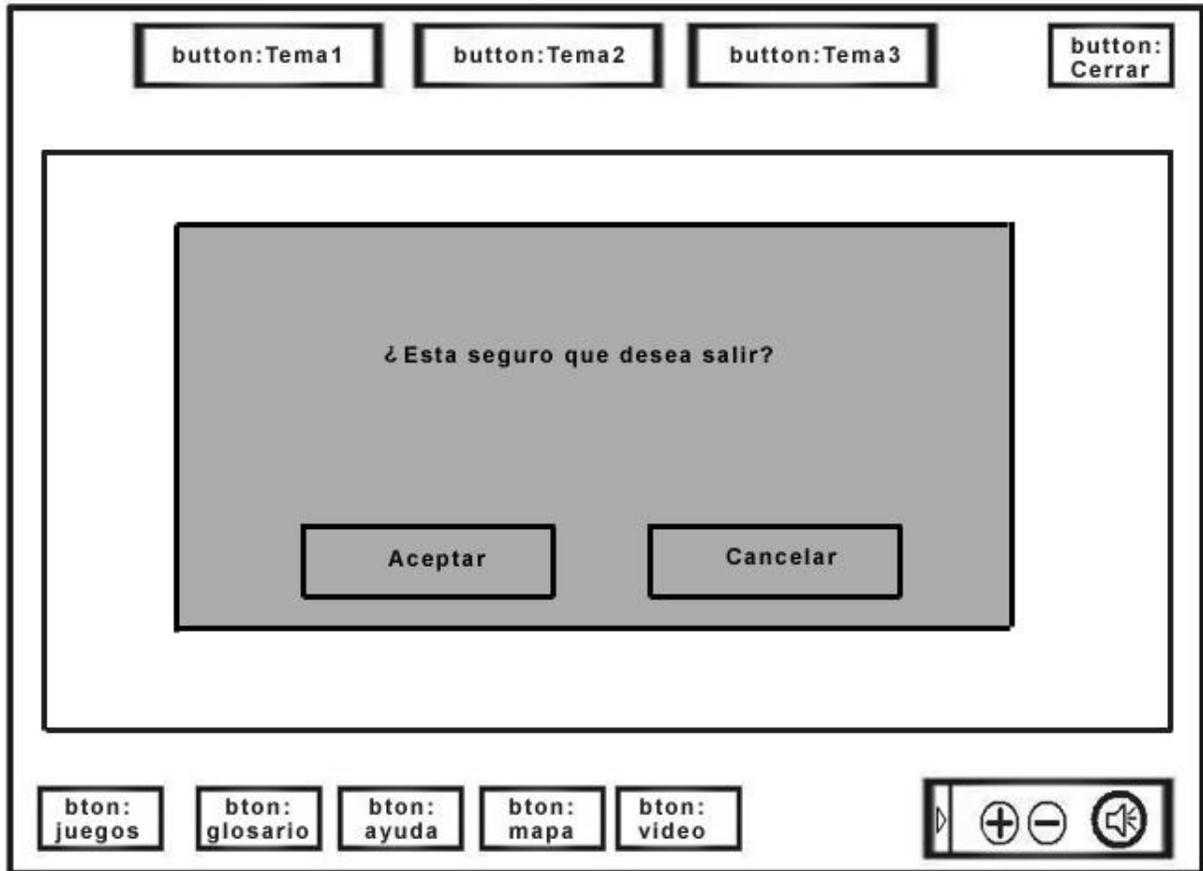


Fig. 9 Diagrama de Presentación Salir

3.3 DESCRIPCIÓN DE ARCHIVOS XML

Los archivos XML son usados en la multimedia para almacenar la información que contiene la aplicación y poder cargar esta información de forma dinámica en cualquier

momento; ofreciendo flexibilidad para organizarla y presentarla a partir de la creación de etiquetas personalizadas. A continuación se muestra una descripción de cada uno de los archivos XML utilizados en la aplicación.

DESCRIPCIÓN DEL ARCHIVO XML CONTENIDO

Para todos los temas que contiene la multimedia, se crea un archivo XML que posee la descripción detallada a continuación:

```
<contenido>

<texto_epígrafe1>Texto que pertenece al epígrafe 1 del tema</texto_epígrafe1>
.
.
.
.

<texto_epígrafe_n>Texto que pertenece al epígrafe n del
tema</texto_epígrafe_n>

</contenido>
```

<texto_epígrafe_n>: Este nodo indica el texto que está contenido en ese epígrafe, desde el 1 hasta el epígrafe n.

</texto_epígrafe_n>: Cierre del nodo.

DESCRIPCIÓN DEL ARCHIVO XML EJERCICIOS

```
<VerdaderoFalso>
```

```
  <cantidad cant="n"> Cantidad de preguntas </ cantidad >
```

```
  <pregunta1 estado=" False" o" True"> Pregunta </pregunta n >
```

```
  .
```

```
  .
```

```
  .
```

```
  <pregunta n estado=" False" o" True"> Pregunta </pregunta n >
```

```
</VerdaderoFalso>
```

<VerdaderoFalso>: Este nodo da inicio al XML con el contenido de la evaluación.

<VerdaderoFalso>: Cierre de nodo.

<cantidad cant="n">: Nodo que contiene el numero de preguntas.

</ cantidad >: Cierre del nodo

<pregunta n estado=" False" o" True"> Nodo que contiene la pregunta y el valor correspondiente, ya sea Verdadero o Falso.

</pregunta n >: Cierre del Nodo.

DESCRIPCIÓN DE ARCHIVO XML MAPA CONCEPTUAL

```
<MapaConceptual>

<concepto1>Texto que brinda el concepto seleccionado por el usuario</concepto1>
.
.
<concepto_n> Texto que brinda el concepto seleccionado por el usuario</concepto_n>

</MapaConceptual>
```

<MapaConceptual>: Dentro de este nodo estará todo el contenido que será visualizado en el mapa conceptual.

</MapaConceptual>: Cierre del tag <MapaConceptual>.

<concepto_n>: Este nodo contiene los diferentes conceptos que serán tratados en el Mapa Conceptual.

</concepto_n>: Cierre del tag <concepto_n>.

3.4 DIAGRAMA DE COMPONENTES

Con el resultado del diseño comenzamos a implementar el sistema en términos de componentes, es decir, scripts, ficheros de código fuente, ficheros de código binarios, ejecutables, etc.). Los diagramas de componentes son usados para modelar la vista estática del sistema; mostrando la organización y dependencia de un conjunto de componentes, ya sean librerías, código fuente, binarios o ejecutables.

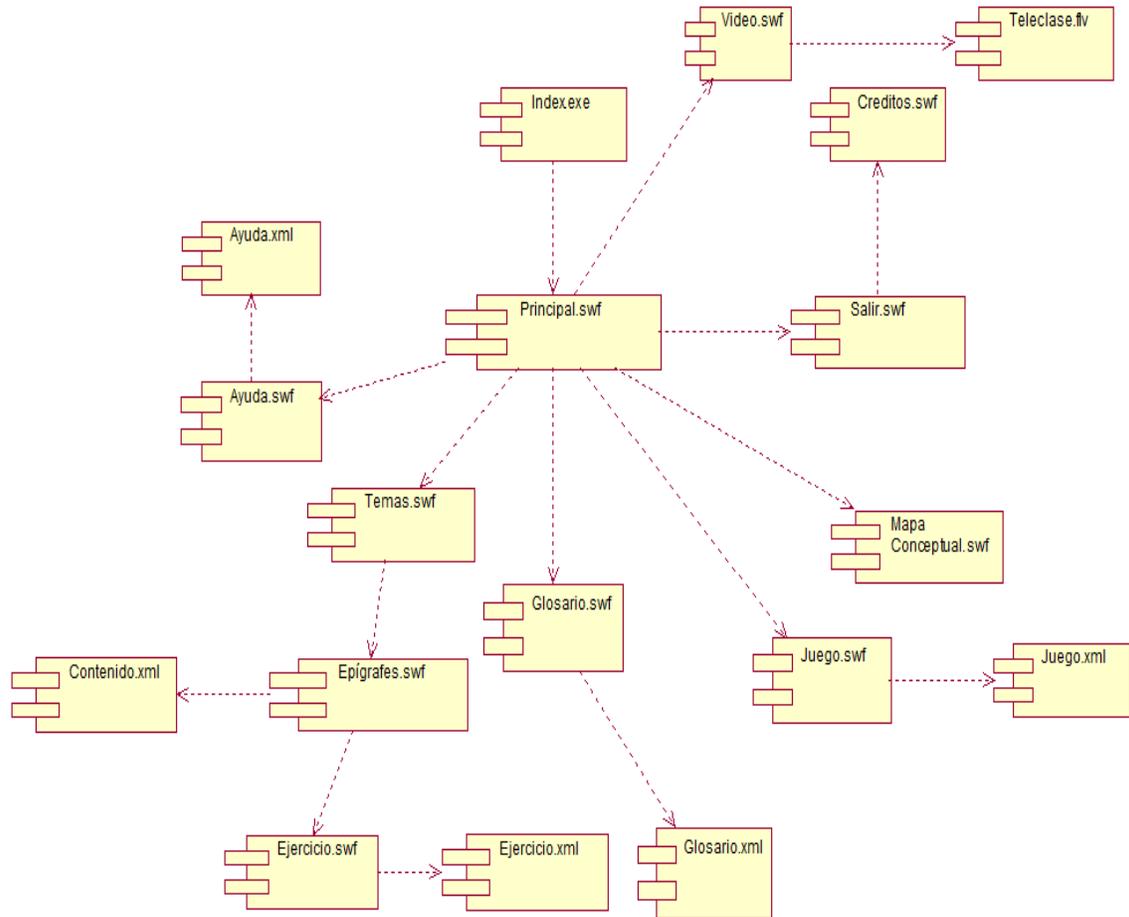


Fig. 10 Diagrama de Componentes

CONCLUSIONES

En el presente trabajo se implementó una multimedia para apoyar el proceso de enseñanza aprendizaje de Gestión de Configuración en la universidad, cumpliéndose el objetivo trazado luego de realizar toda la fundamentación teórica que sustenta la investigación; desarrollándose posteriormente el análisis, diseño e implementación del producto.

La aplicación multimedia se ha implementado utilizando la herramienta Adobe Flash CS3 Profesional. Como lenguaje de programación ActionScript 2.0 selección realizada después de hacer un estudio de las principales tendencias tecnológicas actuales en cuanto a herramientas usadas en este tipo de producto; así como la investigación para determinar en qué metodología de desarrollo de software basarnos. Concluyendo que lo más factible sería la metodología RUP y como lenguaje de modelado OMMMA-L, específico para el desarrollo de software de este tipo, siendo una extensión del lenguaje de modelado UML.

El software utiliza la tecnología XML para el almacenamiento de la información contenida en la multimedia, y se logró un producto con una interfaz amigable, de fácil uso, y funcional.

Se realizó un mapa conceptual que resume e interrelaciona los conceptos más importantes del tema.

La multimedia representa un material muy importante en la integración de conocimientos relacionados con la materia en cuestión, a través de animaciones, videos, etc. De una forma muy amena y atractiva se presenta todo el contenido que se muestra en ella.

RECOMENDACIONES

Se recomienda:

- ✓ Incorporar la multimedia como material complementario del curso de Ingeniería de Software II, en el tema de Gestión de Configuración.
- ✓ Enriquecer el contenido de cada uno de los temas tratados en la multimedia.
- ✓ Incorporarle más juegos y videos al producto.
- ✓ Agregar nuevos conceptos y relaciones al mapa conceptual para el enriquecimiento del mismo.

REFERENCIAS BIBLIOGRÁFICAS

[Adobe, 2008] Adobe. 2008. <http://www.adobe.com/products/>. [Online] 2008. [Cited: 29, 2008.]

[Febles, 2005] Ailyn Febles; Sofía Álvarez. 2005. La Gestión de Configuración y el Desarrollo de software en las Universidades. Una experiencia práctica. 2005.

[Adobe, 2008] Alicante. 2003. Metodologías Ágiles en el Desarrollo de Software. 1112, 2003.

[Barros, 2004] Barros, Diana Rodríguez. 2004. Conceptos generales sobre sistemas hipermediales. 228, 2004.

[Adobe, 2008] Birkdar, Nadia. 2003. <http://www.adobe.com/>. [Online] 22, 2003. [Cited: 49, 2009.] http://www.adobe.com/es/macromedia/proom/2003/01_03/dir_best.html.

[CICEI, 1995] CICEI. 1995. <http://www.ulpgc.es/>. [Online] 1995. [Cited: 103, 2009.] <http://www.ulpgc.es/otros/tutoriales/internet/tutor/hipetext.html>.

[Gaitero, 2008] Domingo, Gaitero. 2008. Gestión del ciclo de vida de los productos y servicios. Junio 2008.

[SOLENZAL y DÍAZ, 2006] Fabio Solenzal; Carmen Díaz. 2006. Autoaprende. ciudad de la Habana : s.n., 6 2006.

[Fernández, 2002] Fernández, Gerardo. 2002. Introducción a Extreme Programming. <http://flash.ciberaula.com/>. [Online] 2005. [Cited: 328, 12.] http://flash.ciberaula.com/articulo/actionscript_xml/.

[Booch, 2000] Ivar Jacobson; Grady Booch; James Rumbaugh. 2000. El proceso unificado de desarrollo de software(RUP). s.l. : sl: Pearson Educación S.A.

IEEE Std 610.12-1990 (R2002), IEEE Standard Glossary of Software Engineering Terminology, IEEE, 1990.

[Izakowitz, 1995] Izakowitz, Tomás. 1995. Methodology for structured hypermedia design. 1995.

[Menjivar, 2001] Menjivar, Eric. 2001. <http://www.maestrosdelweb.com>. [Online] 11 17, 2001. [Cited: 1 17, 2009.] <http://www.maestrosdelweb.com/editorial/multime/>.

[Merino, 2008] Merino, Vilma López. 2008. Herramientas profesionales para desarrollar productos multimedia. 2008.

[MINREX, 2005] MINREX. 2005. Programa sobre la informatización de la sociedad cubana. 4 20, 2005.

[Pressman, 2008] Pressman, Roger. 2001. *Ingeniería de del Software. Un enfoque práctico*. Quinta Edición, Q. 2001.

[Santos, 2008] Santos, Maria del Carmen. 2008. Prevención de las disgrafías escolares: una necesidad de la escuela actual, para la atención a la diversidad. 2008.

[Sherekan, 2008] Sherekan, Just. 2008. <http://sherekan.com.ar/>. [Online] 5 16, 2008. [Cited: 3 29, 2008.] <http://sherekan.com.ar/2008/05/16/introduccion-a-xml/>.

[SWEBOK, 2004] SWEBOK. 2004. Guide to the Software Engineering Body of Knowledge. 2004.

[Wikipedia, 2008] Wikipedia. 2008. <http://es.wikipedia.org/>. [Online] 11 28, 2008. [Cited: 2 7, 2008.] http://es.wikipedia.org/wiki/Herramienta_de_Autor.

BIBLIOGRAFÍA

Pressman, Roger. 2001. Ingeniería de del Software. Un enfoque práctico. Quinta Edicion, Q. 2001.

SEEK. Curriculum Guidelenes for Undergraduate Degree programs in Software Engineering . 2004.

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

Carmona, Enrique Linares. 2008. Multimedia para la comprensión del tema de procesos de Ingeniería de Requisitos. Ciudad Habana : s.n., 7 2008.

Zincke, Miguel Ángel del Pino. 2008. Multimedia Interactiva Pruebas de Software. 7 2008.

Adobe. 2008. <http://www.adobe.com/products/>. [Online] 2008.

ANEXOS



Fig. 11 Pantalla Principal

The screenshot shows a multimedia application window with a dark blue border. At the top, there are three navigation tabs: 'Conceptos Fundamentales', 'Proceso de Gestión de Configuración de Software', and 'Herramientas utilizadas en el proceso de GCS'. The main content area is titled 'Conceptos Básicos' and contains a definition of 'Línea Base' and a reference to a diagram. The diagram is a staircase-like structure representing the software development process, with the following phases and associated activities:

- Ingeniería del sistema**: Especificación del sistema
- Análisis de requisitos**: Especificación de requisitos del software
- Diseño del software**: Especificación de diseño
- Codificación**: Código fuente
- Prueba**: Planes de prueba

At the bottom of the window, there is a footer with the logo 'MGESCO' and the text 'Multimedia para el Aprendizaje de Gestión de la Configuración'. On the left side of the footer, there are several small icons representing different functions like search, help, and navigation.

Fig. 12 Pantalla Contenidos

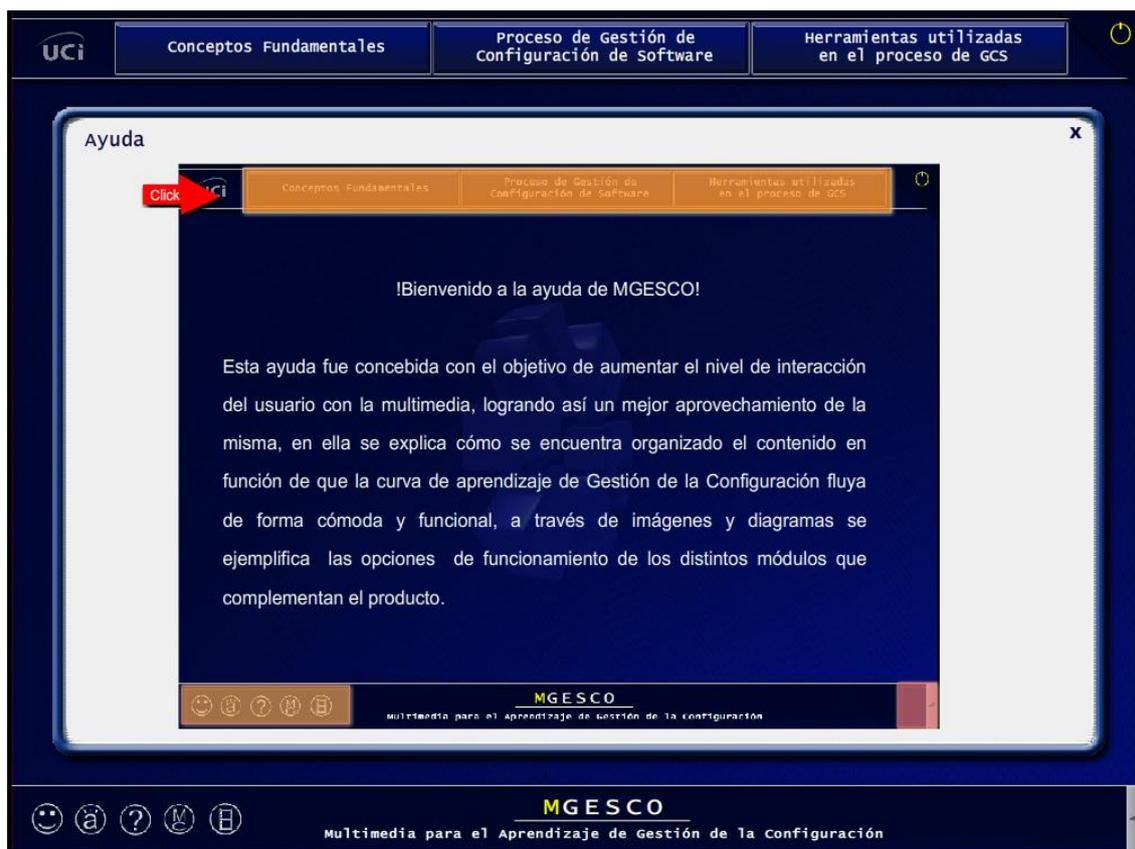


Fig. 13 Pantalla Ayuda

UCI

Conceptos Fundamentales

Proceso de Gestión de Configuración de Software

Herramientas utilizadas en el proceso de GCS

Glosario

A B C D E F G H I J K L M N Ñ O P Q R S T U V W X Y Z

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

Aseguramiento de la calidad: Todas las actividades planeadas y sistemáticas implementadas dentro del sistema de calidad y que se han demostrado que son necesarias para proporcionar la confianza adecuada de que una entidad satisface los requisitos de la calidad.

Atributo: Una característica de un elemento, por ejemplo, su color, tamaño o tipo. (2) Una propiedad física o abstracta mensurable de una entidad.

Auditoría de Software: La auditoría de software es una actividad que se realiza para evaluar independientemente la conformidad de productos de software y procesos con regulaciones, estándares, guías, planes y procedimientos.

MGESCO
Multimedia para el Aprendizaje de Gestión de la Configuración

Fig. 14 Pantalla Glosario

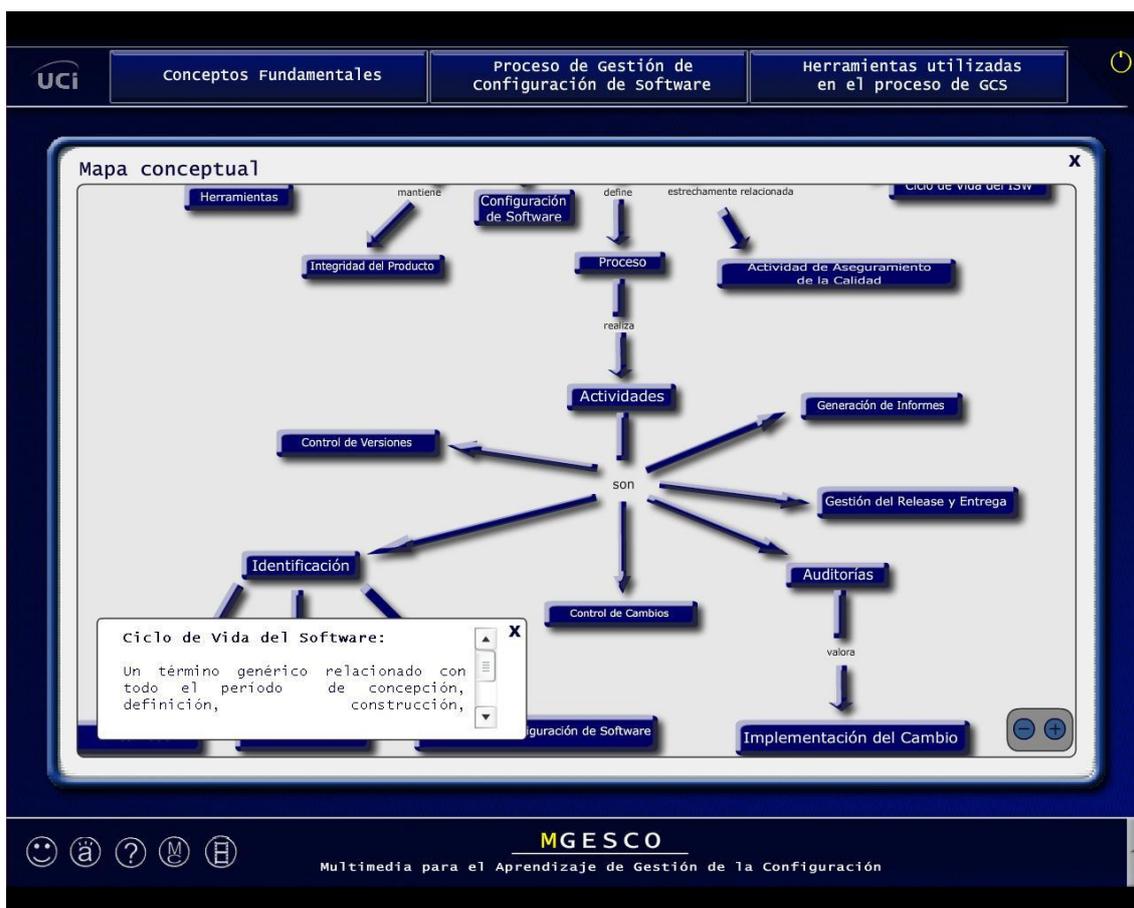


Fig. 15 Pantalla Mapa Conceptual

UCi Conceptos Fundamentales Proceso de Gestión de Configuración de Software Herramientas utilizadas en el proceso de GCS

Juego-Crucigrama

5 1 2 3 4 6

1- La Gestión de Configuración de Software es un __ de soporte del ciclo de vida del software.

2- Una Línea Base puede cambiarse solo a través de __ formales de control de cambios.

3- En un gran proyecto de software el __ incontrolado lleva rápidamente al caos.

4- Sitio centralizado donde se almacena y mantiene información.

5- Concepto de gestión de configuración de software que nos ayuda a controlar los cambios sin impedir seriamente los cambios justificados.

6- Gestión de Configuración de Software: __, cuya misión es controlar la evolución de un sistema software.

Comprobar Limpiar

MGESCO
Multimedia para el Aprendizaje de Gestión de la Configuración

Fig. 16 Pantalla Juego



Fig. 17 Pantalla Video



Fig. 18 Pantalla Salir

GLOSARIO DE TÉRMINOS

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

Aseguramiento de la calidad: Todas las actividades planeadas y sistemáticas implementadas dentro del sistema de calidad y que se han demostrado que son necesarias para proporcionar la confianza adecuada de que una entidad satisface los requisitos de la calidad.

Atributo: Una característica de un elemento, por ejemplo, su color, tamaño o tipo. (2)
Una propiedad física o abstracta mensurable de una entidad.

Auditoría de Software: La auditoría de software es una actividad que se realiza para evaluar independientemente la conformidad de productos de software y procesos con regulaciones, estándares, guías, planes y procedimientos.

Calidad de Software: La calidad es la suma de todos aquellos aspectos o características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades, expresadas o implícitas

Ciclo de Vida: Un término genérico relacionado con todo el período de concepción, definición, construcción, distribución, funcionamiento y disposición de un producto.

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

Firmware: La combinación de un dispositivo hardware y las instrucciones de computadora o datos de la computadora que residen como software de solo lectura sobre el dispositivo hardware.

Gestión: Disciplina del proceso de ingeniería de software, cuyo propósito es planificar y gestionar el proyecto de desarrollo.

Ingeniería de Software: La aplicación de un sistemático y disciplinado, cuantificables enfoque al desarrollo, operación y mantenimiento de software.

Interactivo: Relativo a un sistema o modo de operación en el que a cada entrada dada por el usuario provoca una respuesta ó acción por el sistema.

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

Medición: El acto o proceso de asignación de un número o una categoría a la entidad para describir un atributo de esa entidad.

Medida: Una manera de evaluar o determinar el valor por comparación a una norma.

Procedimiento: Manera especificada de realizar una actividad.

Proceso: Conjunto de actividades interrelacionadas, las cuales transforman entradas en salidas.

Proceso de desarrollo de Software: Un conjunto organizado de actividades realizadas para traducir las necesidades del usuario en un producto de software.

Producto Software: El conjunto de programas de computadora, procedimientos, y posible documentación y datos asociados.

Requisito: Condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

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

Sistema: Una combinación integrada, que consiste en uno o más procesos, hardware, software, facilidades y personas, que proporciona una capacidad de satisfacer una necesidad o un objetivo declarado.

Usuario: Un individuo u organización que utiliza el sistema operativo para ejecutar una función específica.