

Universidad de las Ciencias Informáticas

Facultad # 5



**Título: Multimedia Interactiva Pruebas de
Software.**

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

Autor: Miguel Ángel del Pino Zincke.

Tutor: MSc. Yamilis Fernández Pérez.

Julio 2008

DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo a la Facultad # 5 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

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

Miguel Ángel del Pino Zincke.

Autor.

Yamilis Fernández Pérez.

Tutor.

DATOS DE CONTACTO

Tutor: MSc. Yamilis Fernández Pérez

Correo electrónico: yamilisf@uci.cu

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

PENSAMIENTO

“Solo los necios se encuentran satisfechos y confiados con la calidad de su trabajo.”

Mercedes Milá.

AGRADECIMIENTOS

*A mis **padres**: Por todo el amor y el apoyo que siempre me han brindado, por su paciencia, dedicación, comprensión y por mantener siempre unida esta familia. Gracias por enseñarme el camino correcto, sin ustedes, este momento no hubiese sido posible.*

*A mi **tío**: Por ser el otro padre que siempre me aconseja, me ayuda, me regaña y por ser el profesor de clases de perfección que nunca tuve en ninguna enseñanza. A partir de hoy, vas a tener que buscar otro que te engrase las copillas y te alcance las llaves porque ya soy ingeniero.*

*A mi **abuela**: Por inculcarme la fe y enseñarme a creer que todo es posible, por esa eterna sonrisa y ese inmenso amor que siente por sus nietos. Este momento te lo dedico especialmente a ti.*

*A mi **hermano**: Por ser el más grande regalo que he recibido en la vida y por asumir con tanta decisión y valentía los malos momentos que nos ha impuesto la vida. Desde bien pequeño te convertiste en hombre.*

*A **Yarmay**: Por ser mi novia, mi amiga, mi madre, mi hermana, mi vida. Gracias por tanto amor y tanto apoyo durante tantos años. Espero que todos nuestros sueños se hagan realidad.*

A mi tía Lizzie: Por todo el amor y el apoyo que siempre nos ha brindado. Todos te queremos muchísimo y nunca dejamos de pensar en ti.

A mis socios, amigos y hermanos: No voy a mencionar a ninguno porque todos fueron y son muy importantes para mí. Ustedes son el recuerdo más lindo que me llevo de esta universidad, a todos los quiero y todos forman parte de mi corazón.

A mi tutora, profesores y todos aquellos que hicieron posible que llegara este momento.

DEDICATORIA

A mis padres por todo lo que significan para mí y por su incansable apoyo a lo largo de todos estos años, a mi abuela por inculcarme la fe y enseñarme a creer en mí, a mi tío por ser el profesor de clases de perfección que nunca tuve en ninguna enseñanza, a mi hermano por ser el regalo más grande de mi vida y a Yarmay por ser la mujer que amo....

RESUMEN

La aparición de las Nuevas Tecnologías de la Información y las Comunicaciones (NTIC) ha marcado un cambio significativo en la sociedad con respecto a la información. Las nuevas posibilidades en el acceso, almacenamiento y utilización de la información han motivado una toma de conciencia sobre el valor de la información en el ámbito educacional, dando origen a la implementación de nuevas formas de representar el conocimiento por medios no tradicionales; a raíz de esta tendencia, la Universidad de las Ciencias Informáticas (UCI), desde sus inicios, ha estado a la vanguardia en la creación de sistemas multimedia destinados al apoyo del proceso de enseñanza. En el presente trabajo de diploma se propone una multimedia interactiva para ser utilizada como material de apoyo en el proceso de enseñanza aprendizaje de pruebas de software en la UCI. Para el desarrollo de la misma, se realizó un análisis detallado de las diferentes herramientas existentes para la creación de aplicaciones multimedia, estimándose Macromedia Flash 8 como la más apropiada para la realización del producto. Para el desarrollo de dicho producto se utilizó la metodología de desarrollo RUP, UML en su extensión OMMMA - L como lenguaje representativo, ActionScript como lenguaje de implementación y XML como lenguaje para gestionar y agrupar los datos en volúmenes compactos de información. La multimedia interactiva Pruebas de Software puede significar un excelente material de apoyo al curso optativo que se imparte en la UCI sobre este tema, que por su interactividad, resulta un material de estudio más atractivo y ameno que aquellos a los que los estudiantes tienen alcance hasta este momento.

PALABRAS CLAVE:

Pruebas de software, multimedia, multimedia interactiva, producto multimedia, herramienta, metodología, lenguaje y diagrama.

ÍNDICE

AGRADECIMIENTOS	I
DEDICATORIA	III
RESUMEN	IV
ÍNDICE DE FIGURAS	VIII
ÍNDICE DE TABLAS	VIII
INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	7
INTRODUCCIÓN	7
1.1 DEFINICIÓN DE CALIDAD	7
1.1.1 DEFINICIÓN DE CALIDAD DE SOFTWARE.....	8
1.2 CONTROL DE LA CALIDAD DE SOFTWARE	9
1.3 PRUEBAS	9
1.3.1 CONCEPTOS DE PRUEBAS.....	9
1.3.2 OBJETIVOS DE LAS PRUEBAS.....	10
1.3.3 PRINCIPIOS DE LAS PRUEBAS.....	11
1.3.4 CARACTERÍSTICAS DE LAS PRUEBAS.....	12
1.4 ENSEÑANZA DE PRUEBAS DE SOFTWARE EN LA ACTUALIDAD	13
1.5 SOFTWARE EDUCATIVO	14
1.5.1 CARACTERÍSTICAS ESENCIALES DE LOS PROGRAMAS EDUCATIVOS.....	15
1.6 MULTIMEDIA	16
1.6.1 HIPERTEXTO, MULTIMEDIA E HIPERMEDIA.....	16
1.6.2 DEFINICIÓN DE MULTIMEDIA.....	17
1.6.3 MULTIMEDIA INTERACTIVA.....	18
1.6.4 DEFINICIÓN DE HIPERTEXTO.....	19
1.6.5 DEFINICIÓN DE HIPERMEDIA.....	20
1.6.6 CLASIFICACIÓN DE MULTIMEDIA.....	20

1.6.7 APORTES DE LA MULTIMEDIA AL PROCESO DE ENSEÑANZA APRENDIZAJE.....	21
1.7 TENDENCIAS TECNOLÓGICAS ACTUALES.....	22
1.7.1 HERRAMIENTAS PARA LA CREACIÓN DE SOFTWARE MULTIMEDIA.....	23
1.7.2 HERRAMIENTA A UTILIZAR.....	26
1.7.3 METODOLOGÍAS A CONSIDERAR PARA EL DESARROLLO DEL SOFTWARE.....	28
1.7.4. LENGUAJES DE MODELADO.....	31
1.7.5 FUNDAMENTACIÓN DE LA METODOLOGÍA A UTILIZAR.....	34
1.7.6 LENGUAJES UTILIZADOS EN LA MULTIMEDIA.....	35
1.8 ANÁLISIS CRÍTICO DE LAS HERRAMIENTAS DE ENSEÑANZA EXISTENTES.....	37
CONCLUSIONES.....	38
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN.....	40
INTRODUCCIÓN.....	40
2.1 IDENTIFICACIÓN DE LA AUDIENCIA.....	40
2.2 ESPECIFICACIÓN DEL CONTENIDO.....	41
2.2.1 MAPA CONCEPTUAL.....	42
2.3 DESCRIPCIÓN DEL MODELO DE DOMINIO.....	43
2.3.1 DIAGRAMA DE CLASES DEL MODELO DE DOMINIO.....	44
2.3.2 ANÁLISIS DE LOS CONCEPTOS DE DOMINIO.....	44
2.4 DIAGRAMAS DE NAVEGACIÓN.....	45
2.4.1 MODELO DE NAVEGACIÓN GENERAL.....	46
2.4.1 SUB MODELO DE NAVEGACIÓN TEMAS.....	47
2.5 DESCRIPCIÓN DE LA FUNCIONALIDAD.....	48
2.5.1 REQUISITOS FUNCIONALES.....	48
2.5.2 REQUISITOS NO FUNCIONALES.....	49
2.5.3 REQUISITOS DE APARIENCIA.....	50
2.5.4 REQUISITOS DE USABILIDAD.....	50
2.5.5 REQUISITOS DE SOFTWARE.....	50
2.5.6 REQUISITOS DE HARDWARE.....	51
2.5.7 REQUISITOS DE DISEÑO E IMPLEMENTACIÓN.....	51

2.5.8 REQUISITOS DE SOPORTE.	51
2.6 MODELO DE CASOS DE USO DEL SISTEMA.	52
2.6.1 DETERMINACIÓN Y JUSTIFICACIÓN DE LOS ACTORES DEL SISTEMA.....	52
2.6.2 DIAGRAMA DE CASOS DE USO.	52
2.6.3 BREVE RESUMEN DE LOS CASOS DE USO DEL SISTEMA.	53
CONCLUSIONES	58
CAPÍTULO 3: SOLUCIÓN DE LA DESCRIPCIÓN PROPUESTA.....	59
INTRODUCCIÓN	59
3.1 ANÁLISIS DE LA ARQUITECTURA DE LA INFORMACIÓN UTILIZADA.....	59
3.1.1 PRINCIPIOS Y NORMAS DE DISEÑO.	59
3.1.2 ESTÁNDARES PARA EL DISEÑO DE LA INTERFAZ DEL USUARIO MULTIMEDIA.	61
3.1.3 ESTÁNDARES DE CODIFICACIÓN.	62
3.2 DESCRIPCIÓN DE ARCHIVOS XML.	63
3.3 DIAGRAMAS DE PRESENTACIÓN DEL MODELO DE ANÁLISIS.....	67
3.4 DIAGRAMA DE COMPONENTES.	73
3.5 DIAGRAMA DE DESPLIEGUE.	73
CONCLUSIONES	75
CONCLUSIONES GENERALES	76
RECOMENDACIONES	78
REFERENCIAS BIBLIOGRÁFICAS	79
BIBLIOGRAFÍA	84
GLOSARIO DE TÉRMINOS	85
ANEXOS	87

ÍNDICE DE FIGURAS

FIG. 1 MAPA CONCEPTUAL PRUEBAS DE SOFTWARE..... 43

FIG. 2 MODELO DE DOMINIO 44

FIG. 3 DIAGRAMA DE NAVEGACIÓN GENERAL 46

FIG. 4 DIAGRAMA DE NAVEGACIÓN MÓDULO TEMA. 47

FIG. 5 DIAGRAMA DE CASOS DE USO DEL SISTEMA. 53

FIG. 6 DIAGRAMA DE PRESENTACIÓN PANTALLA ESCOGER TEMA 68

FIG. 7 DIAGRAMA DE PRESENTACIÓN TEMA..... 69

FIG. 8 DIAGRAMA DE PRESENTACIÓN JUEGOS 70

FIG. 9 DIAGRAMA DE PRESENTACIÓN AUTOEVALUACIÓN..... 71

FIG. 10 DIAGRAMA PRESENTACIÓN PANTALLA SALIR..... 72

FIG. 11 DIAGRAMA DE PRESENTACIÓN VIDEOS 73

FIG. 12 DIAGRAMA DE COMPONENTES GENERALES 74

FIG. 13 MODELO DE DESPLIEGUE 75

ÍNDICE DE TABLAS

TABLA 1. CALIFICACIONES PROMEDIO..... 21

TABLA 2. REQUISITOS FUNCIONALES..... 48

TABLA 3. SISTEMAS OPERATIVOS Y NAVEGADORES..... 50

TABLA 4. ACTOR DEL SISTEMA. 52

TABLA 5. CU MOSTRAR PRESENTACIÓN..... 54

TABLA 6. CU MOSTRAR INFORMACIÓN. 54

TABLA 7. CU MOSTRAR VIDEO. 55

TABLA 8. CU REALIZAR AUTOEVALUACIÓN..... 55

TABLA 9. CU EJECUTAR JUEGO. 56

TABLA 10. CU VISUALIZAR MAPA. 56

TABLA 11. CU MOSTRAR MATERIALES COMPLEMENTARIOS..... 57

TABLA 12. CU SALIR DE LA APLICACIÓN..... 57

TABLA 13. CU MOSTRAR CRÉDITOS 58

Introducción

El software se ha convertido en un tema crítico en la sociedad moderna mundial. Todos parecen necesitar mejores software en el menor tiempo posible y a menor costo. Los métodos intuitivos de desarrollo de software que se usan actualmente son, básicamente, aquellos que los propios individuos “artesanalmente” siguen, los cuales sólo servirán mientras la sociedad pueda tolerar la falta de predicción que ellos acarrearán, es por ello que se hace cada vez más imponente el tema de la calidad de los productos de software que se producen en todo el mundo.

La industria del software en el mundo se desarrolla a un ritmo vertiginoso, aunque la producción sigue siendo aún baja y los costos muy elevados. Esta situación se debe, en la mayoría de los casos, a la no aplicación de técnicas de Ingeniería y Gestión de Software y la no definición de roles y procesos adecuados en el desarrollo de software. (FEBLES, 2004)

A nivel mundial la industria del software es una esfera muy inmadura debido a los frecuentes fracasos en los proyectos de desarrollo. Desde el año 1994, Standish Group realiza investigaciones que son publicadas anualmente donde se muestran las cifras de los proyectos rechazados, aceptados y cancelados.

Según una publicación realizada por la revista SD Time en el año 2006 a partir de investigaciones realizadas por Standish Group se puede observar como en la actualidad existen grandes problemas en la creación de software con la calidad requerida (GONZÁLEZ, 2008). El nuevo informe develado por SD Time muestra que en el 2006, sólo el 35 por ciento de los proyectos que se comenzaron se consideran exitosos cumpliendo con el tiempo requerido, el presupuesto y los requisitos exigidos por el cliente. A pesar de mostrarse una marcada mejora con respecto al año 1994 cuando solo el 16.2 por ciento fue exitoso, todavía no se llega a niveles que hagan de la producción de software una industria segura y de éxito. El 46 por ciento de los proyectos comenzados fueron rechazados o no aceptados por los clientes (en 1994, 52.7 por ciento) y el 19 por ciento nunca se terminaron o fueron cancelados (31.1 por ciento en 1994). (RUBINSTEIN, 2007)

Según esta publicación existe “menos caos” si se comparan las cifras actuales con las del año 1994, pero la realidad es que todavía son muy pocos los proyectos que se desarrollan exitosamente a nivel mundial comparados con la cantidad que se inician anualmente.

En la industria del software en Cuba existen problemas con la productividad de los trabajadores, los tiempos de entrega de los productos y las documentaciones, la calidad de las pruebas y la falta de comunicación efectiva entre los usuarios, desarrolladores, administradores, clientes e investigadores. (FEBLES, 2001)

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

La Universidad de las Ciencias Informáticas (UCI) es pionera en la creación de un grupo central dedicado al diseño y aplicación de pruebas de software para validar la calidad de los productos desarrollados, pero la misma no está exenta de los problemas existentes mencionados anteriormente.

Las pruebas de software son la actividad más comúnmente realizada para el control de calidad en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas. El aseguramiento de calidad incluye otras técnicas como, inspecciones y revisiones (automatizadas o no) de modelos y documentos no ejecutables de las primeras fases de desarrollo. Las pruebas de software se definen como *“una actividad en la cual un sistema o uno de sus componentes se ejecuta en circunstancias previamente especificadas, los resultados se observan y registran, y se realiza una evaluación de algún aspecto”*. (IEEE., 1990)

El desarrollo de una estrategia de prueba de software adecuada incluye la integración de las técnicas de diseño de casos de prueba en una serie de pasos bien planificados que dan como resultado una correcta construcción del software. Esta debe incorporar: la planificación de las pruebas, el diseño de los casos de prueba, la ejecución de las pruebas, la recolección y evaluación de resultados.

Con el objetivo de conocer la situación real del proceso de prueba en la UCI (LUGO, 2007) aplicó una encuesta a una muestra de los Grupos de Calidad que realizan el Proceso de Prueba de Liberación de Software para los productos que se exportan. (GONZÁLEZ, 2008)

Algunos de los problemas detectados se muestran a continuación: (LUGO, 2007)

- ✓ El volumen de datos e información manejado en las pruebas y la consulta de muchos especialistas no son registrados correctamente.
- ✓ No se almacena la experiencia ni la información de un proceso de prueba a otro, provocando incurrir en los mismos errores.
- ✓ Se hace engorrosa la gestión de personal y en particular su selección teniendo en cuenta que la mayoría es personal docente.
- ✓ Capacitar el personal de pruebas utilizando los sistemas de formación convencionales basados en cursos presenciales, unido a la indefinición de roles y actividades concretas presenta los siguientes inconvenientes:
 - No se cuenta con profesores capacitados en la materia o disponibles en el momento requerido, lo que afecta considerablemente los cronogramas.
 - El volumen de productos y de las versiones de estos obliga a una formación continua y reciclado periódico de las personas y el conocimiento de estas en el negocio que implementa cada producto.

Como ha sido planteado anteriormente las pruebas de software dentro del proceso de desarrollo del mismo constituyen en eslabón fundamental en el propósito de crear productos con la calidad requerida. Para que un producto software tenga la calidad requerida se necesita de personal suficientemente capacitado en el tema de calidad de software, ya que la calidad final del producto es la que permitirá que el software sea comercializado con garantías.

En la UCI surge la imperiosa necesidad de mejorar la capacitación de los estudiantes que participan en el proceso de prueba para elevar la calidad de los productos que se desarrollan en la misma por lo que es necesario organizar la información que se encuentra en libros, folletos, plantillas o en forma de experiencia en aquellas personas que llevan un tiempo considerable trabajando en la práctica.

Teniendo en cuenta todo lo descrito anteriormente se puede identificar como **problema científico** la siguiente interrogante:

¿Cómo contribuir a mejorar la capacitación de estudiantes que participan en el proceso de pruebas de software en la UCI?

Partiendo del problema científico el **Objeto de Estudio** de este trabajo es:

El proceso de desarrollo de una multimedia interactiva.

Enmarcando el **Campo de Acción** en:

El proceso de desarrollo de una multimedia interactiva que apoye el proceso de enseñanza aprendizaje de pruebas de software.

En correspondencia con el problema científico planteado el **Objetivo General** de este trabajo es el siguiente:

Implementar una multimedia interactiva que apoye el proceso de enseñanza aprendizaje de pruebas de software en la UCI.

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

- ✓ Estudiar aspectos importantes vinculados a la calidad del software.
- ✓ Recopilar y analizar toda la información necesaria sobre pruebas de software para la elaboración del contenido del producto a desarrollar.
- ✓ Confeccionar un mapa conceptual que contenga los conceptos necesarios para la comprensión del proceso de pruebas.
- ✓ Análisis crítico de 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).
- ✓ Análisis crítico de diferentes herramientas para la creación de aplicaciones multimedia.

- ✓ Investigar sobre 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.
- ✓ Realizar el análisis, diseño e implementación de una multimedia que contenga un mapa conceptual y diversos medios (videos, juegos y autoevaluaciones) que ayuden a la comprensión del tema, utilizando la herramienta y el lenguaje de modelado seleccionado.

Para cumplimentar el objetivo y dar respuesta al problema científico planteado se aplicaron los siguientes métodos teóricos: analítico-sintético para el procesamiento de la información recopilada durante la investigación; inductivo deductivo, para extraer regularidades y elaborar conclusiones de la tesis e histórico lógico para constatar la evolución histórica de la situación problemática planteada y la necesidad de darle solución. Como método empírico se entrevistó a profesores y especialistas, específicamente a aquellos que participan en el proceso de pruebas de software en la UCI.

Como novedad científica en la realización de este trabajo se agregará en el producto a realizar un mapa conceptual, por la importancia que tienen los mismos en el apoyo al proceso de enseñanza aprendizaje, que permite, de manera simplificada y haciendo uso de conceptos relacionados entre si, mostrar al usuario una panorámica de lo que estudiará en el curso.

Finalmente para darle solución al problema planteado, se procederá a la elaboración de un producto multimedia denominado “Multimedia Interactiva Pruebas de Software” que apoye el proceso de enseñanza aprendizaje de pruebas de software para los estudiantes de la Universidad de las Ciencias Informáticas.

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

Capítulo 1: “Fundamentación teórica.”

Resume todos los conceptos vinculados al objeto de estudio con el fin de lograr una mejor comprensión del problema al que nos enfrentamos. Se describen además los lenguajes de programación y sistemas para realizar el análisis y diseño de la aplicación así como las tendencias y tecnologías actuales sobre las que se basa.

Capítulo 2: “Descripción de la solución propuesta”

En este capítulo se realiza un análisis del producto Multimedia Interactiva Pruebas de Software, definiendo el dominio de la aplicación, mapa de navegación y el levantamiento de los requerimientos funcionales y no funcionales. También se comprende dentro de este capítulo el modelado del sistema, donde se realiza una descripción de los casos de uso correspondientes.

Capítulo 3: “Solución de la descripción propuesta.”

Este capítulo está dedicado principalmente a la construcción del sistema a desarrollar, incluyendo los diagramas de presentación. También se aborda el modelo de implementación, el cual incluye el diagrama de componentes y el diagrama de despliegue.

Capítulo 1: FUNDAMENTACIÓN TEÓRICA.

Introducción

El uso de las Nuevas Tecnologías de la Información y la Comunicación (NTIC) en la educación, es un fenómeno complejo, de amplias perspectivas y cuyos resultados serán más favorables a largo plazo, en la medida en que la respuesta a la pregunta ¿Cómo utilizar la tecnología ante cada tipo de situación educativa?, esté clara para todos los que de una manera u otra intervienen en el área de la tecnología para la educación, y sean consecuentes con ella. (LABAÑINO, 2005)

Se hace necesaria una renovación en la concepción del proceso de enseñanza aprendizaje tradicional, es por esto que la utilización de los sistemas multimedia en la educación responde a una nueva concepción de la enseñanza, ya que desde un punto de vista educativo ofrece una red de conocimiento interconectado que permite al estudiante moverse por rutas o itinerarios no secuenciales y, de este modo potenciar el desarrollo intelectual y la formación de valores.

El presente capítulo tiene como objetivo exponer los fundamentos teóricos generales que sirven de punto de partida para entender qué significa calidad de software, qué son las pruebas de software, qué es una multimedia, cuales son las metodologías más utilizadas y las herramientas existentes para la creación de las mismas. Además se identificará la herramienta que se utilizó y se expondrán las razones por la cual fue escogida la misma así como los lenguajes que se utilizaron para la creación de la aplicación.

1.1 Definición de Calidad.

El concepto de calidad ha evolucionado en el tiempo y en dependencia de la profesión de la persona que la estudie y la utilice como herramienta en la gerencia de las industrias. Se puede decir que la calidad es sinónimo de eficiencia, flexibilidad, corrección, confiabilidad, mantenibilidad, portabilidad, usabilidad, seguridad e integridad.

Se puede definir calidad como “Una característica o atributo de algo”. La calidad de un producto es medible a través de estándares como longitud, color, propiedades eléctricas y maleabilidad, entre otros.

Según la ISO 8402 se define calidad como "el conjunto de características de una entidad que le confieren la aptitud para satisfacer necesidades establecidas o implícitas" (GRANJA, 2007). Esta definición considera como una entidad no solamente el producto o servicio que se vende, sino también una persona, una organización, un sistema, etc.

De forma general la calidad de un proceso o servicio radica en la satisfacción de las necesidades del usuario/cliente, que teóricamente, se han solicitado.

1.1.1 Definición de calidad de software.

El software se ha convertido hoy en día en uno de los principales objetivos estratégicos de las organizaciones, debido a que cada vez más, los procesos principales de las organizaciones dependen de los sistemas informáticos para su buen funcionamiento, convirtiéndose así la calidad del software en un aspecto importante en la sociedad actual. A continuación se exponen un conjunto de conceptos y definiciones para una mejor comprensión del tema.

Según Pressman, la calidad de software es “la concordancia con los requisitos funcionales y de rendimiento explícitamente establecidos, con los estándares de desarrollo explícitamente documentados y con las características implícitas que se esperan de todo software desarrollado profesionalmente”. (PRESSMAN, 2002)

La definición más utilizada es la brindada por el IEEE: “La calidad del software es el grado con el que un sistema, componente o proceso cumple los requerimientos especificados y las necesidades o expectativas del cliente o usuario”. (FEBLES, 2006)

Para el autor calidad de software puede resumirse como: conjunto de cualidades que caracterizan al software y que determinan su utilidad y existencia.

1.2 Control de la Calidad de Software.

El Control de Calidad es una actividad que se tiene que llevar a cabo para garantizar la calidad de todo producto o servicio que se ofrece a los consumidores. Según Pressman el control de la calidad “es una serie de inspecciones, revisiones y pruebas utilizados a lo largo del proceso del software para asegurar que cada producto cumple con los requisitos que le han sido asignados.” (PRESSMAN, 2002)

Las actividades del control de calidad pueden ser manuales, completamente automáticas o una combinación de herramientas automáticas e interacción humana. (PRESSMAN, 2002)

La calidad de un sistema software es algo subjetivo que depende del contexto y del objetivo que se pretenda conseguir. Para determinar dicho nivel de calidad se deben efectuar una serie de pruebas que permitan comprobar el grado de cumplimiento respecto de las especificaciones iniciales del sistema.

1.3 Pruebas.

Las pruebas de software son un elemento crítico para la garantía de la calidad del software y representa una revisión final de las especificaciones, del diseño y la codificación. La creciente percepción del software como un elemento del sistema y la importancia de los costes asociados a un fallo del propio sistema están motivando la creación de pruebas minuciosas y bien planificadas por lo que no es raro que una organización dedicada al desarrollo de software emplee entre el 30 y 40 por ciento del esfuerzo total de un proyecto en las pruebas. En casos extremos, las pruebas del software para actividades críticas (por ejemplo control de tráfico aéreo, control de reactores nucleares) pueden costar de 3 a 5 veces más que el resto de los pasos de la ingeniería de software juntos. (PRESSMAN, 2002)

1.3.1 Conceptos de pruebas.

Las pruebas de software constituyen actividades fundamentales en muchos procesos de desarrollo, incluyendo el del software.

Según (IEEE., 1990) el concepto de prueba se define como: *“Una actividad en la cual un sistema o componente es ejecutado bajo condiciones específicas, se observan o almacenan los resultados y se realiza una evaluación de algún aspecto del sistema o componente”*.

Y según Pressman *“la prueba del software es un elemento crítico para la garantía de calidad del software y representa una revisión de las especificaciones, del diseño y de la codificación”*. (PRESSMAN, 2002)

De forma general se puede concluir que las pruebas de software pueden ser definidas como: proceso de ejercitar un programa con la intención específica de encontrar errores previos a la entrega al usuario final. La prueba de software es la actividad más común de control de calidad realizada en los proyectos de desarrollo o mantenimiento de aplicaciones y sistemas.

Las pruebas de software no garantizan que un software esté libre de errores, sino que se detecten la mayor cantidad de defectos posibles en el mismo para su debida corrección. Si al ejecutar las pruebas no se encuentran errores esto indica que las pruebas realizadas no son lo suficientemente eficientes.

1.3.2 Objetivos de las pruebas.

Los objetivos fundamentales de las pruebas según Myers en (PRESSMAN, 2002) son:

- ✓ La prueba es un proceso de ejecución de un programa con la intención de descubrir un error.
- ✓ Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error no descubierto hasta entonces.
- ✓ Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Según Pressman el objetivo de la prueba es: *“diseñar pruebas que saquen a la luz diferentes clases de errores con la menor cantidad de tiempo y espacio.”* (PRESSMAN, 2002)

Con estos elementos se define cual es el objetivo fundamental de la prueba de software. A continuación se enuncian algunas actividades que pueden considerarse como parte de los objetivos generales de las pruebas:

- ✓ Encontrar y documentar defectos en la calidad de software.

- ✓ Advertir sobre la calidad percibida del software.
- ✓ Validar y probar las especificaciones hechas en el diseño y especificación de requerimientos a través de una demostración concreta. (RUP, 2003)

De forma general se infiere que el objetivo general de las pruebas es encontrar errores y defectos que atenten contra la calidad final del producto en construcción o en funcionamiento.

1.3.3 Principios de las pruebas.

Para una aplicación adecuada de los diferentes métodos de pruebas que existen se hace necesario que los ingenieros que trabajan en esta línea conozcan una serie de principios que los ayudarán y guiarán en la realización del diseño de casos de prueba que se van a efectuar.

Según (PRESSMAN, 2002)

- ✓ *A todas las pruebas se le debería poder hacer un seguimiento hasta los requisitos del cliente.* Como se ha visto el principal objetivo es encontrar errores. Para el cliente los errores más graves son los que le impiden al sistema cumplir sus requisitos.
- ✓ *Las pruebas deberían planificarse mucho antes de que empiecen.* La planificación de las pruebas puede comenzar tan pronto como esté completo el modelo de requisitos. La definición detallada de los casos de prueba puede empezar una vez que se haya aprobado el modelo de diseño. Por tanto, se pueden planificar y diseñar todas las pruebas antes de generar ningún código.
- ✓ *El principio de Pareto es aplicable a la prueba del software.* El principio de Pareto implica que el 80 por ciento de todos los errores descubiertos durante las pruebas surgen al hacer un seguimiento de sólo el 20 por ciento de todos los módulos del programa. El problema está en aislar estos módulos sospechosos y probarlos.
- ✓ *Las pruebas deberían empezar por lo pequeño y progresar hacia lo más grande.* Las primeras pruebas planeadas y ejecutadas en general se centran en módulos individuales del programa y a medida que avanzan las pruebas, se concentran en encontrar errores en grupos integrados de módulos y finalmente al sistema entero.

- ✓ *No son posibles las pruebas exhaustivas.* Esto se plantea porque incluso en un programa pequeño la cantidad de permutaciones de caminos es muy grande, por lo que es imposible cubrir todas las combinaciones de caminos. Sin embargo es posible cubrir adecuadamente la lógica del programa y asegurarse de que se han aplicado todas las condiciones del diseño procedimental.
- ✓ *Para hacer más eficaces, las pruebas deberían ser realizadas por un equipo independiente.* Se ha mostrado que el ingeniero de software que creó el sistema no es el más indicado para realizar las pruebas al sistema.

A estos principios se pueden agregar otros no menos importantes como:

- ✓ La prueba puede ser usada para mostrar la presencia de errores, pero nunca de su ausencia.
- ✓ La principal dificultad del proceso de prueba es decidir cuando parar.
- ✓ Los casos de pruebas tienen que ser escritos no sólo para condiciones de entrada válida y esperada sino también para condiciones no válidas e inesperadas.
- ✓ Los casos de pruebas tienen que ser escritos para generar las condiciones de salida deseadas.
- ✓ El número de errores sin descubrir es directamente proporcional al número de errores descubiertos.

De forma general se puede concluir que el principio fundamental es que la prueba no puede asegurar la ausencia de defectos; solo puede demostrar que existen defectos en el software.

1.3.4 Características de las pruebas.

Hay ciertas características que son vitales para que una prueba sea calificada como idónea, entre ellas se pueden citar las siguientes:

1. Ha de tener una alta probabilidad de encontrar un fallo. Cuanto más, mejor.
2. No debe ser redundante. Si ya funciona, no se prueba más.
3. Debe ser la “mejor de la cosecha”. Si se tiene de donde elegir, se elige la mejor.

4. No debería ser ni demasiado sencilla, ni demasiado compleja. Si es muy sencilla no aporta nada, si es muy compleja a lo mejor no se conoce lo que ha fallado. (PRESSMAN, 2002)

1.4 Enseñanza de pruebas de software en la actualidad.

Cada día el proceso de desarrollo de software y el control de la calidad están más ligados. No se concibe en la actualidad que una empresa dedicada al desarrollo de software no realice un estricto control de la calidad del producto que están construyendo. Los modelos y estándares de calidad deben ser aplicados desde el mismo comienzo del proceso de desarrollo del producto así como verificaciones y validaciones constantes para asegurar que el producto en construcción cumpla con los requisitos y necesidades expuestos por el cliente para la satisfacción del mismo.

Por esta razón la preparación del personal dedicado al aseguramiento de la calidad del software debe ser cada vez mayor. En países como España, en algunas universidades como la Universidad Politécnica de Madrid las asignaturas de calidad se imparten como cursos optativos que no pertenecen al plan de estudio de dichos centros. Esto trae como consecuencia que cuando determinadas empresas buscan personal de calidad de software los resultados no son igual de satisfactorios al que pudiera existir con otras materias como programación en Java.

Si embargo existen universidades que si tienen estas asignaturas como parte del plan de estudio del curso (Universitat Oberta de Catalunya) y han decidido incluirlo en la asignatura obligatoria de Ingeniería de Software preparando un personal mucho más competente que el resto de las universidades que no imparten estos temas de manera obligatoria por ser considerada una materia “teórica y abstracta”.

En cuanto al tema específico de enseñanza de pruebas de software existe en Barcelona, España, un laboratorio independiente especializado en servicios de ingeniería de calidad y pruebas de software denominado inQA.labs (INQALABS, 2007) dedicado a la definición de procesos de Ingeniería de Software y a la verificación y validación de productos software a lo largo de su ciclo de vida. Esta empresa ofrece una nueva gama de cursos y material de enseñanza sobre test y Calidad de Software (SQA). La compañía imparte una serie de cursos, catalogados en distintos niveles algunos certificados por

organismos internacionales reconocidos (Information Systems Examination Board) dirigidos a técnicos y responsables de sistemas, directores de proyectos y usuarios avanzados, es decir, personal con experiencia en la materia que desea ampliar sus conocimientos en la especialidad. Existen muchas otras instituciones y empresas que se especializan en pruebas de software y ofrecen servicios y herramientas para la verificación y validación de productos de software como por ejemplo el Laboratorio Nacional de Calidad de Software, en España. (INTECO, 2008)

Las tendencias pedagógicas actuales en la enseñanza de pruebas de software como se ha mencionado antes son mediante cursos optativos de pregrado, cursos de postgrado o mediante asignaturas incluidas en el plan de estudios de diferentes instituciones o universidades en sus carreras vinculadas a Ciencias Informáticas. Estos cursos o asignaturas pueden ser impartidos de forma presencial, utilizando el método convencional de enseñanza u online con la ayuda de Internet o Entornos Virtuales de Aprendizaje (EVA). Universidades como la Biblioteca Nacional de Maestros en Argentina (BNM, 2008) y la Universidad Distrital Francisco José de Caldas (UDISTRAL, 2008) ofrecen cursos online sobre pruebas de software.

En la UCI la enseñanza de pruebas se incluye dentro de la asignatura Ingeniería de Software 2, específicamente en el Flujo de Trabajo Pruebas de RUP. Los estudiantes tienen la posibilidad de recibir de forma teórica todo lo que respecta a pruebas de software y después aplicar los conocimientos recibidos de forma práctica en los proyectos que van realizando a medida que el curso avanza. Esta asignatura además de ser impartida en aulas se encuentra disponible en el EVA de la universidad. Mediante técnicas transmisoras fundamentalmente, los estudiantes pueden acceder a este curso a través de la red para el estudio de la asignatura y realizar pruebas online con fines autoevaluativos o exámenes con criterios de evaluación para los profesores que imparten dicha asignatura.

1.5 Software educativo.

Entre los años 1950 y 1960 se empezaron a construir los primeros software con un enfoque lineal y entre los años 1960 y 1970 se caracterizaron por la forma de modelos abiertos, marcados por el uso de computadoras para tareas de práctica y ejercitación, en los cuales las computadoras deberían ayudar en los procesos de enseñanza aprendizaje basados en algunos modelos mayormente matemáticos.

En los últimos años las tecnologías de la información y la comunicación (TIC) han incrementado su importancia, causando importantes cambios en el entorno. Por ello resulta interesante reflexionar acerca de las ventajas, y oportunidades brindadas por estas nuevas tecnologías en la docencia. La revisión realizada permitió acceder a obtener definiciones de varios autores sobre software educativo. Se define al Software Educativo como: cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. En un concepto más restringido se define Software Educativo como aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender. (SÁNCHEZ, 1999)

De forma general para el autor el software educativo puede definirse como programas para computadora destinados a la enseñanza y el autoaprendizaje que permiten el desarrollo de habilidades cognitivas convirtiendo la computadora común, de una máquina de propósito general a una máquina para fines educativos.

1.5.1 Características esenciales de los programas educativos.

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

- ✓ Son materiales elaborados con una **finalidad didáctica**, como se desprende de la definición.
- ✓ **Utilizan el ordenador** como soporte en el que los alumnos realizan las actividades que ellos proponen.
- ✓ **Son interactivos**, contestan inmediatamente las acciones de los estudiantes y permiten un diálogo y un intercambio de informaciones entre el ordenador y los estudiantes.
- ✓ **Individualizan el trabajo** de los estudiantes, ya que se adaptan al ritmo de trabajo de cada uno y pueden adaptar sus actividades según las actuaciones de los alumnos.

- ✓ **Son fáciles de usar.** Los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son similares a los conocimientos de electrónica necesarios para usar un vídeo, es decir, son mínimos, aunque cada programa tiene unas reglas de funcionamiento que es necesario conocer.

Un ejemplo de software educativo son las llamadas Multimedia. Un ejemplo de multimedia es Microsoft Encarta, la cual a través de imágenes con abundantes colores, sonidos y mapas de navegación permite realizar un viaje virtual a través de la historia.

1.6 Multimedia.

La educación ha evolucionado a lo largo de los años, desde su forma meramente práctica en la época antigua, para posteriormente pasar a integrar nuevas técnicas y teorías. Gracias a los avances tecnológicos, se ha ido integrando la computadora para ser utilizada en el proceso de enseñanza aprendizaje. Lo anterior ha exigido el desarrollo de nuevos software, hasta alcanzar la generación de sistemas complejos de multimedia que sientan las bases para la tecnología de la Realidad Virtual. En estos últimos tiempos se puede observar cómo diversos materiales multimedia son utilizados en todos los ámbitos educativos en las diversas modalidades de enseñanza. La multimedia es un sistema tecnológico que al ser aplicado al sistema educativo puede ocasionar toda una innovación sorprendente con respecto a la forma tradicional de la educación. La tecnología multimedia ya sea utilizada en las clases presenciales o a distancia, es fuente importante de información de las cuales pueden surgir diferentes líneas de investigación.

1.6.1 Hipertexto, Multimedia e Hipermedia.

Con el término multimedia aparecen varios conceptos asociados al mismo y que son de vital importancia para su entendimiento, como son hipertexto, hipermedia y multimedia, conceptos muy relacionados y que hoy en día se confunden entre sí. Sin embargo, se puede decir que Hipermedia es un concepto que combina los términos de **Hipertexto** y **Multimedia**.

1.6.2 Definición de Multimedia.

Desde el comienzo de los años 90 el término multimedia es muy utilizado y está relacionado con la informática, las telecomunicaciones, la edición de los documentos, la electrónica de consumo, el cine, la televisión, etc.

Para iniciar con un claro concepto lo que es Multimedia (MM), resulta importante conocer que esta se define de forma general como “muchos medios”.

Una definición más específica sería: Un sistema que es capaz de presentar una combinación de información textual, sonora y audiovisual coordinando gráficos, fotos, secuencias animadas de video, gráficos animados, sonidos y voces, permitiendo en ocasiones una interacción multisensorial del usuario. (GUIZA, 2005)

Con el surgimiento de la computadora personal se hizo posible el almacenamiento de imágenes estáticas y en movimiento, textos y sonidos que podían adaptarse a nuevas situaciones. El creciente desarrollo tecnológico permitió a las computadoras el control de dispositivos de comunicación, de edición de textos, videos y audio, abriendo caminos a la evolución del hardware y software de forma tal que brindaran posibilidades de combinar varios medios, surgiendo de esta forma el empleo del término multimedia.

“En la actualidad en informática se suele identificar como multimedia a la integración de dos o más medios de comunicación que pueden ser controlados o manipulados por el usuario en una computadora. O sea, es un sistema informático interactivo, controlable por el usuario que integra diferentes medios como el texto, la imagen, el video, el sonido y las animaciones.” (LABAÑINO, 2005)

Para que un software sea considerado una multimedia debe cumplir con ciertas características: (DÍAZ, 2000)

- ✓ La integración de diferentes tipos o formas de información: gráfica, sonora, textual y visual.
- ✓ La presentación y el tratamiento de la información no debe ser de forma lineal o secuencial, sino en forma de red y con múltiples ramificaciones y diferentes niveles.

- ✓ La ampliación de las posibilidades de la interacción hasta hacer posible la inmediatez de las respuestas.
- ✓ La sencillez de su uso muy ligada a la intuición.

La multimedia junto a las técnicas del hipertexto, agrega interactividad, lo que hace que el usuario pueda navegar a través de la aplicación, a su libre elección, de acuerdo con sus intereses o necesidades de conocimiento. Esta propiedad hace que este tipo de aplicación educativa se convierta en el mejor colaborador y motivador en el aprendizaje creando la multimedia interactiva.

1.6.3 Multimedia Interactiva.

Dentro del grupo de los materiales multimedia, que integran diversos elementos textuales (secuenciales e hipertextuales) y audiovisuales (gráficos, sonido, vídeo, animaciones...), están los materiales multimedia interactivos. Cuando se habla de interactividad, se dice de un programa que permite una interacción, a modo de diálogo, entre el ordenador y el usuario.

También se puede decir que "...la interacción es la capacidad del usuario de relacionarse con un sistema, con vistas a modificar en todo momento sus parámetros de funcionamiento; actividad que incluye, además, la posibilidad de controlar la navegación, es decir, decidir en qué parte de la aplicación se quiere estar y qué acciones se desea desarrollar. Requiere el empleo de dispositivos de entrada, como el teclado y, sobre todo, el ratón." (COMPANIONI, 2005)

La interacción entre usuario y máquina, se ha visto modificada con respecto a entornos anteriores en cuanto a calidad, velocidad y modo de acceso. Estos sistemas permiten con facilidad situarse dentro de lo que se ha venido denominando "entornos amigables" debido a la fluidez de la comunicación máquina-usuario en ambos sentidos, gracias a un lenguaje y un sistema de acceso amigables y no necesitar una previa formación en el manejo de la máquina. (MORÓN, 2004)

Un entorno multimedia interactivo permite a los aprendices interactuar activamente con la información y luego reestructurarla en formas significativas personales. En la educación, con las multimedia interactivas, los alumnos pueden interactuar con el sistema y recibir información simultánea sobre el tema estudiado de manera más ilustrativa y amena, combinando los diferentes sentidos para lograr una mayor interpretación

y entendimiento del contenido, ampliando las posibilidades de retención de la información y entendimiento del entorno a estudiar.

La aparición de la multimedia brinda una herramienta de grandes potencialidades para ser utilizada en el proceso docente por su calidad, efectividad y novedad en el sistema de aprendizaje para el estudiante y el profesor.

1.6.4 Definición de Hipertexto.

En la actualidad los hipertextos forman parte de la vida cotidiana, se han impuesto a los libros, han sustituido de forma digital a los manuales técnicos existentes. Se encuentran estrechamente relacionados con los novedosos cambios de las comunicaciones y las nuevas tecnologías favoreciendo el desarrollo del proceso docente educativo.

El hipertexto es un documento donde sólo se presenta información en bloques de texto unidos entre sí por nexos o vínculos que hacen que el lector elija o decida en cada momento el camino de lectura a seguir en función de los posibles itinerarios que le ofrece el programa. (DÍAZ, 2000)

Además “el hipertexto es considerado una herramienta y como herramienta posee significado social, refleja prácticas y valores sociales, es además un producto que conforma el fluir de nuestras estructuras culturales” (MARTÍNEZ, 2001). Tiene como característica fundamental la discontinuidad, el salto, el desplazamiento repentino de la posición del lector en el texto y es posible consultar la información que almacena mediante la navegación.

Un hipertexto tiene los siguientes elementos: secciones, enlaces o hipervínculos y anclajes. Las secciones o nodos son los componentes del hipertexto o hiperdocumento. Los enlaces son las uniones entre nodos que facilitan la lectura secuencial o no secuencial del documento. Los anclajes son los puntos de activación de los enlaces. (BARROS, 2002)

En resumen, el hipertexto no es más que una nueva forma de organizar la información contenida en los textos que se puede leer de manera no secuencial o lineal de acuerdo sea la necesidad.

1.6.5 Definición de Hipermedia.

El término Hipermedia, combinación de los conceptos **Hipertexto** y **multimedia**, hace referencia a una tecnología de construcción de (hiper) 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, video, etc.) que conforman el documento. (BARROS, 2002)

Cuando al hipertexto se le añaden dibujos, imágenes, sonidos, etc. aparece el concepto de hipermedia. Ambos son documentos no lineales, cuya información está unida por vínculos que configuran una red o malla de información, estando la diferencia entre ellos en que en el hipertexto solo hay información textual, mientras que el hipermedia incluye aparte del texto, imágenes y sonidos. (DÍAZ, 2000)

De manera general se concluye que un documento hipermedia es siempre una multimedia, pero no al revés que se define como el resultado de la combinación del hipertexto y la multimedia conjugando los beneficios de ambas tecnologías.

1.6.6 Clasificación de Multimedia.

Las aplicaciones multimedia se pueden clasificar en: (DÍAZ, 2000)

- ✓ **Enciclopédicas o de divulgación de conocimientos:** Enciclopedias generales o específicas, así como las obras monotemáticas dedicadas a cualquiera de las ramas del conocimiento (historia, geografía, arte, ciencias, etc.) y no tienen una propuesta o proyecto pedagógico.
- ✓ **Informativas:** Tienen un carácter noticioso, o constituyen memorias de congresos, o informan sobre direcciones, lugares, horarios, precios, intereses, actividades, o son catálogos, listas de referencias, etc.
- ✓ **Educativas o didácticas:** Tienen un proyecto educativo, su fin es enseñar.
- ✓ **Promocionales:** Promueven productos y servicios, instituciones, actividades, proyectos, valores culturales, éticos, etc. Publicitarios y comerciales.
- ✓ **De entretenimiento:** Juegos, adivinanzas, caricaturas, dibujos, cuentos, películas, libros vivos, etc.

1.6.7 Aportes de la Multimedia al proceso de enseñanza aprendizaje.

A nivel mundial existen un gran número de investigaciones realizadas las cuales se han enfocado a analizar cómo ha sido utilizada la multimedia en el proceso de enseñanza aprendizaje y los efectos que ésta ha tenido en dicho proceso.

Según un estudio realizado por Gayle J. Yaverbaum y Uma Nadarajan en 1996 sobre el aprendizaje de conceptos básicos en telecomunicaciones, con el objetivo de determinar si impactaba en el aprendizaje el uso de multimedias, se conformó un grupo de 29 estudiantes el cuál recibió los conceptos básicos de telecomunicaciones a través de una multimedia y otro de 33 estudiantes que recibieron los mismos conceptos a través del método convencional de enseñanza. Se pudo comprobar que el primer grupo, después de aplicarles a ambos grupos una serie de pruebas sobre la asignatura impartida, demostró mayor consistencia en el aprendizaje del contenido que el grupo que recibió el contenido de forma convencional por lo que se deduce que el uso de multimedias es valioso, como un medio más de enseñanza aprendizaje en las enseñanzas de temas ingenieriles. (GUIZA, 2005)

En la Tabla 1, se aprecian los resultados de las pre-pruebas y post-pruebas con un nivel de 95 por ciento de confiabilidad. Las diferencias entre las calificaciones fueron significativas por lo que se deduce que el uso de multimedia propicia un mejor aprendizaje que el método convencional. (GUIZA, 2005)

Tabla 1. Calificaciones promedio.

	Cátedra	Material Multimedia
Pre-prueba promedio	7.24	8.45
Post-prueba promedio	16.67	17.24

1.7 Tendencias tecnológicas actuales.

La digitalización y los nuevos soportes electrónicos están dando lugar a nuevas formas de almacenar y presentar la información. Los tutoriales multimedia, las bases de datos en línea, las bibliotecas electrónicas, los hipertextos distribuidos, Ambientes Virtuales de Aprendizaje (AVA), etc. son nuevas maneras de presentar y acceder al conocimiento que superan en determinados contextos las formas tradicionales de la explicación oral, la pizarra, los apuntes y el manual.

Una de las tendencias actuales en materia de educación a distancia son los Ambientes Virtuales de Aprendizaje (AVA). Estos ambientes se desarrollan a través de cursos en línea, que presentan contenidos en diferentes formatos con el fin de aprovechar las bondades que aportan las tecnologías de la información y la comunicación como son el texto, la imagen, las animaciones, los sonidos, los diagramas, los mapas conceptuales, las cápsulas de video y otros; además se propicia la interacción facilitador-alumno, alumno-alumno, facilitador-facilitador, a través de las herramientas de comunicación como el *chat*, foros de discusión y el correo electrónico.

Actualmente existen en el mercado varias herramientas, unas hechas por Instituciones universitarias y otras por industrias, para la creación de enseñanza basada en la Web (*Web Based Instruction*). Algunas de las más usadas se exponen a continuación:

1. **WebCT**, desarrollado por el Departamento de Ciencia Informática de la Universidad de British Columbia, Canadá, es una herramienta que facilita la creación de ambientes educativos basados en Web. Puede ser usada para crear cursos en línea completos, o simplemente para publicar materiales que complementen cursos ya existentes. Este software utiliza tecnología de navegadores estándares para el acceso de los estudiantes y para el diseño de cursos por parte del instructor. *WebCT* también usa una serie de herramientas para añadir características a cursos existentes, incluyendo correo electrónico, un sistema de conferencias, chat, gestión de cursos, y evaluación/exámenes. (PUC, 2007)
2. **Moodle** es una herramienta para producir cursos basados en Internet y páginas web, permite presentar un curso contentivo de recursos de información (en formato textual o tabular, fotografías o diagramas, audio o video, páginas web o documentos Acrobat, entre muchos otros) así como

actividades para estudiantes tipo tareas enviadas por la web, exámenes, encuestas, foros entre otros. (FERNÁNDEZ, 2005)

La UCI cuenta con un AVA para la preparación y aprendizaje de los estudiantes, profesores y trabajadores del centro. Actualmente, gracias al Entorno Virtual de Aprendizaje (EVA) todas las personas que interactúan con el mismo pueden acceder a los materiales que necesitan para el estudio de las diferentes asignaturas que reciben como parte del curso que se imparte para cada año, así como a los diferentes cursos optativos para aumentar la preparación profesional y las pruebas online que permiten realizar una autoevaluación sobre los conocimientos adquiridos. Estos son solo algunos de los servicios que ofrece este entorno pues existen otros como los foros que tienen el objetivo de aumentar la comunicación y la preparación del personal que utiliza dicho entorno.

1.7.1 Herramientas para la creación de Software Multimedia.

Con el avance de la tecnología han surgido varias herramientas para el desarrollo de productos multimedia. A continuación se enuncian diferentes herramientas que el autor propone emplear destinadas al desarrollo de aplicaciones multimedia, las cuales tienen cualidades y características particulares.

✓ **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 el entorno de autor multimedia más potente 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)

✓ **ToolBook**

Esta herramienta está más bien definida para la creación de aplicaciones destinadas al proceso de enseñanza aprendizaje. ToolBook permite diseñar una amplia variedad de actividades de distinta índole, combinando textos, imágenes, sonidos, animaciones, vínculos a sitios web y autoevaluaciones en distintos formatos. (MOTA, 2003)

Desde su aparición a principios de 1990, ToolBook se ha convertido en una de las aplicaciones preferidas por los expertos constructores de aplicaciones multimedia pero la misma tiene una gran desventaja y es que esta herramienta tiene gran dependencia de la plataforma Windows ya que para poder visualizar el contenido en ordenadores que no poseen ToolBook se proporciona un reproductor gratuito que trabaja sobre cualquier versión existente de Windows pero el mismo sólo existe para plataformas Windows, no para Macintosh ni ninguna otra. (PASCUAL, 2003)

✓ **Authorware**

Es un programa orientado a objetos que se utiliza para crear aplicaciones multimedia. Se trata de un software diseñado para desarrollar manuales, enciclopedias interactivas y todo tipo de material, ya que permite combinar imágenes, sonido, animaciones digitales, video y todos los elementos necesarios. Authorware utiliza la interfaz que es característica en los productos de Macromedia MX, por lo que son mínimas las dificultades para familiarizarse con el uso del programa. (ALEGSA, 2008)

Authorware proporciona poca flexibilidad, teniendo que ajustarse explícitamente a los patrones preestablecidos. Sirva un ejemplo. En Authorware el método de presentación de textos largos consiste en la paginación, o bien, en el desplazamiento del mismo. Si lo que se desea es el desplazamiento del texto pero con una barra de desplazamiento personalizada, es poco menos

que imposible. Tras un increíble esfuerzo trabajando con animaciones, máscaras y otros recursos, sería posible llegar a conseguirlo. Pero si lo que deseamos no es una, sino dos cajas de texto con barras de desplazamiento personalizadas, entonces sí que nos podemos olvidar de Authorware, concluyendo así que la flexibilidad de Authorware es muy limitada. (PASCUAL, 2003)

✓ **Revolution**

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

Revolution utiliza un lenguaje de programación de alto nivel orientado a objetos llamado **Transcript**. Esta herramienta permite proyectar y desarrollar aplicaciones fácil y rápidamente. Sin embargo hay que reconocer también que las aplicaciones generadas son, por lo general, algo más lentas y "voluminosas" que las desarrolladas con lenguajes de bajo nivel del tipo de C ó C++.

(VERA, 2003)

✓ **Macromedia Flash 8**

Hace ya algunos años la aparición de flash con sus diferentes versiones, ha traído hasta nuestras pantallas todo un mundo de interactividad, movimiento y color. Macromedia Flash o simplemente Flash diseña gráficos de vectores; gráficos definidos como puntos y líneas en lugar de píxeles. Es decir que los vectores son como un conjunto de instrucciones matemáticas que por medio de valores le dan forma a una imagen. Además permite incluir audio comprimido en diversos formatos

como el mp3, importar gráficos creados con otros programas, formularios y algo de programación. (HENST, 1999)

Flash mezcla gráficos vectoriales, bitmaps, sonidos, animaciones y una interactividad avanzada para crear multimedias que atraigan y entretengan a los clientes. Permite a los diseñadores y desarrolladores integrar video, texto, audio y gráficos en experiencias dinámicas que le permiten al cliente adentrarse en su vivencia y que producen resultados superiores para marketing y presentaciones interactivas, aprendizaje electrónico e interfaces de usuario de aplicaciones.

En la actualidad, Macromedia Flash va más allá de las animaciones simples, convirtiéndose en una herramienta de desarrollo completa, para crear principalmente elementos multimedia e interactivos para Internet. Flash es el entorno de autoría más avanzado del mercado para la creación de sitios Web interactivos, experiencias digitales y contenidos para dispositivos móviles por lo que las aplicaciones generadas con esta herramienta son multiplataforma.

La versión 8 de Flash presenta facilidad de manejo, una gran potencia para diseñar y crear contenido interactivo dinámico con vídeo, gráficos y animación obteniendo sitios Web, presentaciones, etc., perfecciona la creación de contenidos interactivos e incorpora innovaciones en opciones de texto y filtrado al más puro estilo de Photoshop por lo que ya no es necesario importar textos y gráficos creados con otras herramientas. Además, esta versión recupera el asistente de ActionScript que había desaparecido en la versión anterior. (AULACLIC, 2006)

Esta herramienta está orientada claramente a diseñadores, aunque con alguna que otra atractiva novedad para la comunidad de desarrolladores, la reciente versión 8 de Macromedia Flash ofrece una serie de importantes mejoras que ameritan la actualización del producto. (AULACLIC, 2006)

1.7.2 Herramienta a utilizar.

Para el desarrollo del producto, Multimedia Interactiva Pruebas de Software, se escogió la herramienta de autor Macromedia Flash, específicamente la versión 8.0, porque es el entorno de autoría más avanzado del mercado para la creación de aplicaciones con tecnología multimedia interactivas, porque las aplicaciones se pueden ejecutar en la Web, en Windows, Macintosh, Unix y Linux. Otro punto a su favor

es su potencia en la creación de animaciones. Además se tuvo en cuenta que se necesitaba de un lenguaje potente como lo es ActionScript para tener una mayor potencia en la programación del software con tecnología multimedia. (AULACLIC, 2006)

A continuación se presentan algunas características de Macromedia Flash 8 que demuestran el por qué se seleccionó esta herramienta para el desarrollo de la aplicación: (AULACLIC, 2006)

- ✓ **Diseños más atractivos:** Flash 8 permite el uso de efectos visuales que facilita la creación de animaciones, presentaciones y formularios más atractivos y profesionales, así mismo incorpora nuevos mecanismos, tales como filtros y modos de mezcla, para hacer el trabajo más cómodo y rápido.
- ✓ **Optimización de fuentes:** Incorpora también opciones de legibilidad para fuentes pequeñas, haciendo la lectura de los textos más agradables y de alta legibilidad. Además de poder modificar la optimización, Flash 8 permite también la selección de configuraciones preestablecidas para textos dinámicos y estáticos.
- ✓ **Mayor potencia de animación:** Flash 8 permite un mayor control de las interpolaciones habilitando un modo de edición desde el que se podrá modificar independientemente la velocidad en la que se apliquen los diferentes cambios de rotación, forma, color, movimiento, etc., de las interpolaciones.
- ✓ **Mayor potencia gráfica:** Evita la repetición innecesaria de la representación de objetos vectoriales simplemente señalando un objeto como mapa de bits. Aunque el objeto se convierta al formato de mapa de bits, los datos vectoriales se mantienen tal cual, con el fin de que, en todo momento, el objeto pueda convertirse de nuevo al formato vectorial.
- ✓ **Nuevas opciones para el trabajo con video:** Flash 8 utiliza un nuevo sistema de importación, compresión e implementación de video, que resulta extremadamente sencillo en comparación con versiones anteriores.
- ✓ **Utilización de filtros y blending modes:** Efectos y filtros típicos de Photoshop (como sombras y difuminados), están ya disponibles en el entorno de edición de Flash, con lo cuál ya no será necesario importar imágenes obtenidas de otras aplicaciones. No obstante, lo que genera más entusiasmo es la posibilidad de aplicar y animar estos filtros en tiempo real a través de

ActionScript, algo que sin dudas abre un nuevo campo de expresión para la comunidad de usuarios de Flash.

- ✓ **Nuevos formatos aceptados para la carga dinámica de imágenes:** El soporte para la carga dinámica de imágenes ha sido extendido a los formatos GIF y PNG. Hasta la versión 7 de Flash sólo podían importarse dinámicamente imágenes en formato JPG

1.7.3 Metodologías a considerar para el desarrollo del software.

RMM (Metodología de Administración de Relaciones)

La RMM o Metodología de Administración de Relaciones, se define como un proceso de análisis, diseño y desarrollo de aplicaciones hipermedia. Los elementos principales de este método son el modelo E-R (Entidad-Relación) y el modelo RMDM (Relationship Management Data Model) basado en el modelo HDM. (LAPUENTE, 2006)

Es una metodología creada por Isakowitz, Stohr y Balasubramanian, se basa en los conceptos implantados en el Modelo de diseño de hipertexto HDM, es decir, en los tipos de entidades y las entidades mismas. Una de las aportaciones más importantes de esta metodología son los conceptos de slices y m-slices, ambos consisten en la agrupación de datos de una entidad en diferentes pantallas.

Propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipermedia de la aplicación. Los objetos del dominio se definen con la ayuda de entidades, atributos y relaciones asociativas (LAPUENTE, 2006) . Tiene como objetivo mejorar la navegación por medio del análisis de las entidades que se presentan en el sistema. Visto teóricamente con este modelo se obtiene una navegación más estructurada logrando que para el usuario esta sea más intuitiva.

Programación Extrema (XP)

Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en el desarrollo del software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo. XP se basa en la retroalimentación continua entre el cliente y el equipo de desarrollo, comunicación fluida entre todos los participantes, simplicidad en las

soluciones implementadas y coraje para enfrentar los cambios. XP se define como especialmente adecuada para proyectos con requisitos imprecisos y muy cambiantes, y donde existe un alto riesgo técnico. Los principios y prácticas son de sentido común pero llevadas al extremo, de ahí proviene su nombre. A pesar de ser muy útil para el desarrollo de software en un tiempo relativamente corto y ser más flexible en cuanto a requisitos cambiantes, esta metodología presenta varios inconvenientes: (CANÓS, 2002)

- ✓ No produce una potente documentación del sistema.
- ✓ Está dirigida a equipos pequeños o medianos (no más de 10 integrantes).
- ✓ El entorno físico debe ser un ambiente que permita la comunicación y colaboración entre todos los miembros del equipo durante todo el tiempo.
- ✓ El usuario debe convertirse en un integrante más del grupo de desarrollo pues de otra forma la metodología no daría los resultados esperados en la creación del software.
- ✓ Falta aún un cuerpo de conocimiento consensuado respecto de los aspectos teóricos y prácticos de la utilización de metodologías ágiles, así como una mayor consolidación de los resultados de aplicación.
- ✓ Aún faltan líneas como: métricas y evaluación del proceso, herramientas específicas para apoyar prácticas ágiles y aspectos humanos y de trabajo en equipo.

Proceso Unificado de Software (RUP)

RUP es un proceso que define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Como su enfoque está basado en modelos utiliza un lenguaje bien definido para tal fin, el Lenguaje Unificado de Modelado (UML). Está preparado para desarrollar grandes y complejos proyectos, unifica los mejores elementos de metodologías anteriores y es orientado a objetos.

En el libro “El Proceso Unificado de Desarrollo de Software”, James Rumbaugh, Grady Booch e Ivar Jacobson, opinan que “El problema del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las múltiples cadenas de trabajo de un gran proyecto de software”. Para desarrollar un software se necesita una forma coordinada de trabajo, un proceso que integre las múltiples facetas del desarrollo, cuyo objetivo será producir software de alta calidad, es decir, que cumpla con los

requerimientos de los usuarios dentro de una planificación y presupuesto establecido, cubriendo el ciclo de vida y desarrollo del software (JACOBSON, 2000)

Características del Proceso Unificado de Software (RUP): (JACOBSON, 2000)

- ✓ **Dirigido por casos de uso:** La razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado usuario y los utiliza tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígase diseño, implementación y prueba.
- ✓ **Centrado en la arquitectura:** La arquitectura es una vista del diseño completo con las características más importantes. Esta no sólo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red, con los que debe coexistir el sistema. La arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.
- ✓ **Ser iterativo e incremental:** El alto nivel de complejidad de los sistemas actuales, hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un producto.

Un proyecto realizado siguiendo RUP se divide en cuatro fases y nueve flujos de trabajo. Las fases y los flujos de trabajo están estrechamente relacionados.

En cada fase se ejecutarán una o varias iteraciones (de tamaño variable según el proyecto). Una vez concluida cada una de ellas se encuentra un hito principal del proyecto:

- ✓ Al finalizar la fase de Inicio está presente el hito “Objetivos del Proyecto”. Durante la fase de inicio se define el modelo del negocio y el alcance del proyecto. Se identifican todos los actores y Casos de Uso, y se diseñan los Casos de Uso más esenciales (aproximadamente el 20 por ciento del modelo completo).
- ✓ Al concluir la fase de Elaboración se encuentra el hito “Arquitectura Estable”. El propósito de la fase de elaboración es analizar el dominio del problema, establecer los cimientos de la arquitectura, desarrollar el plan del proyecto y eliminar los mayores riesgos.
- ✓ En el final de la fase de Construcción se encuentra la “Capacidad Operacional Inicial” como hito. En esta fase se determina si el sistema es apto para su despliegue en versión beta al entorno operacional. La finalidad principal de esta fase es alcanzar la capacidad operacional del producto de forma incremental a través de las sucesivas iteraciones.
- ✓ Como última fase se encuentra la fase de Transición, en la cual al finalizar está presente el hito “Entrega del Sistema”. La finalidad de la fase es poner el producto en manos de los usuarios finales, para lo que se requiere desarrollar nuevas versiones actualizadas del producto, completar la documentación, entrenar al usuario en el manejo del producto, y en general tareas relacionadas con el ajuste, configuración, instalación y facilidad de uso del producto.

1.7.4. Lenguajes de modelado.

UML (Lenguaje Unificado de Modelado) y el Rational Rose.

UML está diseñado a través de un lenguaje de diagramas y artefactos fácilmente ajustables para especificar aspectos distintivos de un sistema a modelar. Se agrupan en cuatro categorías, diagramas de caso de uso y estructurales (los cuales interactúan directamente con las descripciones de los modelos estáticos, estructurales y de comportamiento dinámicos identificados anteriormente), de comportamiento e implementación. Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML Booch, Rumbaugh y Jacobson, que soporta de forma completa la especificación del UML, cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases y entregables. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas

creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. (JACOBSON, 2000)

El desarrollo de sistemas con UML siguiendo el proceso unificado incluye actividades específicas, cada una de ellas a su vez contienen otras subactividades las cuales sirven como una guía de cómo deben ser las actividades desarrolladas y secuenciadas con el fin de obtener sistemas exitosos; consecuentemente el desarrollo de los sistemas puede variar de desarrollador en desarrollador, de proyecto en proyecto, de empresa en empresa adoptando siempre un Proceso de Desarrollo. (JACOBSON, 2000)

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

Han sido propuestos muchos lenguajes de modelado para la descripción del proceso de desarrollo de aplicaciones multimedia, sin embargo aún no existe un estándar que cubra todos los aspectos relacionados con el comportamiento dinámico e interactivo asociado a las interfaces gráficas para una generalización de herramientas, productos y procesos.

El Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L) se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario, siendo este un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. (ENGELS, 2001)

Para OMMMA – L se puede modelar la estructura a través de diagramas de objetos y clases, mientras que el comportamiento puede ser descrito en los diagramas de interacción, estado y actividad. Por último, la distribución espacial de media contemplada en el modelo vista, puede ser descrita a través de un nuevo artefacto propuesto para el lenguaje, el diagrama de presentación. La semántica asociada a dichos diagramas conserva en muchos casos su significado, en otras se adaptan a la interpretación de los conceptos propios de multimedia. UML ofrece varios diagramas para modelar el comportamiento de una aplicación, dado el énfasis que muestran en modelar restricciones de tiempo los diagramas de secuencia se destacan en OMMMA – L para modelar el comportamiento temporal predefinido de una aplicación

multimedia, además tiene características como la espera de activación y desactivación para el manejo de la sincronización entre medias y el perfeccionamiento del eje de tiempo mediante la introducción de marcas de la parametrización de diagramas de secuencia, para diferenciar su funcionamiento entre los establecimientos de sincronización temporal y el tradicional paso de mensajes. (WESLEY, 2002)

Este lenguaje está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular, las cuales son:

- ✓ **Vista Lógica:** modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias. 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). Estos diagramas de presentación pueden ser divididos en capas virtuales de presentación donde en cada uno de ellas sólo se haga referencia a una clase específica de componentes (por ejemplo, una vista para los objetos de visualización y otra para los de interacción, u otro tipo de división para la representación de los intereses de los desarrolladores).
- ✓ **Vista de Comportamiento Temporal Predefinido:** modelada por el diagrama de secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos; marcas de inicio y fin de ejecución que permite soportar su reusabilidad;

marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media y la activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.

- ✓ **Vista de Control Interactivo:** modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML. Sintácticamente igual a este último pero 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. (BRUNO, 2005)

1.7.5 Fundamentación de la metodología a utilizar.

La metodología de desarrollo de software RUP, ha sido elegida para el desarrollo de la aplicación multimedia pues además de ser puntera a nivel mundial, permite una mejor interacción con el usuario en todo el proceso de desarrollo software, lo cual puede ayudar a maximizar en tiempo y calidad el desarrollo de la aplicación propuesta.

Rational Rose, por ser la herramienta CASE que soporta de forma completa la especificación de UML y porque esta herramienta es comercializada por la institución que desarrolló UML.

Y se escogió OMMMA -L para desarrollar la multimedia ya que este no es un nuevo lenguaje, sino que es una extensión del UML, que se imparte en la Universidad y del cual se tiene conocimiento, por lo que no se tiene que estudiar a fondo, solo se deben interpretar las características extendidas que corresponden al OMMMA -L, centradas a la lógica de funcionamiento de una multimedia. Es robusto y altamente descriptivo, refleja el proceso en todas sus etapas y hereda del RUP, el ciclo de vida basado en iteraciones y el flujo de trabajo incremental, centrado en casos de uso y en la arquitectura.

1.7.6 Lenguajes utilizados en la multimedia.

XML (Lenguaje de Marcas Extensible)

El Lenguaje de Marcas Extensible (XML) es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Es una simplificación y adaptación del SGML (Lenguaje de Marcación Generalizado) y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML). Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. (WIKIPEDIA, 2008)

XML es la lengua franca de Internet, se propone como un estándar para el intercambio de información estructurada entre diferentes plataformas. Se puede usar en bases de datos, editores de texto, hojas de cálculo, etc. Es una tecnología sencilla y tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil.

Entre sus principales características se encuentran: (GONZÁLEZ, 2005)

- ✓ Es una arquitectura más abierta y extensible. No se necesita versiones para que puedan funcionar en futuras aplicaciones. Los identificadores pueden crearse de manera simple y ser adaptados en el acto en Internet/Intranet por medio de un validador de documentos (parser).
- ✓ Se realiza el intercambio de documentos entre las aplicaciones tanto en la propia PC como en una red local o extensa.
- ✓ La extensibilidad y flexibilidad de este lenguaje permite agrupar una variedad amplia de aplicaciones, desde páginas web hasta bases de datos.
- ✓ En el caso de aplicaciones para la Web, los motores de búsqueda devuelven respuestas más adecuadas y precisas, ya que la codificación del contenido Web en XML consigue que la estructura de la información resulte más accesible.
- ✓ El concepto de "hipertexto" se desarrolla ampliamente (permite denominación independiente de la ubicación, enlaces bidireccionales, enlaces que pueden especificarse y gestionarse desde fuera del documento, hiperenlaces múltiples, enlaces agrupados, atributos para los enlaces, etc. Creado a través del XLL (Lenguaje de Enlaces Extensible).

XML es un lenguaje de marcas con el que se consigue la portabilidad de los datos. Macromedia Flash, herramienta a usar para la confección de la aplicación, permite la interpretación de estos datos e incluso la generación de formatos propios. El uso del objeto XML, destinado exclusivamente a la gestión de archivos y contenidos formateados en este estándar, permite a una película Flash importar y exportar fácilmente información desde y hacia lenguajes de servidor o bases de datos. XML se encarga de estructurar estos datos de forma tal que puedan ser leídos e interpretados sin problemas por cada una de las partes.

Su campo de acción no se limita únicamente a las aplicaciones de carácter dinámico. Una de las principales razones por la que se recomienda la integración de XML con ActionScript es evitar una recurrente edición del archivo fuente (.fla) cada vez que se necesite introducir algún cambio en el contenido. (CIBERAULA, 2005)

Las ventajas que provee XML es que la flexibilidad que ofrece es altamente útil para diferentes propósitos, la información se provee de manera estructurada y descriptivamente visual, la información ya no es un conjunto de datos sin sentido, que solo entendía el sistema para el cual es enviada. Ahora puede ser leída tanto por aplicaciones ajenas a la aplicación que las envía (o provee) así como por "humanos", vale decir que la información es accesible también para los desarrolladores y lo más importante que es entendible. (VELÁZQUEZ, 2002)

Flash puede leer XML y eso es un aspecto muy bueno, porque así se tiene una manera de añadir contenidos a una película flash una vez terminada, exportada y publicada. La gran ventaja es que es muy fácil modificar el XML, añadir o quitar información. Flash puede comunicarse con el exterior mediante el protocolo HTTP, puede recibir y enviar datos XML y puede incluso hacer una conexión a un puerto determinado utilizando sockets XML. (FREDDIE, 2004)

ActionScript

ActionScript es un lenguaje de programación orientado a objetos. Fue lanzado con la versión 4 de la herramienta Macromedia Flash, y desde entonces hasta la fecha, ha ido ampliándose poco a poco, hasta llegar a niveles de dinamismo y versatilidad muy altos en las últimas versiones de dicha herramienta. (CIBERAULA, 2005)

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: (CIBERAULA, 2005)

- ✓ 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.

1.8 Análisis crítico de las herramientas de enseñanza existentes.

En la actualidad además de utilizarse los métodos anteriormente mencionados para la preparación de personal en el tema de pruebas de software existen herramientas que sirven como soporte o ayuda para una mejor comprensión del tema. En Argentina, la Biblioteca Nacional de Maestros (BNM, 2008), pone a disposición de sus alumnos y profesores videoconferencias para la capacitación del personal dedicado a la realización de pruebas de software y de calidad de software en general ya que se abordan temas como métricas de software, modelos de calidad de software, etc.

En la UCI, en el EVA se puede encontrar información sobre pruebas de software en forma de documentos que forman parte del objeto de estudio de la asignatura de Ingeniería de Software así como teleconferencias disponibles en la red para la preparación de todos los estudiantes pero no existe un producto en soporte multimedia que posibilite estudiar las pruebas de software de forma interactiva.

Con el objetivo de facilitar una nueva herramienta para una mejor preparación de los estudiantes de la UCI y como apoyo al proceso de enseñanza del tema de pruebas de software la multimedia creada al término de este trabajo podría incluirse en el EVA o en algún sitio web existente en la UCI destinado a Calidad de Software.

Conclusiones

En este capítulo se han analizado conceptos significativos, necesarios para la realización adecuada de este trabajo, como multimedia, hipertexto e hipermedia, logrando un mayor entendimiento de los detalles más importantes para continuar el desarrollo eficiente de la solución del problema a resolver. Posteriormente se llevó a cabo una investigación con el objetivo de conocer otras soluciones existentes en el mundo y de igual forma poder concluir en que no existe ningún software multimedia en la UCI que apoye el proceso de enseñanza aprendizaje de pruebas de software.

Gracias a las tecnologías actuales y al desarrollo de las mismas hoy en día es posible de manera más sencilla y organizada, y con el apoyo de todo un equipo de desarrollo poder construir un software utilizando metodologías que contemplan todos los detalles para la realización del mismo. Tal es el caso de la metodología RUP la cual tras un detallado estudio resultó ser la más óptima para desarrollar esta aplicación con tecnología multimedia utilizando el lenguaje OMMMA-L por ser el más indicado para modelar y analizar el software, y por ser utilizado por la metodología anteriormente mencionada para describir los artefactos que se generan. Por último, para la creación de la aplicación después de un exhaustivo y minucioso estudio de diferentes herramientas de creación de aplicaciones con tecnología multimedia se escogió Macromedia Flash 8 por ser la herramienta más extendida en la actualidad para la creación de este tipo de aplicaciones, por la facilidad de uso que presenta y por ser una herramienta que

genera archivos multiplataforma que pueden ser visualizados en sistema operativos Windows, Macintosh y GNU/Linux.

Capítulo 2: DESCRIPCIÓN DE LA SOLUCIÓN.

Introducción

El presente capítulo está dedicado a describir la solución que se propone para dar cumplimiento al problema planteado en la investigación. En el mismo se hace una breve descripción de la información que contendrá el producto final, se identificarán los principales conceptos asociados al dominio del problema, creándose el modelo de dominio correspondiente, y se identificarán los principales requisitos funcionales y no funcionales. Además, se realiza el diagrama de casos de uso del sistema y una breve descripción de cada uno de estos posibilitándose con esto que cualquier usuario pueda comprender el entorno que se quiere mostrar.

2.1 Identificación de la audiencia.

¿A quién va dirigida la aplicación?. Esta pregunta, permite identificar la audiencia a la que va dirigida esta multimedia. La identificación de la audiencia constituye un aspecto de suma importancia, pues permite la correcta identificación del usuario final del sistema.

Teniendo en cuenta que es muy poco probable que los estudiantes que ingresan a la universidad tengan idea de qué son las pruebas de software, pues este tema no está presente en el plan de clases de ninguna de las vías de ingreso a la Universidad. Por tratarse de un tema que forma parte de la asignatura Ingeniería de Software que se imparte en el tercer año de la carrera, esta multimedia va dirigida principalmente a los estudiantes que cursen este año , como apoyo al plan de estudio de esta asignatura, lo cual no significa que aquellos estudiantes de años inferiores que de forma autodidacta hayan realizado estudios sobre este tema, no puedan utilizar la misma para una mejor comprensión del mismo.

Los estudiantes que cursan esta asignatura en la actualidad, necesitan una aplicación que tenga como premisa fundamental la organización de la información, pues resulta ineficiente el estudio, si la información presenta errores, o no está bien explícita y organizada. El uso de nuevas tecnologías como la multimedia

para apoyar al proceso de enseñanza- aprendizaje, de seguro sería un paso adelante en la realización de un proceso pleno y con calidad.

2.2 Especificación del contenido.

El contenido del producto multimedia interactivo Pruebas de Software está dividido en 6 temas, estos son:

Tema 1: Fundamentos de Pruebas de Software.

Este tema está compuesto por cinco epígrafes en los que el usuario podrá conocer las principales definiciones de pruebas de software, los objetivos fundamentales, principios y características y facilidad de las mismas.

Tema 2: Flujo de Pruebas

El tema Flujo de Pruebas está compuesto por 3 epígrafes principales: Roles, Actividades, Artefactos. El usuario podrá conocer las responsabilidades y tareas que realizan los encargados de realizar las pruebas de software; las actividades que se realizan durante el proceso de pruebas y los artefactos que generan cada una de estas actividades. Además, el epígrafe Artefactos contendrá 2 subepígrafes: Plan de Prueba y Estrategias de Prueba, los mismos forman parte del epígrafe Artefactos donde se explicará qué es un plan de prueba, su propósito y los aspectos principales que contiene y qué es una estrategia de prueba, sus características y los aspectos estratégicos que se deben tener en cuenta a la hora de crear una buena estrategia de prueba.

Tema 3: Métodos de Prueba

En este tema el usuario conocerá los principales métodos de pruebas que existen, sus características y los tipos de pruebas que se aplican para cada método existente. El tema está compuesto por dos epígrafes fundamentales: Pruebas de Caja Blanca que a su vez contiene 4 subepígrafes donde se enuncian las principales técnicas de pruebas de caja blanca que existen y Pruebas de Caja Negra que contiene 4 subepígrafes donde se enuncian las principales técnicas de pruebas de caja negra existentes.

Tema 4: Proceso de Pruebas

En este tema el usuario podrá conocer el proceso de pruebas, dónde comienza el proceso, a partir de qué artefacto se generan las pruebas, en qué momento se ejecutan las mismas, qué acciones se deben realizar después de la ejecución, etc. Además se explicará qué es un caso de prueba y cuál es el procedimiento que se lleva a cabo para la realización de un caso de uso a partir de la descripción del mismo. El tema está compuesto por tres epígrafes: Proceso de Pruebas, Casos de Uso y Procedimiento de Pruebas.

Tema 5: Herramientas

En este tema se le ofrecen al usuario herramientas existentes para la automatización de pruebas de software.

2.2.1 Mapa Conceptual.

Además de los 6 temas antes mencionados, el producto contiene un mapa conceptual que resume de forma simplificada el contenido del curso. Un mapa conceptual es la representación gráfica de un conjunto de conceptos interrelacionados. Es un esquema gráfico en el que se muestra, o se intenta mostrar, a un conjunto de ideas y sus relaciones conceptuales. El objetivo final es expresar la estructura conceptual del tema representado.

Un mapa conceptual sirve para mostrar la estructura conceptual de un conocimiento. Su realización activa la capacidad de síntesis de su autor haciendo explícito su conocimiento significativo. Por tanto es un instrumento altamente útil para la formación, ya sea como actividad de aprendizaje o de evaluación. (ROVIRA, 2003)

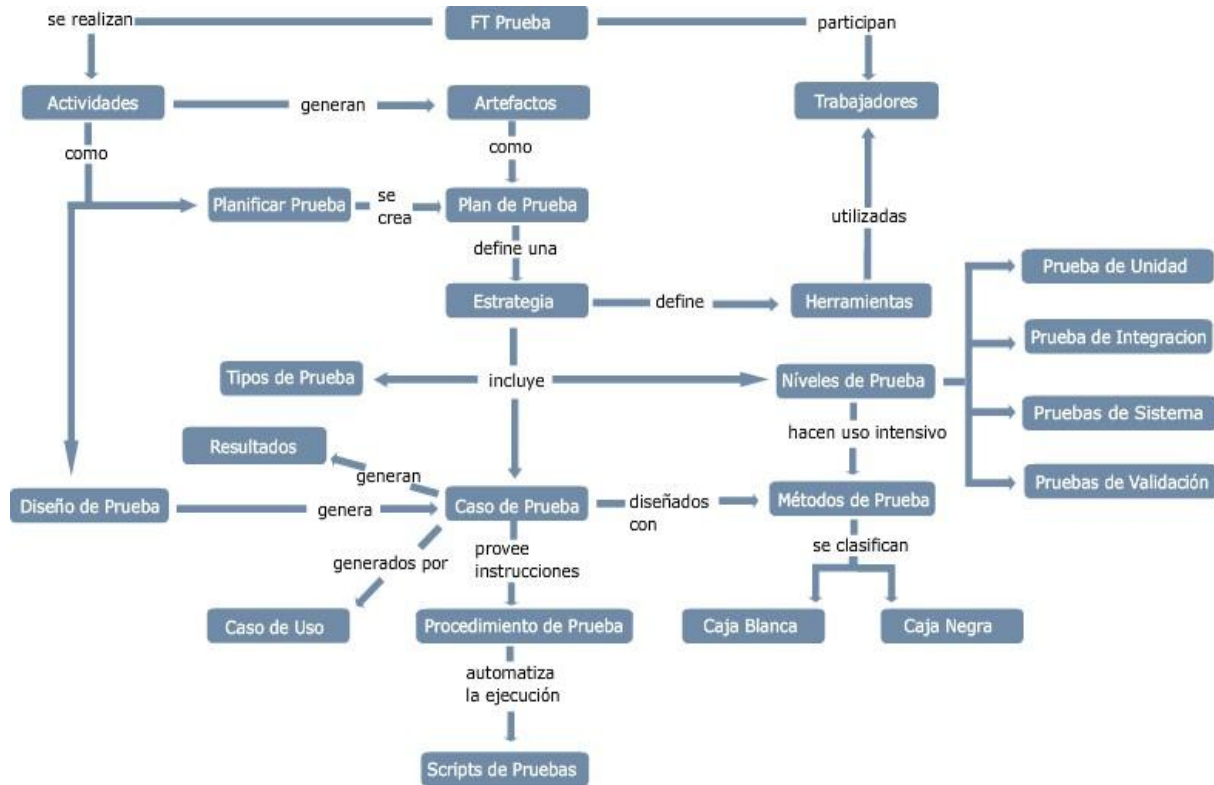


Fig. 1 Mapa Conceptual Pruebas de Software

2.3 Descripción del Modelo de Dominio.

Al no contarse con una visión clara de los procesos del negocio se plantea un modelo de dominio que permita una mejor comprensión de los conceptos del sistema. Para esto se describe el modelo de dominio a través de un diagrama UML donde se especifican las principales clases conceptuales que pueden intervenir en el sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema.

2.3.1 Diagrama de clases del Modelo de Dominio.

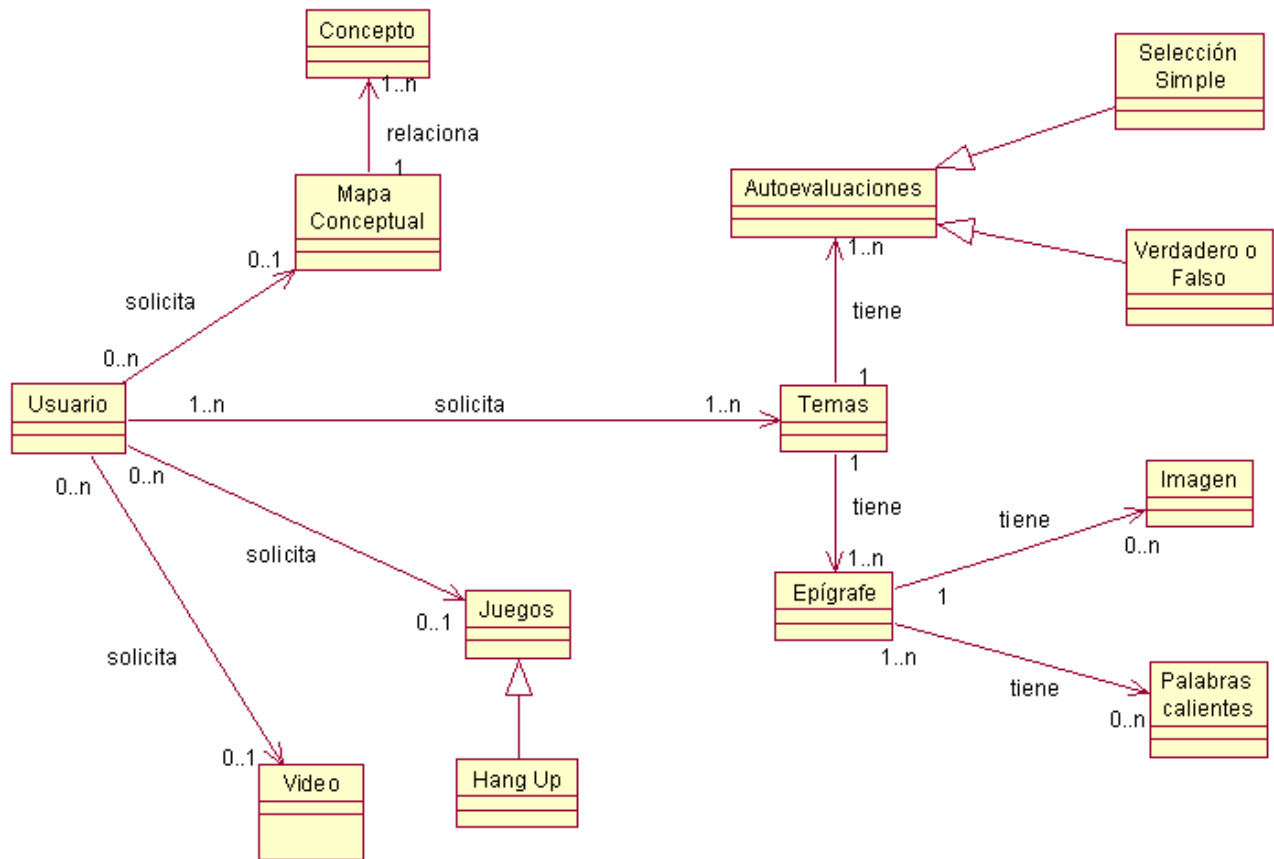


Fig. 2 Modelo de Dominio

2.3.2 Análisis de los conceptos de dominio.

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

- ✓ **Usuario:** Es toda persona que interactúe con el sistema.

- ✓ **Temas:** Se le denomina temas al contenido del curso que está dividido en 5 tópicos fundamentales: Fundamentos de Pruebas de Software, Estrategias de Pruebas de Software, Métodos de Pruebas, Diseño de Casos de Prueba y Herramientas de Pruebas.
- ✓ **Epígrafe:** Se le denomina epígrafe a los diferentes contenidos que se tratan dentro de un tema.
- ✓ **Palabra Caliente:** hipertexto que abre una ventana flotante que profundiza en el significado de alguna palabra o concepto tratado en un epígrafe.
- ✓ **Juego:** Representan objetos que ejercitan de una manera didáctica los conocimientos adquiridos como es el caso del juego del Ahorcado, donde el usuario intentará descubrir palabras claves pertenecientes al tema del curso.
- ✓ **Video:** representa una videoconferencia sobre pruebas de software.
- ✓ **Mapa Conceptual:** Mapa estructural de conceptos relacionados entre sí.
- ✓ **Concepto:** Los conceptos son las estructuras o esquemas mentales mediante las cuales se identifican las características esenciales para definir las cosas, hechos o eventos que dan significado a nuestra realidad inherente y circundante.
- ✓ **Evaluaciones:** Representan el conjunto de ejercicios divididos por temas para la evaluación de los contenidos (3 o más ejercicios por tema).
- ✓ **Imagen:** Gráficos y figuras que se utilizan para representar un proceso, procedimiento, componente, etc.

2.4 Diagramas de Navegación.

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

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

2.4.1 Modelo de navegación general.

La aplicación comienza mostrando una presentación que da paso al menú principal desde donde el usuario podrá visualizar los diferentes temas que contiene el curso, el mapa conceptual, una videoconferencia y un juego de entretenimiento. Desde cualquiera de las pantallas anteriormente mencionadas el usuario podrá abandonar la aplicación visualizándose los créditos en caso de que esto ocurra.

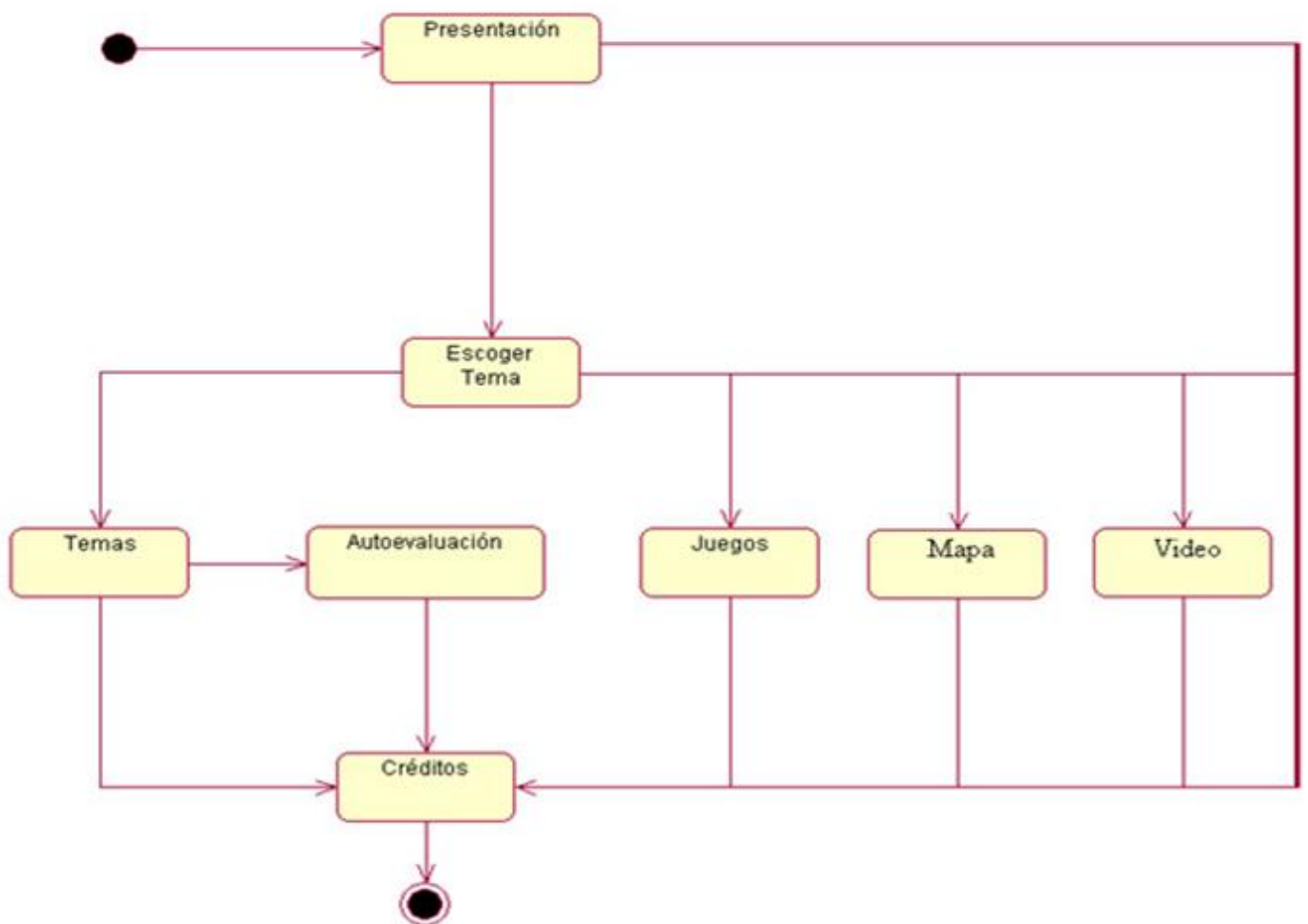


Fig. 3 Diagrama de Navegación General

2.4.1 Sub modelo de navegación Temas.

Una vez que el usuario está en la pantalla Menú Temas, el mismo podrá acceder a cualquiera de los 6 temas existentes en el curso. Cuando el usuario decide entrar a consultar un determinado tema se encuentra una pantalla donde encontrará el contenido del tema organizado en diferentes epígrafes. Además podrá encontrar palabras claves que por su significado están señaladas en rojo (palabras calientes) para llamar la atención del usuario y provocar que el mismo haga clic sobre ellas. Al hacer clic sobre una de estas palabras aparecerá una ventana flotante con información referida a esa palabra la cual podrá ser cerrada una vez que el usuario termine de visualizar dicha información haciendo clic en el botón salir que se incorpora en la esquina superior derecha.

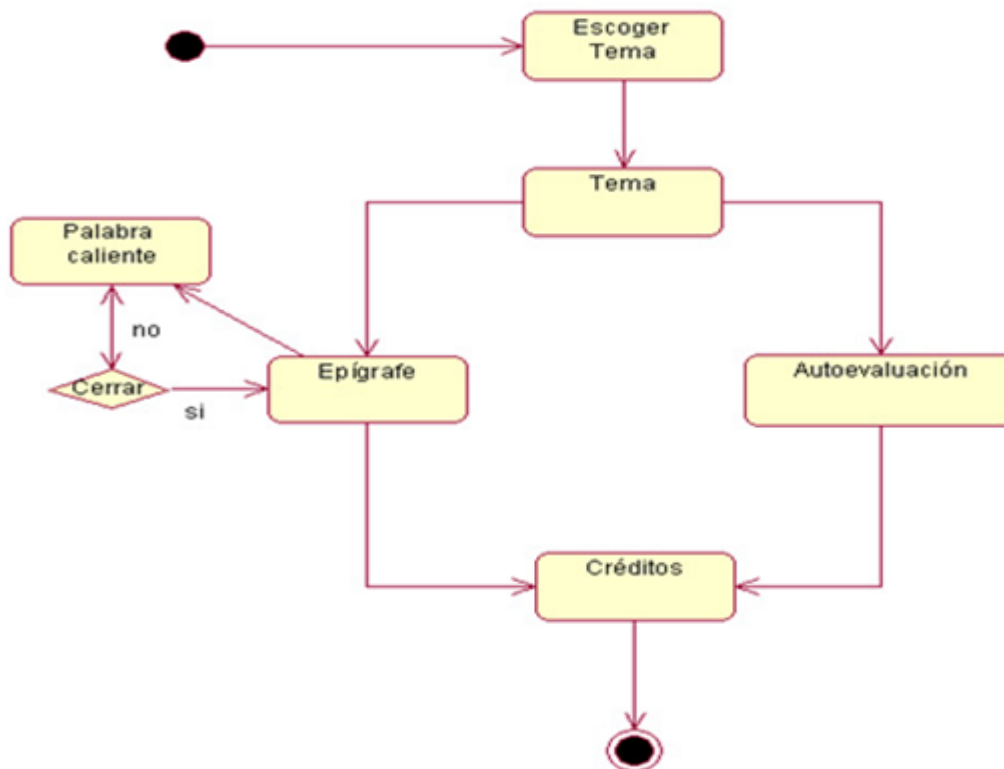


Fig. 4 Diagrama de Navegación módulo Tema.

2.5 Descripción de la funcionalidad.

Es importante definir cuales son las preferencias del cliente para diseñar un producto que se ajuste a lo que verdaderamente necesita. A continuación se muestra un listado de los requisitos funcionales y no funcionales que se han tenido en cuenta para el desarrollo de la multimedia.

2.5.1 Requisitos Funcionales.

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

Tabla 2. Requisitos Funcionales.

Referencia	Función
R1	Mostrar la pantalla de presentación de la aplicación en la cual se explicará el curso que el usuario estudiará.
R2	Mostrar fondo animado de forma automática.
R3	Reproducir música de fondo automática de forma predeterminada.
R4	Permitir al usuario parar, pausar y reanudar música de fondo.
R5	Permitir la navegación por el contenido del producto.
R6	Mostrar los temas que conforman la aplicación.
R7	Permitir al usuario conocer qué contenido encontrará dentro de cada tema.
R8	Permitir al usuario seleccionar un tema deseado.
R9	Mostrar información contenida dentro del tema seleccionado por el usuario.
R10	Permitir al usuario seleccionar los epígrafes existentes dentro de un tema.
R11	Mostrar información contenida dentro del epígrafe seleccionado por el usuario.
R12	Permitir al usuario desplazar el scroll de texto para una completa visualización de la información en aquellas pantallas que requerían el

	uso del mismo.
R13	Permitir al usuario realizar autoevaluaciones sobre los contenidos estudiados en la aplicación.
R14	Mostrar mensaje de respuesta correcta e incorrecta.
R15	Mostrar al usuario resultados obtenidos en las autoevaluaciones realizadas.
R16	Permitir al usuario ejecutar el juego que se encuentra disponible.
R17	Visualizar información referente a palabras calientes consultadas por el usuario.
R18	Permitir al usuario visualizar materiales complementarios.
R19	Visualizar video conferencia.
R20	Controlar sonido del video visualizado por el usuario.
R21	Permitir al usuario maximizar video visualizado.
R22	Permitir visualizar mapa conceptual.
R23	Mostrar concepto asociado al nodo seleccionado por el usuario en el mapa conceptual.
R24	Permitir siempre ir al menú Escoger Tema desde cualquier pantalla en que se encuentre el usuario.
R25	Permitir siempre ir a la presentación desde cualquier pantalla en que se encuentre el usuario.
R26	Permitir salir de la aplicación en cualquier momento que el usuario desee.
R27	Mostrar créditos al salir de la aplicación.

2.5.2 Requisitos no funcionales.

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

2.5.3 Requisitos de Apariencia.

- ✓ Utilizar botones que expresen su función ya sea que se intuya o expresados con texto.
- ✓ El sistema debe contener una interfaz amigable al usuario, brindándole toda las facilidades necesarias para que esta sea lo más cómoda y agradable posible.
- ✓ La aplicación debe utilizar como idioma principal el español excepto aquellas palabras técnicas que no puedan ser traducidas.
- ✓ La opción salir de la aplicación estará disponible desde cualquier parte de la aplicación, haciendo clic sobre ella, se saldrá de la misma.

2.5.4 Requisitos de usabilidad.

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

2.5.5 Requisitos de software.

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

Tabla 3. Sistemas Operativos y navegadores.

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

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

2.5.6 Requisitos de Hardware.

- ✓ Los requisitos mínimos para la ejecución de la aplicación son: Procesador Intel Pentium III de 800 MHz (o equivalente) y versiones posteriores y 128 de RAM. Pantalla de 16 bits de 1024 x 768 (se recomienda de 32 bits) y 710 MB de espacio en disco disponible.

2.5.7 Requisitos de diseño e implementación.

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

2.5.8 Requisitos de soporte.

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

2.6 Modelo de Casos de Uso del Sistema.

El modelado de casos de uso es una de las técnicas que se utilizan para modelar los requisitos del sistema. Los casos de uso, son fragmentos de funcionalidad que el sistema ofrece, para aportar un resultado de valor para sus actores permitiendo modelar el funcionamiento del sistema de la forma que el cliente desee. Utilizando las facilidades que brinda UML se pueden representar mediante un diagrama de casos de uso, las funcionalidades con que debe cumplir el sistema, que no son más que los requisitos funcionales capturados anteriormente y los actores que van a interactuar con el producto.

2.6.1 Determinación y justificación de los actores del sistema.

En la siguiente tabla se identifica el actor principal del sistema.

Tabla 4. Actor del sistema.

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

2.6.2 Diagrama de Casos de Uso.

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

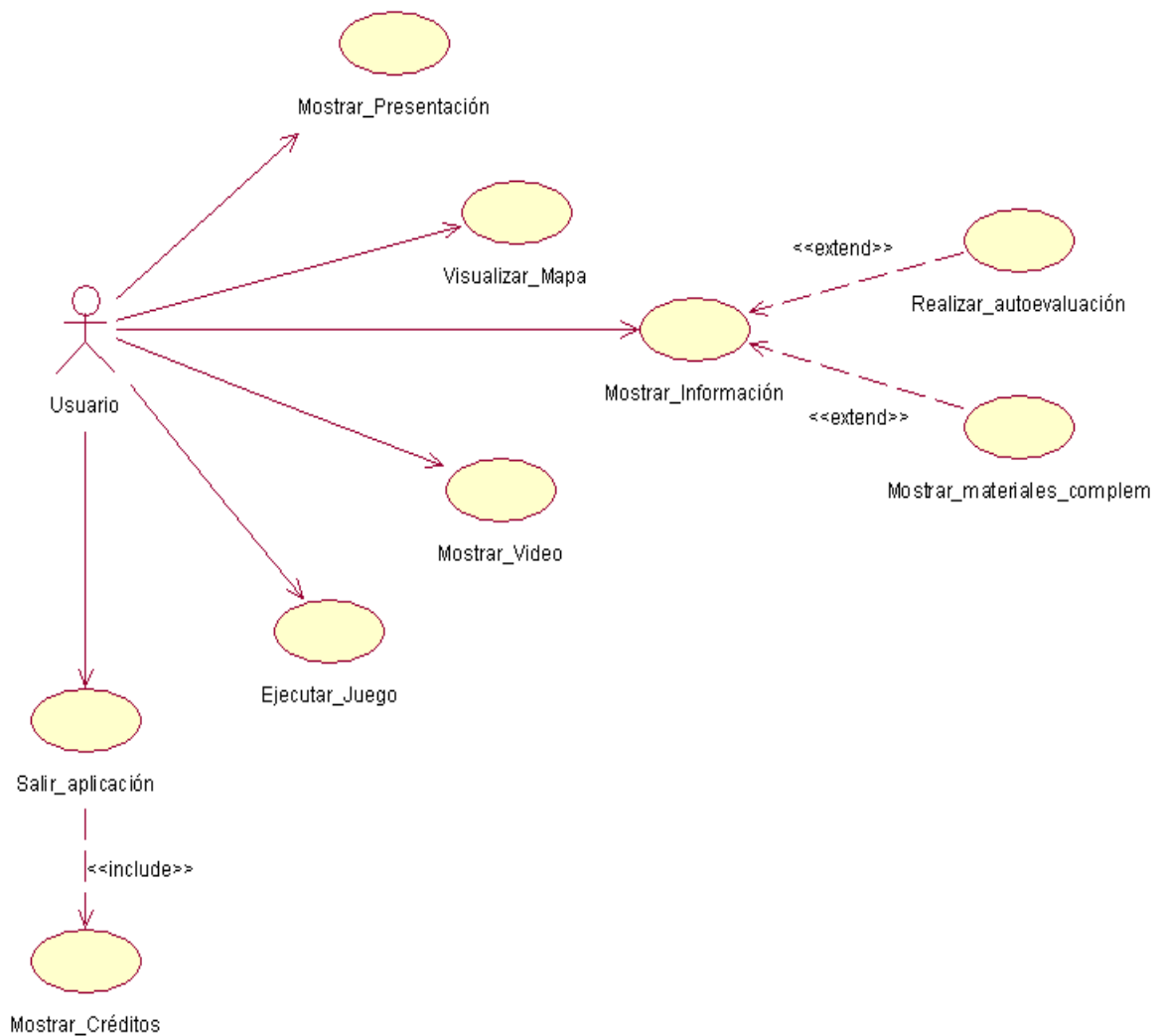


Fig. 5 Diagrama de Casos de Uso del Sistema.

2.6.3 Breve resumen de los casos de uso del sistema.

Para entender la esencia de cómo funciona cada uno de los casos de uso del sistema, se hace un pequeño resumen (Tabla 5-13) que describe la funcionalidad del caso de uso (CU) y el requisito al que da respuesta.

Tabla 5. CU Mostrar Presentación.

Caso de Uso:	Mostrar Presentación
Actor(es):	Usuario
Propósito:	Mostrar la presentación de la multimedia.
Resumen:	Al ejecutar la aplicación, se muestra la presentación del producto. Al concluir la presentación de la aplicación se pasa automáticamente al menú principal de la misma.
Referencias:	R1

Tabla 6. CU Mostrar información.

Caso de Uso:	Mostrar Información
Actor(es):	Usuario
Propósito:	Mostrar la información contenida en la multimedia. Texto, imágenes, información contenida en palabras calientes, conceptos pertenecientes al mapa conceptual, autoevaluaciones y materiales complementarios.
Resumen	El usuario de la multimedia puede seleccionar el tipo de información que desea ver. Epígrafes dentro de cada uno de los temas, conceptos pertenecientes a palabras calientes o mapa conceptual, materiales complementarios, autoevaluaciones, etc. Para poder visualizar la información que desea, tendrá que hacer clic sobre los botones que contienen la información deseada y el sistema se encargará de obtener y mostrar la información, la cual es cargada dinámicamente desde diferentes archivos XML.
Referencias:	R5, R6, R7, R8, R9, R10, R11, R17, R18, R23

Tabla 7. CU Mostrar Video.

Caso de Uso:	Mostrar Videos
Actor(es):	Usuario
Propósito:	Mostrar la videoconferencia contenida en la multimedia.
Resumen	El caso de uso se inicia cuando el usuario decide visualizar la videoconferencia contenida en la aplicación. El usuario se dirige al botón que da la opción de visualizar la misma y con sólo hacer clic, se muestra la pantalla en la cual se verá dicho video y desde donde podrá realizar diferentes actividades como parar, pausar, reproducir, subir o bajar el volumen del mismo.
Referencias:	R19, R20, R21

Tabla 8. CU Realizar Autoevaluación.

Caso de Uso:	Realizar Autoevaluación
Actor(es):	Usuario
Propósito:	Mostrar y ejecutar los ejercicios de autoevaluación de los temas estudiados en el curso.
Resumen	El caso de uso se inicia cuando el usuario después de haber consultado un determinado tema decide realizar una autoevaluación del tema estudiado haciendo clic sobre el botón Autoevaluación. Seguidamente se muestra en pantalla un cuestionario que el usuario deberá responder haciendo clic sobre la respuesta que considere correcta. Para este cuestionario se utilizarán preguntas de verdadero y falso o de selección simple. De manera automática la aplicación irá pasando de una pregunta a otra hasta llegar al final del cuestionario. Una vez concluido el mismo, el usuario podrá visualizar el porcentaje de respuestas que fueron aceptadas.
Referencias:	R13, R14, R15

Tabla 9. CU Ejecutar juego.

Caso de Uso:	Ejecutar juego
Actor(es):	Usuario
Propósito:	Ejecutar el juego disponible en la aplicación.
Resumen	El caso de uso comienza cuando el actor selecciona la opción de juego en el menú principal. En ese momento aparecerá el juego del ahorcado el cual consiste en descubrir una palabra relacionada con el tema que se trata en el curso. El juego irá guardando en la memoria de la máquina, un historial que será con valores positivos o negativos en dependencia de la cantidad de aciertos o fallas con respecto a las palabras a adivinar.
Referencias:	R16

Tabla 10. CU Visualizar mapa.

Caso de Uso:	Visualizar Mapa
Propósito:	Mostrar el mapa conceptual que contienen los principales conceptos del curso de estudio.
Actor(es):	Usuario
Resumen:	El caso de uso comienza cuando el actor hace clic sobre el botón Mapa. Automáticamente se carga una pantalla que contiene el mapa conceptual. El usuario se puede desplazar por el mismo, e ir viendo cada uno de los conceptos que en el mismo muestra a través de nodos interrelacionados. Haciendo clic sobre un nodo podrá visualizar en una ventana flotante el concepto del mismo.
Referencias:	R22, R23

Tabla 11. CU Mostrar materiales complementarios.

Caso de Uso:	Mostrar materiales complementarios.
Actor(es):	Usuario
Propósito:	Mostrar materiales complementarios que ayuden a una mejor comprensión de un determinado tema.
Resumen:	El caso de uso comienza cuando el actor después de haber consultado información referente a un tema determinado, decide visualizar un documento que la aplicación le aconseja que visualice para una mejor comprensión del tema que se trata.
Referencias:	R18

Tabla 12. CU Salir de la aplicación.

Caso de Uso:	Salir de la aplicación.
Actor(es):	Usuario
Propósito:	Permitir salir de la aplicación en cualquier momento deseado y desde cualquier lugar de la multimedia.
Resumen:	El usuario de la multimedia puede seleccionar salir de la aplicación en el momento que lo desee. Cuando el usuario seleccione el botón de salir, la aplicación le mostrará un mensaje de confirmación y el usuario puede confirmar o retractarse de salir. Una vez confirmada la salida, se mostrarán los créditos.
Referencias:	R26

Tabla 13. CU Mostrar créditos.

Caso de Uso:	Mostrar Créditos
Actor(es):	Usuario
Propósito:	Mostrar los créditos de la aplicación una vez que se decida salir de la misma.
Resumen	El caso de uso comienza cuando el actor decide abandonar la aplicación desde cualquier lugar en que se encuentre en la misma, haciendo clic sobre el botón Cerrar ubicado en la parte superior derecha de la aplicación. En ese momento aparece una pantalla indicando al actor si realmente desea abandonar la aplicación. Si este decide hacerlo entonces se mostrará otra pantalla con los créditos.
Referencias:	R27

Conclusiones

Con la culminación del presente capítulo han quedado definidos los principales conceptos relacionados con el negocio que se estudió, los cuales fueron utilizados para la creación del Modelo de Dominio, así como los requisitos funcionales y no funcionales que debe tener el producto que se realizará. De esta forma ya se puede proceder al diseño e implementación de la aplicación dando así cumplimiento al objetivo general de este trabajo.

Capítulo 3: SOLUCIÓN DE LA DESCRIPCIÓN PROPUESTA.

Introducción

En el capítulo que comienza a continuación se analizan los principios de diseños que se tendrán en cuenta para la creación del producto multimedia en cuestión. Como parte de OMMMA – L se muestran los Diagramas de Presentación, mediante los cuales se hace una ubicación espacial de todos los elementos que conforman una aplicación con tecnología multimedia. Lo que se conoce también como escenas de un guión o escenarios de interacción con el usuario. También se muestran como parte de la implementación del producto los Diagramas de Componentes y se hace una descripción de la estructuración de los archivos XML utilizados para mostrar la información que es presentada en el producto. Para finalizar se identifican de que forma se relacionan los nodos (ordenador del usuario) y procesadores (servidor) mediante el Modelo de Despliegue.

3.1 Análisis de la arquitectura de la información utilizada.

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

3.1.1 Principios y normas de diseño.

En la elaboración de un sistema el diseño de interfaces de usuario es un aspecto fundamental pues es lo que motiva al usuario a interactuar con dicho sistema. El éxito de un sistema depende de la calidad de las interfaces.

Existen tres reglas de oro para el desarrollo de la interfaz según (PRESSMAN, 2002)

1. Dar el control al usuario.
2. Reducir la carga de memoria del usuario.
3. Construir una interfaz consecuente.

Dado que la interfaz es la que da forma a la percepción del software por parte del usuario, tiene que estar bien diseñada.

- ✓ El usuario interactúa con el sistema por medio de una interfaz gráfica por tal razón la misma debe ser amena, amigable, agradable, debe presentar facilidad de uso logrando que el usuario se sienta identificado con ella.
- ✓ Las principales interfaces de la aplicación, ya sean la presentación, menús, opciones de menú, se encontrarán a pantalla completa con una resolución de 800x600 píxeles.
- ✓ Para el diseño de la interfaz de usuario hay que tener en cuenta aspectos tan importantes como lo son: la organización de la información, y la forma en que va ha estar distribuida en la pantalla.
- ✓ Las pantallas del sistema contienen la información necesaria para evitar la sobrecarga haciendo que la navegación sea más rápida.

Como la aplicación que se está desarrollando es de enseñanza aprendizaje debe desarrollarse con colores claros que resulten agradables a la vista haciendo marcado uso del color azul por ser un color muy identificado con la UCI.

- ✓ La utilización de las fuentes (textos), mantendrán un estilo sencillo y acorde a los colores a utilizar en cada unas de las pantallas donde se mostrarán los textos, teniendo en cuenta que el tipo de letra a utilizar (Verdana 10) permitirá al estudiante ver con mayor claridad y eficiencia las letras del texto.
- ✓ Se repite el estilo de todas las pantallas creando consistencia. Se relacionan elementos de una manera determinada en un área y esta tendencia es repetida en otras áreas.
- ✓ Con el objetivo de que el sistema sea más atractivo al usuario se le ha incorporado sonido, el mismo puede ser activado o desactivado cuando el usuario lo solicite.

3.1.2 Estándares para el diseño de la interfaz del usuario multimedia.

El diseño de interfaces de usuario para aplicaciones multimedia comprende, en general, a un conjunto de elementos de diseño y evaluación mucho más amplio que el asociado a interfaces de usuario convencionales, basadas únicamente en texto y gráficos. Se dispone de diferentes técnicas y opciones de diseño. Las interfaces de usuario multimedia incorporan, integran y sincronizan distintos medios (medios estáticos, tales como texto, gráficos e imágenes y medios dinámicos, tales como sonido, animaciones, video u otras modalidades sensoriales), pudiendo ampliarse, para cada uno de estos medios, la relación de medios ya incluidos.

Algunos de los estándares para el diseño de la interfaz del usuario multimedia que se tienen en cuenta son:

1. Controlabilidad:

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

2. Conformidad con las expectativas del usuario:

Los usuarios son capaces de visualizar el video que deseen así como controlarlo, es decir que pueden controlar la puesta en marcha, pararlo o iniciarlo nuevamente, sin tener que ir nuevamente al botón que los visualiza.

3. Tolerancia al error:

Si un video ha sido parado por el usuario, puede ser puesto en funcionamiento de nuevo en la misma posición, de forma que el usuario no tenga que volver al principio, para ello se utiliza un botón pause y un botón play, que permite poner en funcionamiento nuevamente al video en la misma posición que estaba antes.

4. Apoyo de la orientación del usuario:

El usuario puede controlar su posición ya que el título de cada epígrafe seleccionado del tema, se presenta encima del contenido, orientándole al usuario el contenido que está viendo en ese momento.

3.1.3 Estándares de codificación.

Los estándares de codificación son reglas específicas a una lengua que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Con el objetivo de que este trabajo pueda someterse a cambios en el futuro, realizados por cualquier otro programador y teniendo en cuenta que no existe nada escrito acerca de las técnicas de programación a utilizar por aquellos que utilizan ActionScript, se utilizan las siguientes convenciones generales que ayudan a implementar el código de una forma más rápida y eficiente.

Elegir nombres representativos para las variables, funciones e instancias:

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

ventanacerrar_btn

inicio_btn

Utilizar sufijos para nombres de instancia:

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

Button: salir_btn.

MovieClip: principal_mc

Comentar el código:

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

```
//Botón Epígrafe 1
```

```
epig1_mc.mc_1.b1_btn.onRelease = function()
```

```
{
```

```
//se carga desde el nodo 2 del XML el nombre del epígrafe que se visualiza pinchando el botón 1
```

```
//se carga desde el nodo 7 del XML la información correspondiente al epígrafe 1 que se visualiza  
//pinchando el botón 1
```

```
this._parent.nombre_epig_txt.htmlText = tema1_XML.firstChild.childNodes[2].toString();
```

```
    this._parent.mitexto.htmlText = tema1_XML.firstChild.childNodes[7].toString();
```

```
    _parent.intro_btn._visible = true;
```

```
}
```

3.2 Descripción de archivos XML.

El desarrollo de la aplicación está basado en la utilización del metalenguaje XML para el almacenamiento de la información (textos e imágenes), permitiendo la realización de modificaciones sin necesidad de interactuar con el diseño de la aplicación. En la aplicación existen 4 estructuras principales de archivos

XML. Los archivos identificados como Tema# (donde # se corresponde con el número de cada tema), Escoger_Tema, Juegos y Autoevaluaciones

Descripción de archivos XML Tema

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

```
<tema>
  <nombreTema><p align='center'><i><b>Nombre del tema</b></i>
</p></nombreTema>
  <epígrafe1><b><i>Nombre del epígrafe 1</i></b> </epígrafe1>
  .
  <epígrafe_n><b><i>Nombre del epígrafe n</i></b> </epígrafe_n>
  <texto_epígrafe1> Texto que pertenece al epígrafe 1 del tema</texto_epígrafe1>
  .
  .
  <texto_epígrafe_n> Texto que pertenece al epígrafe 1 del tema</texto_epígrafe_n>
</tema>
```

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

Descripción de archivos XML Escoger_Tema

```

<escogerTema>

    <bienvenida><p align='center'><font color='#000000' size='22'><b> Pruebas de
Software</b></font></p>

Bienvenido al curso de Pruebas de Software !!! </bienvenida>

<tema1><p align='center'><font color='#FF0000' size='22'><b> Nombre del tema 1
</b></font></p><font size='15'>

Se explica al usuario el contenido que encontrará dentro del tema y se le explica que al
terminar puede realizar una autoevaluación.

</font></tema1>

.

<tema_n><p align='center'><font color='#FF0000' size='22'><b> Nombre del tema n
</b></font></p><font size='15'>

Se explica al usuario el contenido que encontrará dentro del tema y se le explica que al
terminar puede realizar una autoevaluación sobre lo aprendido en el mismo.

</font></tema_n>

</escogerTema>

```

Este XML está estructurado por un nodo padre, un nodo hijo que contendrá una breve información de bienvenida al curso y n nodos que contendrán información sobre lo que el usuario encontrará en cada uno de los temas que existen en el curso, así como una invitación a realizar la autoevaluación una vez concluido el estudio del tema.

Descripción de archivos XML Autoevaluación

```
<quiz>

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

  <items>

    <item>

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

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

      <answer>Verdadero</answer>

    </item>

    <item>

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

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

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

    </item>

  </items>

</quiz>
```

El XML está estructurado de la siguiente manera:

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

</quiz> cierre de nodo.

<title> nombre del título de la evaluación.

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

</ítems> cierre de nodo.

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

</ítem> cierre de nodo.

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

</question> cierre de nodo.

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

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

</answer> cierre de nodo.

3.3 Diagramas de Presentación del modelo de Análisis.

A través de los diagramas de presentación (Figura 6-11), también conocidos como escenas de un guión o escenarios de interacción con el usuario, se hace una ubicación espacial de todos los elementos que conforman una aplicación con tecnología multimedia.

Diagrama de Presentación Pantalla Escoger Tema

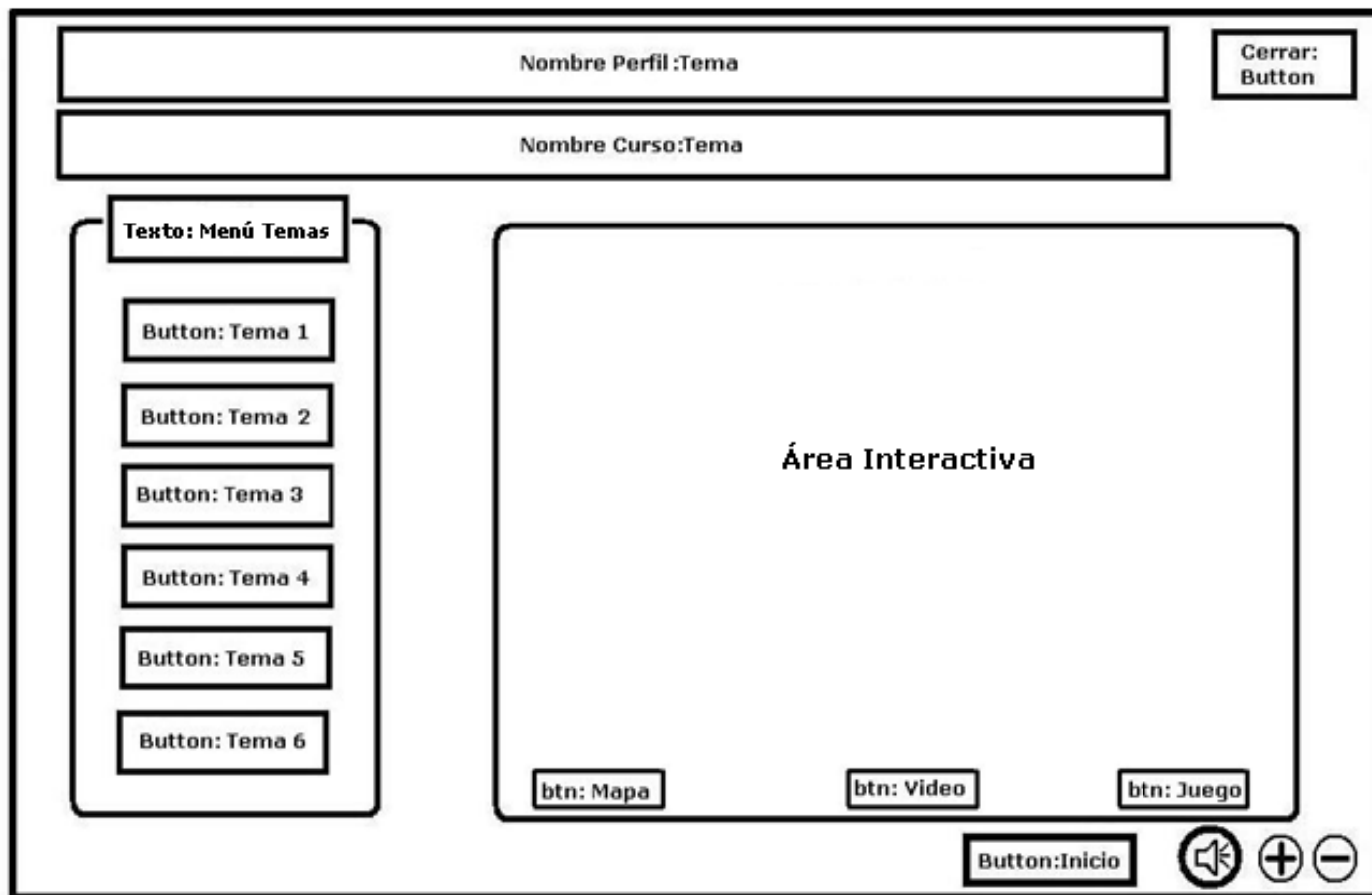


Fig. 6 Diagrama de Presentación Pantalla Escoger Tema

Diagrama de Presentación Tema

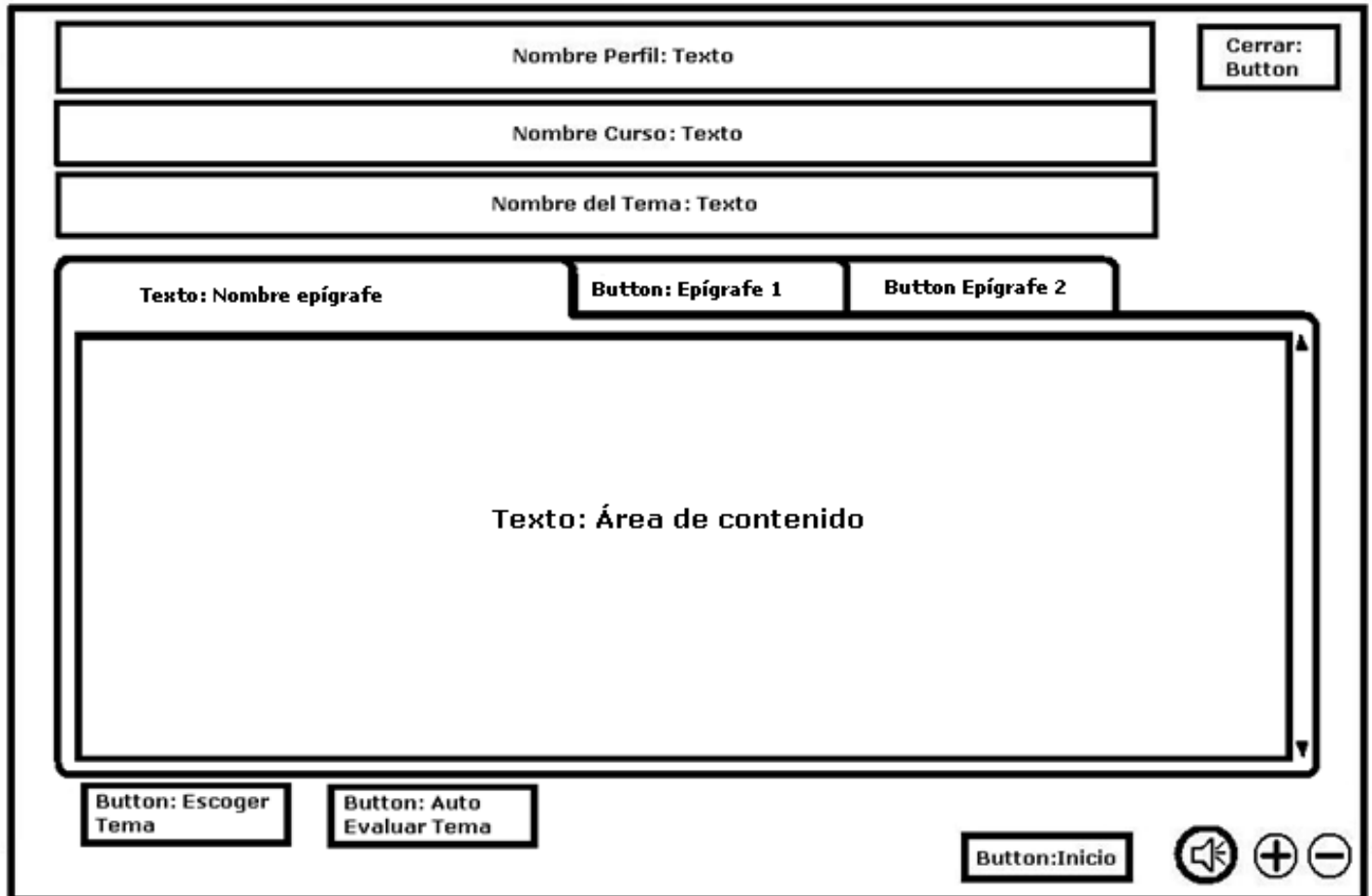


Fig. 7 Diagrama de Presentación Tema

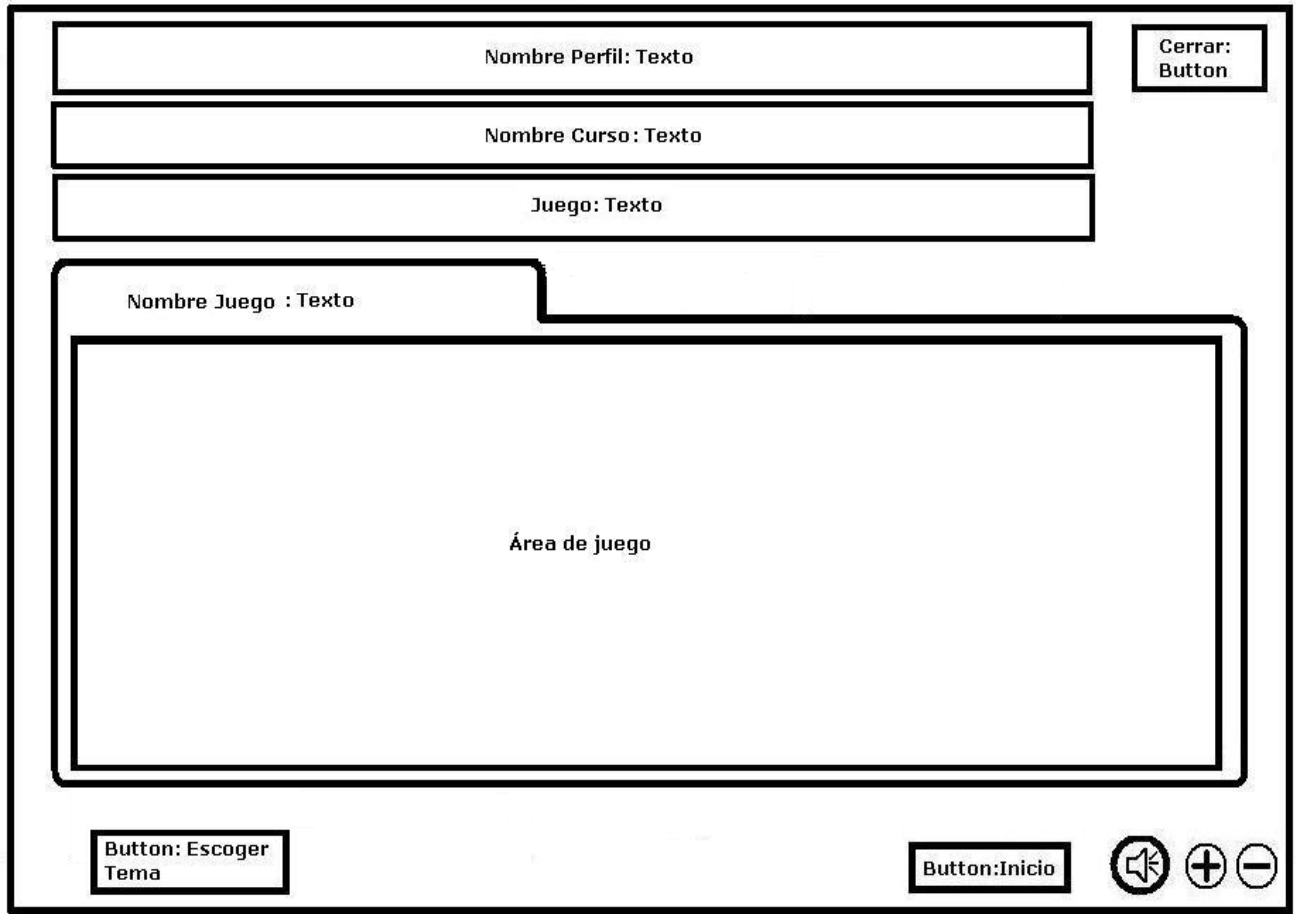
Diagrama de Presentación Juegos**Fig. 8 Diagrama de Presentación Juegos**

Diagrama de Presentación Autoevaluación

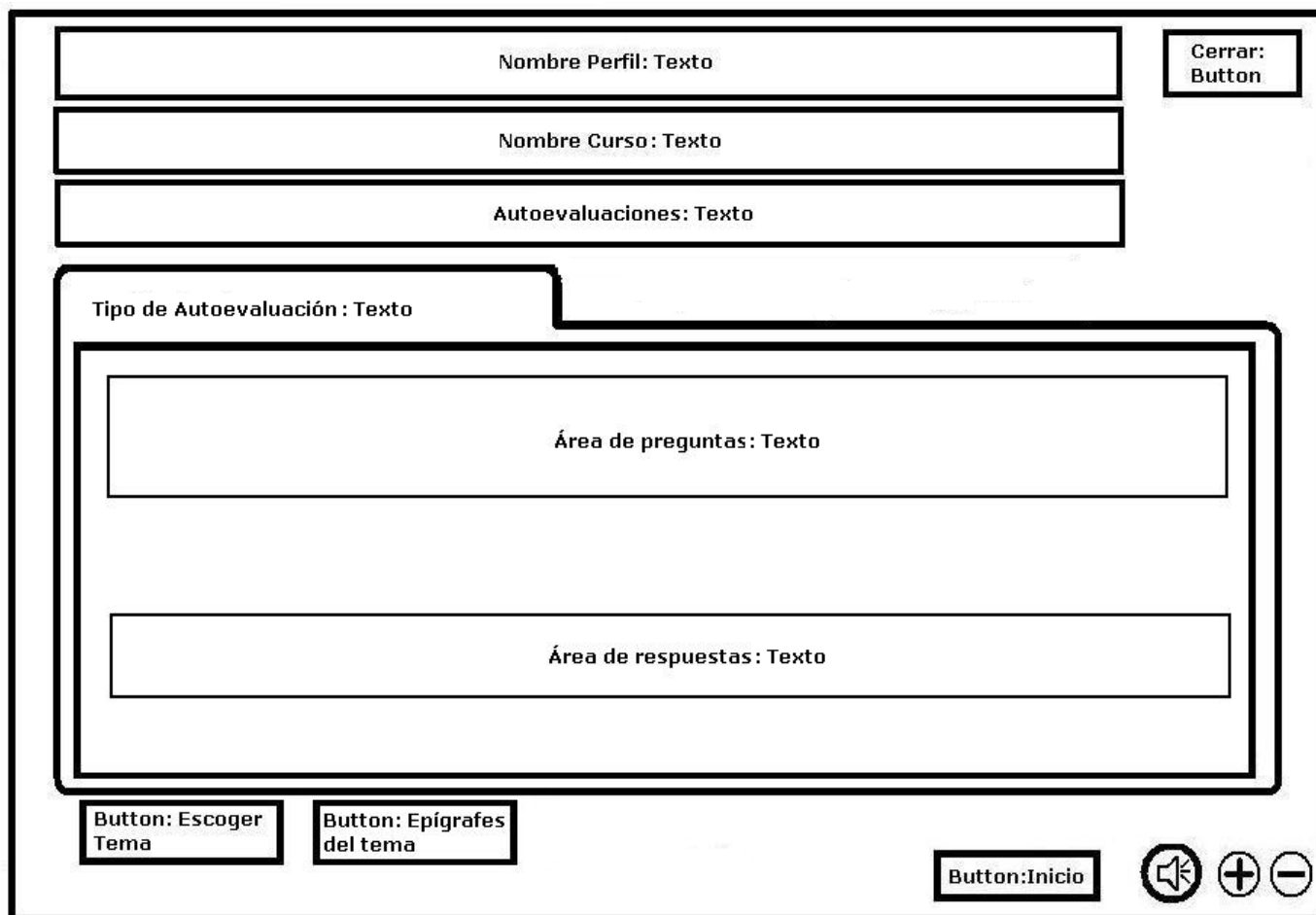


Fig. 9 Diagrama de Presentación Autoevaluación

Diagrama Presentación Pantalla Salir

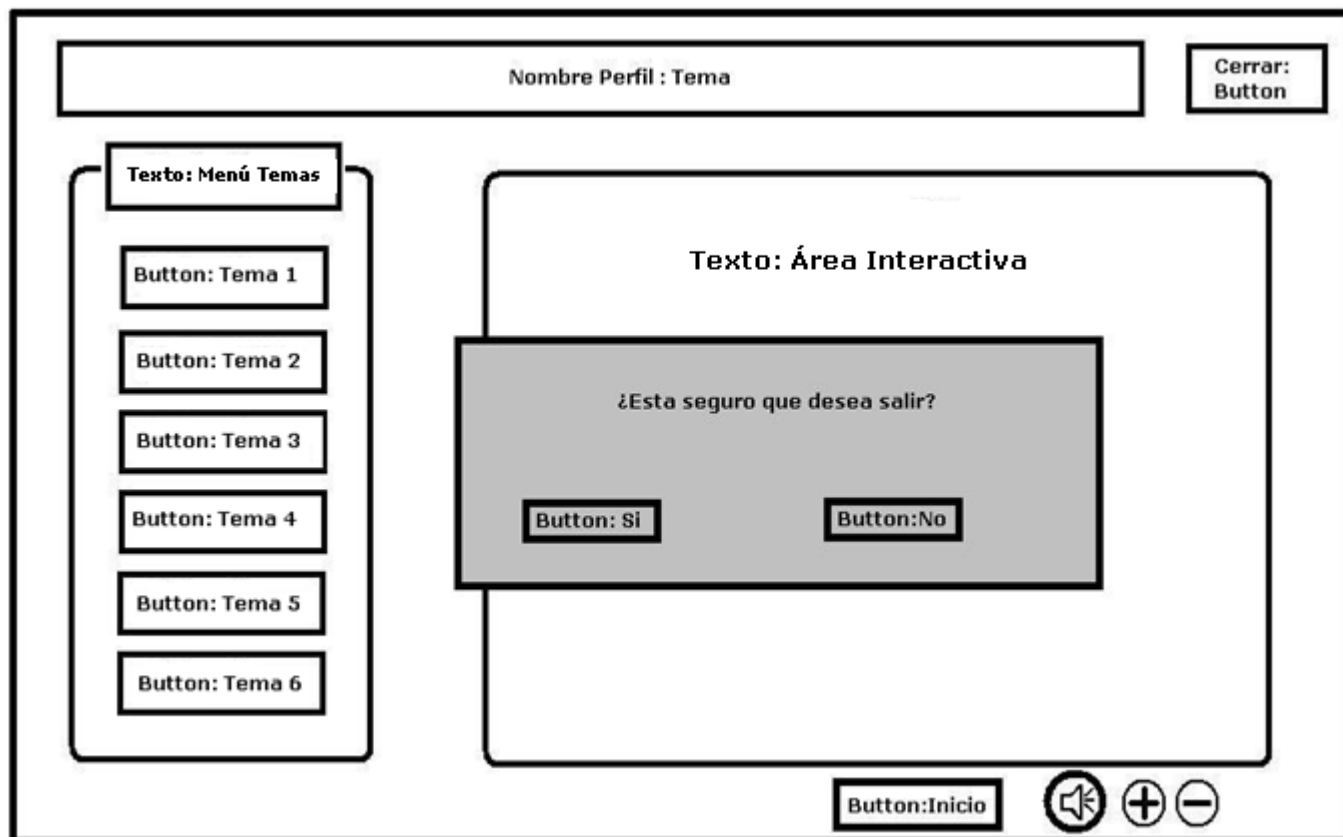


Fig. 10 Diagrama Presentación Pantalla Salir

Diagrama de Presentación Videos

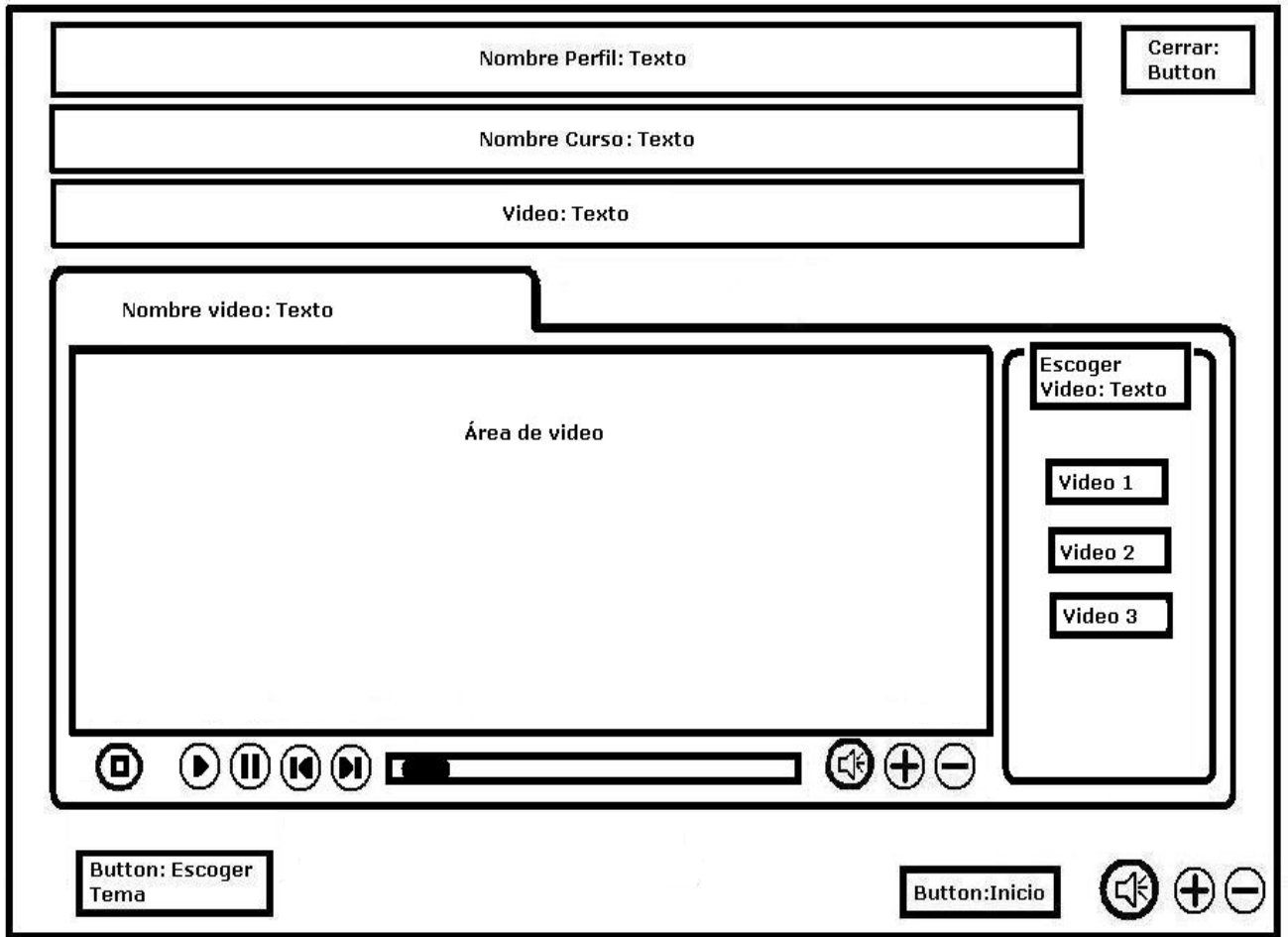


Fig. 11 Diagrama de Presentación Videos

3.4 Diagrama de Componentes.

El Diagrama de Componentes se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables. En el Diagrama de componentes que se muestra a continuación también se representan los

archivos XML. La tecnología XML fue empleada para mostrar la información del producto de forma dinámica.

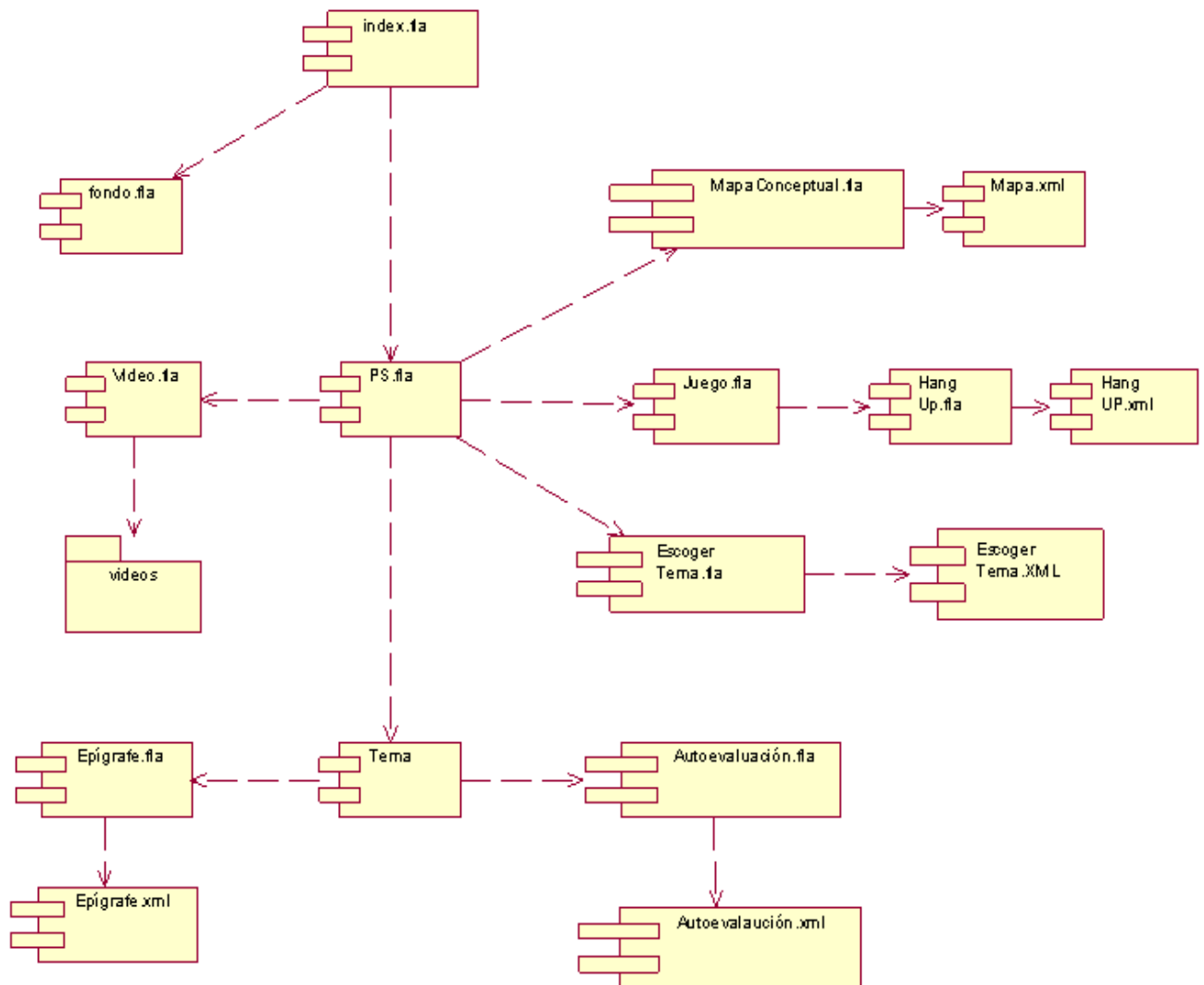


Fig. 12 Diagrama de Componentes General

3.5 Modelo de Despliegue

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



Fig. 13 Modelo de Despliegue

Conclusiones

En este capítulo que concluye se han desarrollado una serie de artefactos con el objetivo de finalizar la modelación producto. Se ha utilizado el Lenguaje Unificado de Modelado (UML) haciendo alusión a las vistas que han resultado necesarias, como también se ha utilizado el Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA-L) con el fin de representar la distribución espacial de media, la cual se contempla en un nuevo artefacto propuesto por este lenguaje, el diagrama de presentación. Para mejorar la comprensión del modelo de implementación ambos lenguajes han servido de apoyo para representar el flujo de implementación mediante el diagrama de componentes, teniéndolos en cuenta además para el desarrollo del modelo de despliegue.

CONCLUSIONES GENERALES

Después de haber realizado un exhaustivo estudio para el desarrollo de esta aplicación y haber concluido las fases de elaboración y construcción del producto, se ha determinado lo siguiente:

- ✓ En el presente trabajo de diploma se desarrolló un producto multimedia interactivo sobre pruebas de software. Durante todo el proceso de desarrollo del producto se utilizó como metodología de desarrollo de software RUP (Proceso Unificado de Desarrollo del Software) y como lenguaje de modelado para multimedia OMMMA-L (Lenguaje Orientado Objetos para Aplicaciones Multimedia), siendo este una extensión del conocido lenguaje de modelado UML (Lenguaje Unificado de Modelado).
- ✓ Para el desarrollo del producto final se utilizó la herramienta de creación de aplicaciones con tecnología multimedia Macromedia Flash 8 por ser una herramienta de fácil uso que genera archivos multiplataforma que pueden ser visualizados con cualquier navegador.
- ✓ La aplicación cuenta con una interfaz amigable y de fácil uso que integra la utilización del lenguaje XML para gestionar y agrupar los datos en volúmenes compactos de información, quedando reflejado el vínculo estrecho existente entre XML y la tecnología multimedia.
- ✓ Se realizó la implementación de una multimedia que contiene un mapa conceptual y diversos medios (videos, juegos y autoevaluaciones) asociados a la misma que mejoran el proceso de enseñanza aprendizaje de pruebas de software.

La multimedia interactiva Pruebas de Software cuenta con un contenido acorde a lo que se imparte en la Universidad referente a pruebas de software, lo que puede significar una excelente alternativa para aumentar la velocidad y calidad del proceso de aprendizaje en el curso optativo Pruebas de Software, posibilitando o adecuando escenarios variados, dinámicos o con animación, que hacen que el alumno se

muestre activo y motivado durante su uso y como consecuencia de ello, se sienta cómodo practicando lo aprendido.

RECOMENDACIONES

Se recomienda:

- ✓ Poner a disposición de los estudiantes la Multimedia Interactiva Pruebas de Software, e introducirla como material de apoyo al curso optativo Pruebas de Software impartido por la Dirección de Calidad de Software en la UCI.
- ✓ Realizar entrevistas a especialistas de pruebas de software donde se traten importantes aspectos del tema y que las mismas sean grabadas en formato de video para ser incorporadas posteriormente a la galería de videos de la aplicación.
- ✓ Integrar en un único producto las aplicaciones multimedia: Pruebas de Software, Introducción a CMMI, Introducción al Proceso de Desarrollo de Software y Calidad y Curso Ingeniería de Requisitos, con el objetivo de que los estudiantes puedan acceder desde una misma aplicación a cualquiera de los cursos que el mismo desee consultar sin necesidad de tener que ejecutar diferentes aplicaciones.

REFERENCIAS BIBLIOGRÁFICAS

ALEGSA. 2008. Alegsa.com. [En línea] 2008. [Citado el: 13 de Diciembre de 2007.] <http://www.alegsa.com.ar/Dic/authorware.php..>

AULACLIC. 2006. Introducción a Flash 8 (I), De Flash MX 2004 a Flash 8. [En línea] 2006. [Citado el: 25 de Marzo de 2008.] http://www.aulaclic.es/flash8/t_1_1.htm.

BARROS, D. R. 2002. Conceptos generales sobre sistemas hipermediales. [En línea] 2002. [Citado el: 13 de Diciembre de 2007.] <http://www.america.farq.unr.edu.ar/el/data/epapers/+pdf/SH%20%20Conceptos%20generales.pdf>.

BELLONCH, J. V. 2006. MetaCard y Revolution: Herramientas de autor. [En línea] 2006. [Citado el: 16 de Diciembre de 2007.] <http://www.disca.upv.es/magustim/mmmultiplataforma/c154.htm>.

BIRKДАР, N. 2003. Macromedia Director MX nombrado la mejor solución de autor multimedia por la revista Presentations. [En línea] 2003. [Citado el: 12 de Febrero de 2008.] http://www.adobe.com/es/macromedia/proom/2003/01_03/dir_best.html .

BNM. 2008. Biblioteca Nacional de Maestros. [En línea] 2008. [Citado el: 25 de Enero de 2008.] <http://www.me.gov.ar/bnm/>.

BRUNO, Leighton. 2005. Una propuesta Metodológica para el Diseño de Interfaces y Mapas de Navegación en Aplicaciones Hipermediales. [En línea] 2005. [Citado el: 8 de Enero de 2008.] http://www.inf.udec.cl/revista/edicion4/paper_metodologia.htm..

CANÓS, José H. y P. L., M^a Carmen Penadés. 2002. (XP) Metodologías Ágiles en el Desarrollo de Software. [En línea] 2002. [Citado el: 13 de Diciembre de 2007.] <http://www.willydev.net/descargas/prev/TodoAgil.Pdf..>

CIBERAULA. 2005. Actionscript y XML. [En línea] 2005. [Citado el: 16 de Enero de 2008.] http://flash.ciberaula.com/articulo/actionscript_xml/ .

COMPANIONI, R. D. 2005. Interactuar con un software educativo. [En línea] 2005. [Citado el: 2 de Diciembre de 2007.] <http://www.monografias.com/trabajos42/interactuar-software/interactuar->

DÍAZ, Dionisio. 2000. Multimedia en la enseñanza. [En línea] 2000. [Citado el: 20 de 11 de 2007.] <http://www.dionisiodiaz.com/multimensenanza/multimenzenanza.html>.

ENGELS, G. 2001. UML-based Behavior. Specification of Interactive multimedia. [En línea] 2001. [Citado el: 7 de Enero de 2008.] <http://wwwcs.upb.de/cs/ag-engels/Papers/2001/Sauer HCC01.pdf> .

FEBLES, Ailín. 2001. *Case Corporativo para el proceso de control de cambios. Tesis presentada en opción al título de Máster en Informática Aplicada.* 2001.

FEBLES, Ailin. 2004. *Modelo de referencia para la Gestión de Configuración en la pequeña y mediana empresa de software.* 2004.

FEBLES, Ailín. 2006. *UCI, Calidad del software.* Ciudad de la Habana : s.n., 2006.

FERNÁNDEZ, J. M. 2005. La plataforma educativa Moodle. [En línea] 2005. [Citado el: 29 de 11 de 2007.] <http://www.eurolegends.org/moodle/file.php/48/recursos/intro.htm>.

FREDDIE, C. 2004. Tutorial de XML en Flash MX 2004. [En línea] 2004. [Citado el: 21 de Enero de 2008.] <http://www.cristalab.com/tutoriales/12/tutorial-de-xml-en-flash-mx-2004>.

GÓNZÁLEZ, B. 2005. XML: el lenguaje de los Servicios Web. [En línea] 2005. [Citado el: 15 de Enero de 2008.] <http://www.desarrolloweb.com/articulos/1574.php>.

GÓNZALEZ, Michael. 2008. *Proceso de prueba para la liberación de productos de software.* 2008.

GRANJA, Juan Carlos. 2007. *Métricas, técnicas del software.Auditoría.* 2007.

GUIZA, Ezkauriatza. 2005. Multimedia como herramienta en la educación. [En línea] 2005. http://www.congresoretosyexpectativas.udg.mx/Congreso%203/Mesa%205/Mesa5_18..pdf.

HENST, C. V. D. 1999. Flash, la tecnología multimedia para el web. [En línea] 1999. [Citado el: 15 de Diciembre de 2007.] <http://www.maestrosdelweb.com/editorial/flash/>.

IEEE. 1990. *Standard 610, Computer Dictionary*. 1990.

INQALABS. 2007. [En línea] 2007. <http://www.inqalabs.com/eng/index.html>.

INTECO. 2008. Laboratorio Nacional de Calidad de España. [En línea] 2008. http://www.inteco.es/Calidad_del_Software/Formacion_4.

JACOBSON, Booch Grady y Rumbaugh, James. 2000. *El proceso unificado de desarrollo de software (RUP)*. 2000.

LABAÑINO, C y Del Toro, M. 2005. *Multimedia para la educación*. s.l. : Pueblo y Educación, 2005.

LAPUENTE, M. J. 2006. *El nuevo concepto de documento en la cultura de la imagen*. 2006.

LUGO, M. F. F. N. P. 2007. *Conceptos de Gestión de Conocimiento Asociados al Proceso de Pruebas de Software en la UCI*. 2007.

MARQUÉZ, Peré. 2004. El software educativo. [En línea] 2004. [Citado el: 2 de Febrero de 2008.] http://www.lmi.ub.es/te/any96/marques_software/.

MARTÍNEZ, J.L.G. 2001. Hacia un Nuevo paradigma: el hipertexto como faceta sociocultural de la tecnología. [En línea] 2001. <http://www.ensayistas.org/critica/teoria/hipertexo/gomez>.

MORÓN, A. 2004. Multimedia en Educación. [En línea] 2004. [Citado el: 15 de Noviembre de 2007.] <http://redalyc.uaemex.mx/redalyc/pdf/158/15800311.pdf>.

MOTA, J.C. 2003. Enseñanza asistida y diseño de sitios webcon Toolbook II. [En línea] 2003. [Citado el: 12 de Diciembre de 2007.] http://alfaomega.com.mx/shopsite_sc/store/html/product204.html.

- PASCUAL, J. 2003.** Herramientas profesionales para la creación de aplicaciones multimedia. [En línea] 2003. [Citado el: 20 de Febrero de 2008.] http://www.macworld.es/pcworld/index.asp?link=estructura/i_articulo_centroArticulo.asp&IdArticulo=50218.
- PEREZ, Isabel. 2002.** *Métricas para el control de proyectos de software*. 2002.
- PRESSMAN, R. 2002.** *Ingeniería del Software: Un Enfoque Práctico*. s.l. : Mc Graw Hil, 2002.
- PRESSMAN, Roger. 2002.** *Ingeniería del Software: Un enfoque práctico*. 2002.
- PUC. 2007.** Pontificia Universidad Católica de Puerto Rico. [En línea] 2007. [Citado el: 10 de Diciembre de 2007.] http://www.pucpr.edu/pbanner/webct_info.htm.
- ROVIRA, Cristófol. 2003.** Mapas conceptuales para la representación del conocimiento. [En línea] 2003. [Citado el: 20 de Enero de 2008.] <http://www.hipertext.net/modulo7/ayudaEditor2.htm>.
- RUBINSTEIN, David. 2007.** There's Less Development Chaos Today SDTimes Software Development. [En línea] 2007. [Citado el: 28 de Noviembre de 2007.] <http://www.sdtimes.com/content/article.aspx?ArticleID=30247>.
- RUP. 2003.** *Ayuda del Rational Unified Process*. 2003.
- SÁNCHEZ, José. 1999.** *Construyendo y Aprendiendo con el Computador*. 1999.
- SWEBOK. 2004.** *Guide to the Software Engineering Body of Knowledge* . 2004.
- UDISTRAL. 2008.** Universidad Distrital Francisco José de Caldas. [En línea] 2008. [Citado el: 10 de Enero de 2008.] <http://www.udistrital.edu.co/comunidad/grupos/arquisoft/fileadmin/Estudiantes/>.
- VELÁZQUEZ, M. A. 2002.** XML como definición y manejo de datos. [En línea] 2002. [Citado el: 19 de Enero de 2008.] http://www.informatizate.net/articulos/xml_como_definicion_y_manejo_de_datos_20030808.html.

VERA, J. F. 2003. Revolution 2.0 El poder de la sencillez. [En línea] 2003. [Citado el: 2 de Marzo de 2008.] http://www.macuarium.com/actual/pruebas/2003/07/01_revolution.shtml.

WESLEY, Addison. 2002. *Una explicación de la programación extrema.* 2002.

WIKIPEDIA. 2008. Wikipedia. [En línea] 2008. [Citado el: 12 de Enero de 2008.] <http://es.wikipedia.org/wiki/XML>.

BIBLIOGRAFÍA

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

LABAÑINO, C y Del Toro, M. 2005. *Multimedia para la educación*. s.l. : Pueblo y Educación, 2005.

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

ROVIRA, Cristófol. 2003. Mapas conceptuales para la representación del conocimiento. [En línea] 2003. [Citado el: 20 de Enero de 2008.] <http://www.hipertext.net/modulo7/ayudaEditor2.htm>.

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

GLOSARIO DE TÉRMINOS

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

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

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

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

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

MVC: es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos.

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

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

Kioscos de información: son máquinas multimedia situadas en espacios públicos estratégicos, con determinado tipo de dispositivos que, mediante una aplicación, accedan datos y permiten al usuario interactuar con ellos, obteniendo, así, información.

Palabras calientes: son palabras en formato cursiva y subrayada que se encuentran en los contenidos de la aplicación donde tan sólo haciendo un clic sobre ella se puede acceder a determinada información del producto en correspondencia con la palabra que se haya seleccionado.

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

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

ANEXOS

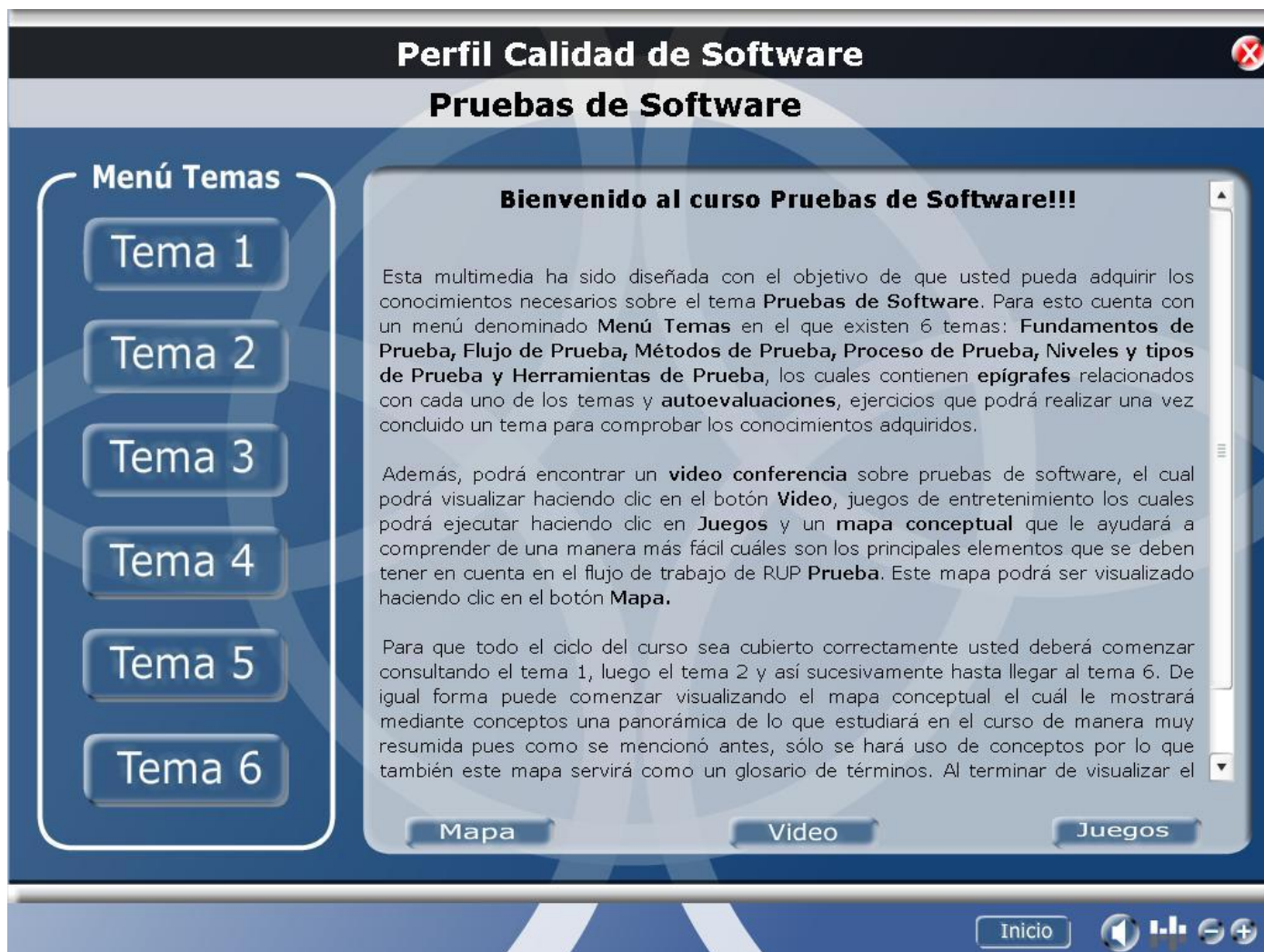


Figura 1. Pantalla de Bienvenida.

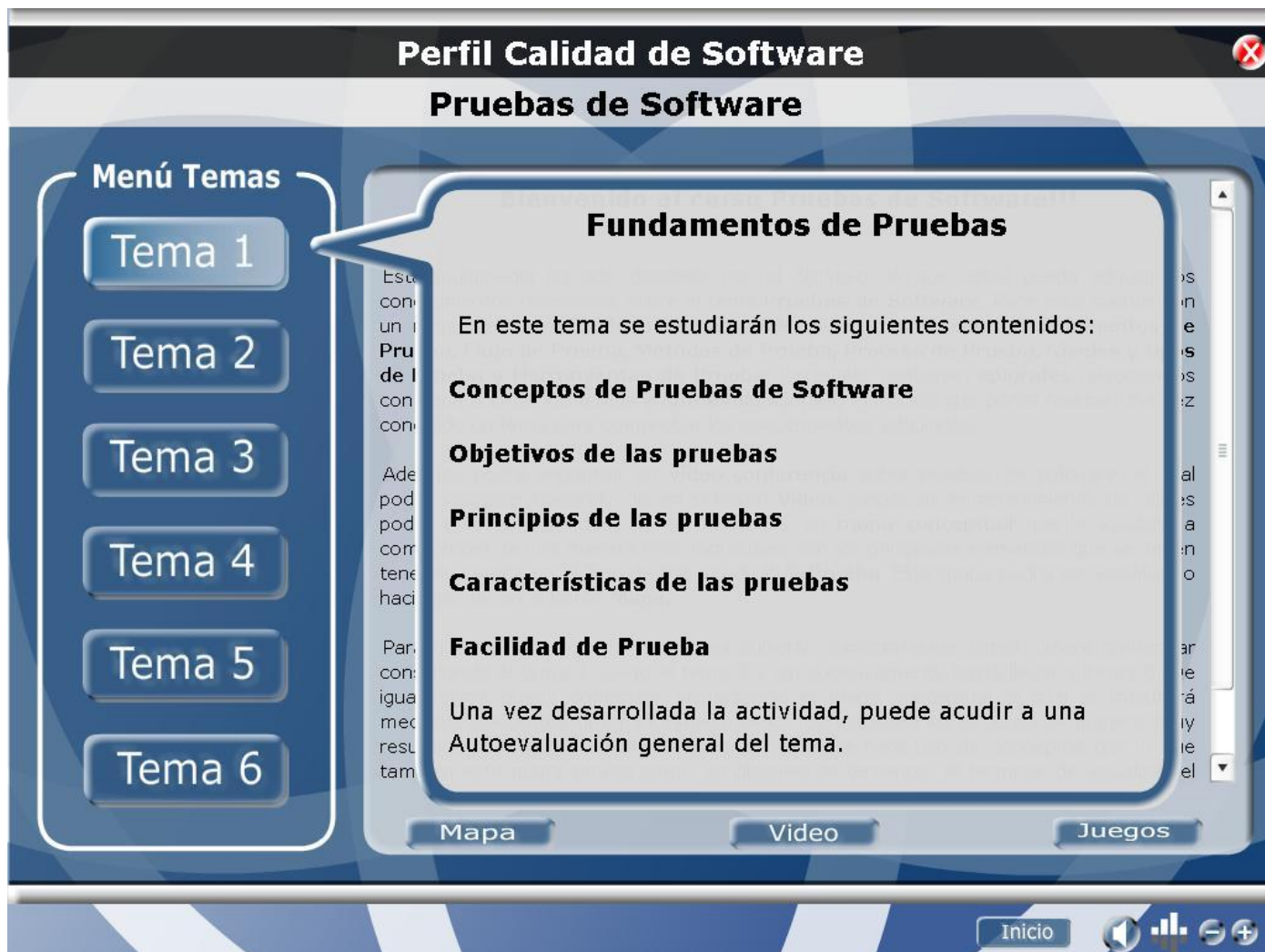


Figura 2. Pantalla Menú Principal.

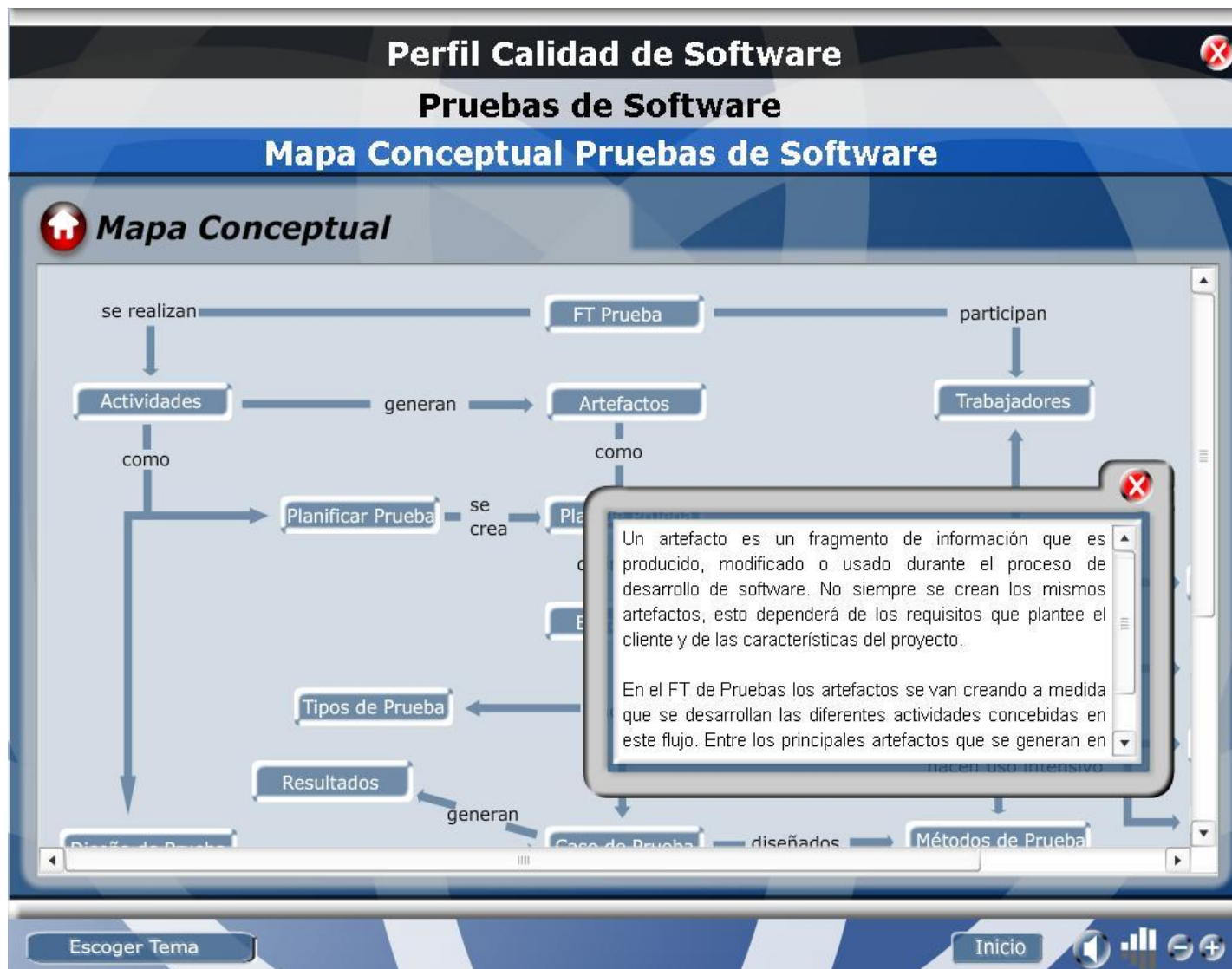


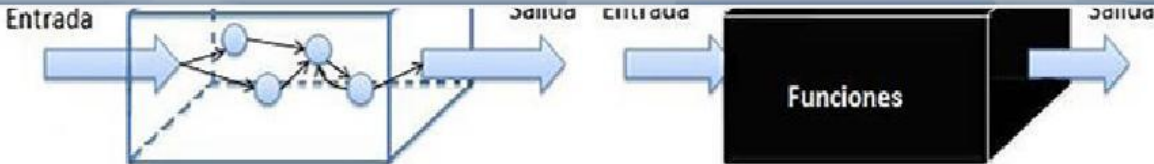
Figura 3. Pantalla Mapa Conceptual con ventana flotante.



Figura 4. Pantalla Visualización de video.

Perfil Calidad de Software
Pruebas de Software
Métodos de Pruebas de Software

Métodos de Pruebas



El diseño de pruebas para el software o para otros productos se realiza al inicio del desarrollo inicial del producto. Sin embargo los ingenieros de software a menudo proponen casos de prueba que parezcan adecuados, pero que no cubren todos los caminos lógicos del software proponiendo los casos de prueba que examinen que están correctas todas las condiciones y/o bucles para determinar si el estado real coincide con el esperado o afirmado por lo que estas pruebas requieren del conocimiento de la estructura interna del programa y son derivadas a partir de las especificaciones internas de diseño o el código. En el epígrafe Caja Blanca se abordarán con

Cualquier producto de ingeniería puede probarse de 2 maneras:

- 1-) Conociendo la función específica por la que fue diseñada, se prueba que cada función es completamente operativa y al mismo tiempo se verifica que no se produzcan errores.
- 2-) Conociendo el funcionamiento del producto, se prueba que la operación interna se ajusta a las especificaciones y que todos los componentes internos se han comprobado de forma adecuada. El primer enfoque de prueba se denomina prueba de **caja blanca** y el segundo, prueba de **caja negra** los cuales serán estudiados en los epígrafes de este tema.

Escoger Tema Autoevaluar Tema Inicio

Figura 5. Pantalla mostrar palabras calientes.

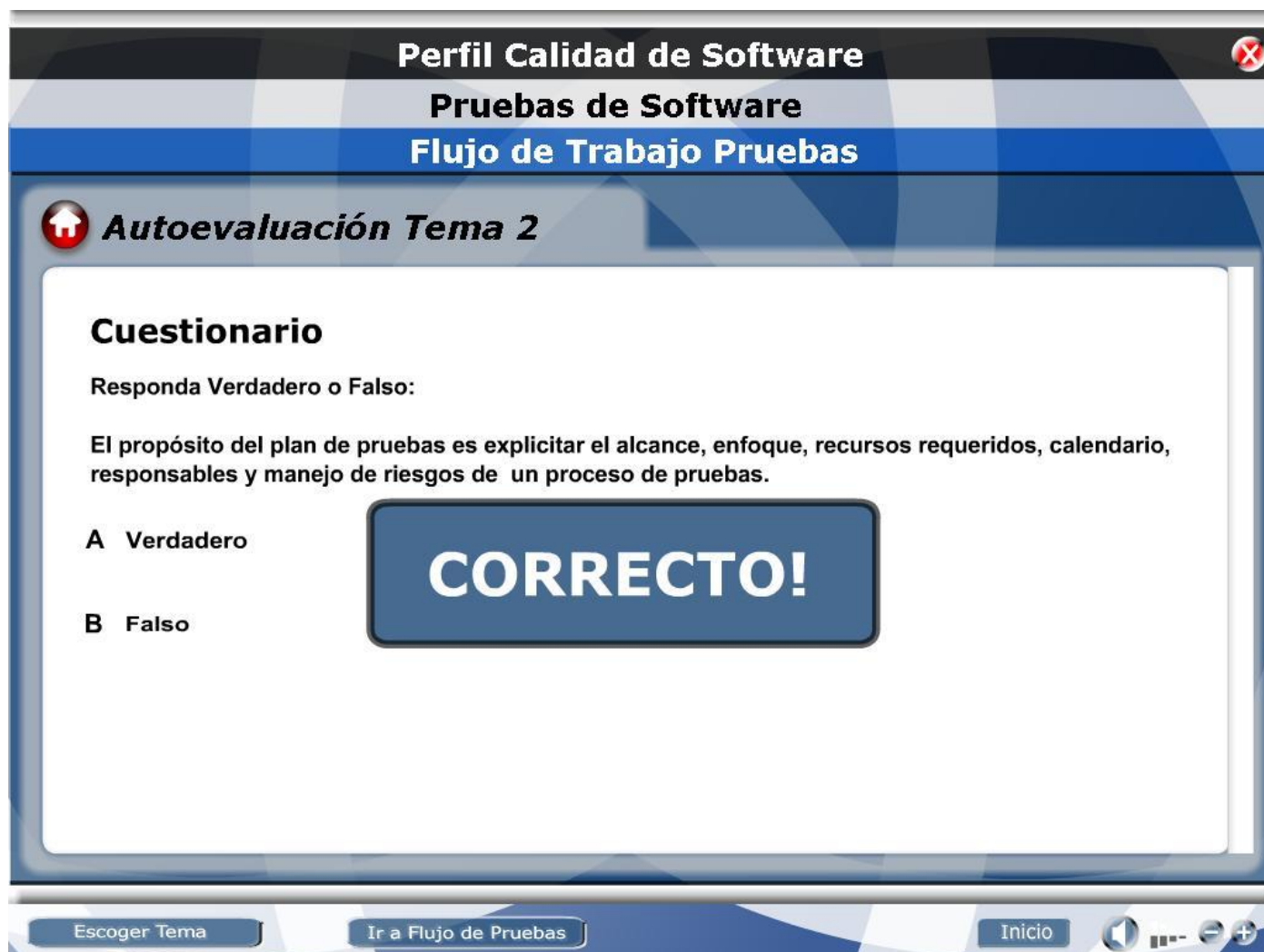


Figura 6. Pantalla Autoevaluación.