



Facultad 8

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Análisis y Diseño del producto



Autor: Yusimy Rodríguez Ruiz

Tutor: Sergio Díaz Catalá

Ciudad Habana, Junio 2007

DECLARACIÓN DE AUTORÍA

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

Para que así conste firmo la presente a los días del mes de Junio del año 2007

Yusimy Rodriguez Ruiz

Firma de la Autora

Sergio Díaz Catalá

Firma del Tutor

*“Haz lo necesario para lograr tu más ardiente deseo,
y acabarás lográndolo”*

Ludwig Van Beethoven

AGRADECIMIENTOS

Le agradezco a mis padres por tanto amor y tanto sacrificio, por estar siempre ahí dándome todo su apoyo y todo su amor. Por impulsarme hacia el camino correcto y hacer que me sienta hoy satisfecha por el esfuerzo que he hecho. Por haber formado una familia que me ha dado lo mejor del mundo y ha puesto todo su empeño en la realización de este sueño.

A mi hermanita linda, por las llamadas de las mañanas que me alentaban a seguir adelante y a esforzarme más. Por darme todo su apoyo y por estar desde que yo era pequeñita a mi lado y preocuparse mucho por mí y por mis cosas....Por darme a Marquito, esa personita para la que quiero ser un gran ejemplo para que se sienta orgulloso de su tía que tanto lo ama.

A Eric, por su apoyo, su comprensión.... y por malcriarme.

A Yoisy, por las tantas veces que me socorrió incondicionalmente y por los apuros de los que me sacó.

A todita mi familia, que de una u otra forma, en algún u otro momento me apoyaron.....

A los que no me conocían y aun así me ayudaron, a los que sí y también lo hicieron....

A mi tutor.....

DEDICATORIA

A mi mamá y a mi papá les dedico mi tesis, su sueño.

RESUMEN

El presente trabajo aborda los principales aspectos que se desarrollaron con el fin de realizar el análisis y diseño del software educativo “Pizarrón Virtual”. En él se muestran todos los pasos que se siguieron para diseñar una herramienta que cumpliera con los requerimientos del cliente. A partir del mismo se obtuvo una documentación para guiar una correcta implementación de una herramienta que apoye el trabajo en el aula, de una forma didáctica, amena y divertida. Por otro lado sirve como material de estudio para futuras investigaciones y constituye una importante guía para la realización de proyectos similares. Con este documento quedó consumado el objetivo general de la investigación: “realizar el análisis y diseño de una aplicación para apoyar las técnicas de resolución de problemas en la asignatura “Algorítmica y Programación” de la mención de Informática en la escuela Técnica Comercial “Luís Razetti”” y además; demuestra que con las preferencias que tienen hoy los jóvenes por las Tecnologías de la Información y las Comunicaciones, la herramienta que aquí para una posterior implementación, constituye la respuesta ideal y completamente factible ante la situación existente en dicho centro.

ÍNDICE

Introducción	1
Capítulo 1	5
Fundamentación Teórica	5
1.1 Introducción.....	5
1.2 Los medios de enseñanza en la educación de hoy	6
1.3 Software educativo. Proceso de desarrollo.....	8
1.4 Tendencias a desarrollar software educativo.....	10
1.4.1 Aplicaciones WEB	11
1.4.2 Hipertexto, Multimedia, e Hipermedia	11
1.5 Análisis de otras soluciones existentes.....	13
1.6 Metodologías más utilizadas	14
1.7 RUP Y UML.....	17
1.7.1 RUP (Proceso Unificado de desarrollo de software).....	17
1.7.2 UML (Lenguaje Unificado de Modelado).....	20
1.8 OMMMA_L como extensión de UML para aplicaciones con multimedia	21
1.9 Rational Rose.....	23
1.10 Macromedia Flash 8.....	24
1.10.1 ActionScript	26
1.11 Soporte educativo	27
1.11.1 Características de la audiencia	27
1.11.2 Modelo pedagógico	28
1.11.3 Naturaleza de Actividades Educativas	30
1.12 Conclusiones.....	31
Capítulo 2	32
Descripción de la solución propuesta	32
2.1 Introducción.....	32
2.2 Modelo del Dominio.....	32

2.2.1	Diagrama del Modelo del Dominio	33
2.2.2	Conceptos	33
2.3	Diagrama de Navegación.....	34
2.4	Descripción del sistema propuesto	35
2.4.1	Descripción de la funcionalidad.....	35
1.	Requisitos funcionales del sistema	36
1.	Requisitos no funcionales del sistema	37
2.5	Modelo de Casos de Uso	39
2.5.1	Actores	39
2.5.2	Casos de Uso del Sistema	40
2.5.2.1	Diagrama y descripción textual de los Casos de Uso	40
2.5.2.2	Vista Global de los actores y casos de uso del sistema.....	50
2.6	Conclusiones.....	52
Capítulo 3	53
Construcción de la solución propuesta.....		53
3.1	Introducción.....	53
3.2	Modelo de Diseño	53
3.2.1	Diagrama de clases del modelo de objetos	53
3.2.2	Diagrama de Presentación.....	55
3.3	Modelo de Implementación	61
3.3.1	Diagrama de paquetes	62
3.3.2	Diagrama de componentes	62
3.3.2.1	Diagrama de componentes general	63
3.3.2.1.1	Diagrama del paquete “Conceptos”	64
3.3.2.1.2	Diagrama del paquete “Técnicas”	65
3.3.2.1.3	Diagrama del paquete “Análisis”	66
3.3.2.1.4	Diagrama del paquete “Ejercicios”	66
3.4	Modelo de Despliegue.....	68
3.5	Conclusiones.....	69
Capítulo 4	70

Estudio de factibilidad	70
4.1 Introducción.....	70
4.2 Planificación.....	70
4.2.1 Estimación del esfuerzo.....	71
4.2.2 Tiempo total estimado para el proyecto y costo.....	79
4.3 Beneficios tangibles e intangibles.....	81
4.3.1 Tangibles.....	81
4.3.2 Intangibles.....	81
4.4 Análisis de costos y beneficios.....	81
4.5 Conclusiones.....	82
Conclusiones	83
Recomendaciones	84
Referencias bibliográficas	85
Bibliografía	89
Glosario de Términos	92

ÍNDICE DE ILUSTRACIONES

<i>Ilustración 1. Proceso Unificado de Desarrollo de Software</i>	18
<i>Ilustración 2. Típico perfil de un proyecto</i>	19
<i>Ilustración 3. El vocabulario de UML</i>	21
<i>Ilustración 4. Logotipo de la herramienta Macromedia Flash 8</i>	25
<i>Ilustración 5. Modelo de Dominio</i>	33
<i>Ilustración 6. Mapa de navegación</i>	35
<i>Ilustración 7. Diagrama de Casos de Uso del Sistema Cargar _presentación</i>	40
<i>Ilustración 8. Diagrama de Casos de Uso del Sistema Controlar _locución</i>	42
<i>Ilustración 9. Diagrama de Casos de Uso del Sistema Consultar _glosario</i>	43
<i>Ilustración 10. Diagrama de Casos de Uso del Sistema Controlar _navegación</i>	44
<i>Ilustración 11. Diagrama de Casos de Uso del Sistema Salir _del _sistema</i>	45
<i>Ilustración 12. Diagrama de Casos de Uso del Sistema Mostrar _contenido _de _módulos</i>	46
<i>Ilustración 13. Diagrama de Casos de Uso del Sistema Interactuar _con _ejercicio</i>	48
<i>Ilustración 14. Diagrama de Casos de Uso del Sistema</i>	51
<i>Ilustración 15. Diagrama de Clases del Modelo de Objetos</i>	54
<i>Ilustración 16. Diagrama de presentación Inicio</i>	56
<i>Ilustración 17. Diagrama de presentación Módulos</i>	57
<i>Ilustración 18. Diagrama de presentación Glosario</i>	58
<i>Ilustración 19. Diagrama de presentación Ejercicio</i>	59
<i>Ilustración 20. Diagrama de presentación Repetir Ejercicio</i>	60
<i>Ilustración 21. Diagrama de presentación Salir</i>	60
<i>Ilustración 22. Diagrama de presentación Créditos</i>	61
<i>Ilustración 23. Diagrama de paquetes</i>	62
<i>Ilustración 24. Diagrama de componentes general</i>	63
<i>Ilustración 25. Diagrama de componentes del paquete "Conceptos"</i>	64
<i>Ilustración 26. Diagrama de componentes del paquete "Técnicas"</i>	65
<i>Ilustración 27. Diagrama de componentes del paquete "Análisis"</i>	66
<i>Ilustración 28. Diagrama de componentes del paquete "Ejercicios"</i>	67
<i>Ilustración 29. Diagrama de despliegue</i>	68

ÍNDICE DE TABLAS

<i>Tabla 1</i> Posibilidad interactiva	30
<i>Tabla 2</i> Naturaleza de actividades educativas.....	30
<i>Tabla 3</i> Casos de Uso del sistema.....	40
<i>Tabla 4</i> Caso de uso Cargar _ presentación.....	41
<i>Tabla 5</i> Caso de uso Controlar _ locución.....	42
<i>Tabla 6</i> Caso de uso Consultar _ glosario.....	43
<i>Tabla 7</i> Caso de Uso Controlar _ navegación.....	44
<i>Tabla 8</i> Caso de Uso Salir _del _sistema.....	45
<i>Tabla 9</i> Caso de Uso Mostrar _contenidos _de _módulos.....	46
<i>Tabla 10</i> Caso de Uso Interactuar _con _ejercicio.....	49
<i>Tabla 11</i> Cálculo de Puntos de Casos de Uso sin Ajustar.....	71
<i>Tabla 12</i> Cálculo del Factor de Peso de los Actores sin ajustar.....	72
<i>Tabla 13</i> Cálculo del Factor de Peso de los Casos de Uso sin ajustar.....	72
<i>Tabla 14</i> Obtención de los Puntos de Casos de Uso Ajustados.....	73
<i>Tabla 15</i> Factores de complejidad técnica (TCF).....	73
<i>Tabla 16</i> Factores Ambiente.....	75
<i>Tabla 17</i> Esfuerzo por flujo de trabajo.....	78
<i>Tabla 18</i> Resultados estimados	82

INTRODUCCIÓN

El surgimiento y desarrollo de las Tecnologías de la Información y las Comunicaciones (TICs) le ha impuesto múltiples retos a la educación actual. Uno de ellos es dar respuesta a los profundos cambios sociales, económicos y culturales que se prevén para los próximos años. Esto conlleva a crear nuevos estilos de enseñanza que adapten a los futuros profesionales a estos tiempos de cambio; lo que implica innovar dentro de la actividad docente, mejorar la comunicación didáctica y conseguir una enseñanza más activa con un mayor protagonismo por parte de los estudiantes.

La clave para alcanzar tal propósito está dada por el desarrollo de la computación como medio de enseñanza, la cual mejora la motivación, el rendimiento y las capacidades cognitivas del alumnado (resolución de problemas, planificación de tareas, etc.).

En los últimos años, el auge alcanzado por las Tecnologías de la Información y las Comunicaciones, ha revolucionado el sistema de enseñanza y aprendizaje a nivel mundial, transformando el modelo clásico de la clase y propiciando así una mejor comprensión de los estudiantes y un mayor enriquecimiento intelectual para ellos y para los profesores.

En torno a este acontecer ha surgido el Centro Nacional de Tecnologías de Información (CNTI), el cual tiene como objetivo impulsar y respaldar las actividades de docencia y desarrollo científico-tecnológico de las Instituciones Académicas y Centros de Investigación en Venezuela, así como diseñar estrategias en materia de Tecnologías de Información que fomenten su implementación. La gerencia de dicho centro, en el marco del convenio Cuba-Venezuela, puso en marcha en Septiembre del 2004 diversas actividades orientadas a la creación del centro de Contenidos Educativos de Tecnología de Información y Comunicación (CETIC) dirigido a la producción de soluciones educativas computarizadas, donde también participa el Ministerio de Educación y Deportes a través de la Fundación Bolivariana de Informática y Telemática (FUNDABIT) la cual tiene como misión incorporar las Tecnologías de la Información y las Comunicaciones (TIC), en el proceso educativo para contribuir con la formación integral de la persona.

Al calor de este convenio surge la idea de hacer el software educativo “Pizarrón Virtual” para la escuela técnica comercial “Luis Razetti” donde los estudiantes de 2do año de la mención Informática de la especialidad de Servicios Administrativos presentan dificultades a la hora de resolver problemas en la asignatura “Algorítmica y Programación”, puesto que el aprendizaje de la programación no solo requiere la adquisición de conocimientos, sino también del desarrollo de las capacidades del pensamiento. También gran parte de las complicaciones que tienen estos estudiantes para resolver problemas tiene que ver con el enunciado, ya que no pueden identificar la información relevante que se proporciona en los problemas y tampoco logran la imagen o representación mental o interna del enunciado. Además, en dicho centro estudiantil, existen escasos proyectos educativos que contemplen el uso de la tecnología en el proceso de enseñanza-aprendizaje; los docentes presentan insuficiente conocimiento sobre las posibilidades que ofrecen las redes para potenciar su formación, capacitación y actualización; tanto profesores como estudiantes tienen restricciones de acceso a una bibliografía adecuada y actualizada.

A partir de la situación anterior surge el **problema** de que la falta de una herramienta didáctica que utilice las Tecnologías de la Información y las Comunicaciones para apoyar las técnicas de resolución de problemas en la asignatura “Algorítmica y Programación” de la mención de Informática en la escuela Técnica Comercial “Luís Razetti”, está afectando el proceso de enseñanza-aprendizaje de los estudiantes en dicha asignatura. Para resolver esto se traza como **Objeto de estudio** de la investigación: el Proceso de desarrollo de un software que aplique las técnicas hipermedia. Donde se plantea como **Campo de acción**: Análisis y diseño del software educativo “Pizarrón Virtual”.

Se define como **Objetivo general** del trabajo: realizar el análisis y diseño del software educativo “Pizarrón Virtual” para apoyar las técnicas de resolución de problemas en la asignatura Algorítmica y Programación de la mención de Informática de la escuela Técnica Comercial “Luis Razetti”.

Con este trabajo se defiende la idea de que si se hace un correcto análisis y diseño del software educativo “Pizarrón Virtual”, se podrá implementar una herramienta que apoye las técnicas de resolución de problemas en la asignatura Algorítmica y Programación de la escuela “Luis Razetti”.

Para complementar el objetivo de la investigación se han formulado algunos **Objetivos específicos** que posibilitan el desarrollo del mismo.

- Identificar y analizar la información referente al proyecto.
- Estudiar el estado del arte de las herramientas educativas en las aulas.
- Asimilar los procesos de análisis y diseño de desarrollo de software educativo.
- Facilitar la implementación de un software eficiente y de máxima calidad.

Se han definido también algunas **Tareas de investigación** que responden a estos objetivos planteados:

- Realizar un diagnóstico para determinar el estado actual del problema.
- Sintetizar el estudio realizado sobre el problema.
- Seleccionar la metodología y herramientas más afines al diseño de aplicaciones hipermedias.
- Realizar el análisis y diseño del software.

Como **aportes** de este trabajo se espera obtener el diseño de un software educativo que guíe la correcta implementación de una herramienta que apoye el trabajo en el aula, de una forma didáctica, amena y divertida, y que constituya un medio dinámico para aprender y desarrollar habilidades en la solución de problemas algorítmicos. Además, el trabajo sirve como material de estudio para futuras investigaciones y constituye una importante guía para la realización de proyectos similares.

El mismo se encuentra estructurado en 4 capítulos:

El capítulo 1 aborda la fundamentación teórica de trabajo. En él se habla de las tendencias y herramientas fundamentales que se tuvieron en cuenta para elaborar la respuesta, así como de la situación actual en que se encuentra el proceso de desarrollo del software educativo en el mundo, y en la que se encuentran las aulas con la inclusión de las Tecnologías de la Información y las Comunicaciones.

En el capítulo 2 se describe, a través de diagramas, cómo está conformado el entorno que se desea automatizar y cómo funcionará luego de que esté automatizado. Además, se detallan aspectos funcionales y no funcionales con los que cumplirá la aplicación.

Igualmente se exponen diagramas en el capítulo 3, los cuales guían la construcción del sistema detallando el diseño previo a la implementación con diagramas propuestos por la metodología seleccionada, describiendo los componentes en los que se implementará y acomodando el despliegue de la herramienta.

Por último, en el capítulo 4 se desarrolla un estudio de factibilidad dado por la planificación, los costos, los beneficios tangibles e intangibles y el análisis de costos-beneficios cuyo fin es concretar si es factible o no la realización del proyecto.

CAPÍTULO 1

FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

“Los que se enamoran de la práctica sin la teoría son como los pilotos sin timón ni brújula, que nunca podrán saber a dónde van”

Leonardo Da Vinci

El desarrollo del software en la actualidad está guiado por tendencias y tecnologías que son precisas tener en cuenta para el desarrollo de este trabajo. En el presente capítulo se describe cómo se encuentra actualmente el proceso de desarrollo del software, más específicamente del software educativo y cómo se encuentra el mundo con la entrada de este último a las aulas. También se puntualiza por qué se decidió hacer un software educativo con la utilización de la técnica hipertexto y se exponen las tendencias actuales que giran alrededor de la producción de este tipo de software; además, se presenta un estudio de las metodologías más afines a estos procesos así como la justificación de la metodología y herramientas seleccionadas para la confección de la aplicación.

1.2 Los medios de enseñanza en la educación de hoy

La evolución de los medios de enseñanza a lo largo de la última década no es sólo una consecuencia del avance tecnológico, que facilita la inclusión de los medios informáticos en el aula, sino también del cambio producido en el mundo de los adolescentes, cuyo vehículo de transmisión de la información que reciben y generan es ahora diferente a la tradicional expresión oral y escrita. (1)

El marco en el que ahora se pueden desarrollar los aprendizajes es totalmente distinto al que existía dos décadas atrás: virtualidad, nuevos roles del profesorado y de los estudiantes, nuevos materiales formativos son aspectos que nos acontecen hoy en día.

El lenguaje audiovisual y la interacción de los niños y adolescentes con los ordenadores en los juegos, búsqueda en Internet, televisión, conversaciones en chat, etc., influyen en la interpretación de la información que les llega a través de cualquier medio. Esto sugiere que los medios informáticos son instrumentos que pueden favorecer el aprendizaje, pero se hace necesario aclarar que, como dice el doctor Jaime Sánchez, “No es la tecnología la que debe salir a la búsqueda de una aplicación educativa. Por el contrario, la idea es partir de la educación, la que ante una necesidad busca un soporte tecnológico para que sea socio en la construcción significativa del aprende” (2)

Por estas razones y tras estudios realizados se ha hecho evidente la necesidad que existe de introducir medios tecnológicos en los procesos de aprendizaje, dentro de los cuales juega un papel fundamental el software educativo.

“Se ha descubierto que como consecuencia de muchas actividades emprendidas cuando se utiliza software educativo, los estudiantes pueden responsabilizarse más de su propio aprendizaje que en otros casos”. A su vez, se ha observado que la utilización de estos recursos tiene implicancias en el clima de la clase y “ayuda a crear ambientes enriquecidos de aprendizaje y favorece el aprendizaje significativo”, promueven la motivación, aportan estímulos nuevos, activan la respuesta del alumno, proporcionan información, estimulan la práctica, establecen la sucesión de aprendizajes y proporcionan recursos.(3)

Un estudio realizado por la OECD (Organization for Economic Co-operation and Development) indica que los estudiantes que utilizan regularmente el computador, obtienen mejores resultados en los contenidos claves que aquellos con menos experiencia y menos confianza en sus competencias tecnológicas. Los datos de este estudio titulado: "¿Están los alumnos preparados para un mundo tecnologizado?", demuestran que los estudiantes que han usado el computador durante varios años tienen mejor desempeño en Matemáticas que el promedio de los alumnos. Por el contrario, los estudiantes que no tienen el acceso a computadores o quienes han estado usándolos sólo por un corto período de tiempo, tienden a quedarse atrás respecto a sus compañeros. (4)

Por ello, hoy en día el papel de los maestros no es tanto "enseñar" (explicar-examinar) unos conocimientos que tendrán una vigencia limitada y estarán siempre accesibles, como ayudar a los estudiantes "aprender a aprender" de manera autónoma en esta cultura del cambio y promover su desarrollo cognitivo y personal mediante actividades críticas y aplicativas que, aprovechando la inmensa información disponible y las potentes herramientas TIC, tengan en cuenta sus características (formación centrada en el estudiante) y les exijan un procesamiento activo e interdisciplinario de la información para que construyan su propio conocimiento y no se limiten a realizar una simple recepción pasiva, o memorización de la información.

Por otra parte, la diversidad de los estudiantes y de las situaciones educativas que pueden darse, aconsejan que los maestros aprovechen los múltiples recursos disponibles (que son muchos, especialmente si se utiliza el ciberespacio) para personalizar la acción docente, y trabajen en colaboración con otros colegas (superando el tradicional aislamiento, propiciado por la misma organización de las escuelas y la distribución del tiempo y del espacio) manteniendo una actitud investigadora en las aulas, compartiendo recursos (por ejemplo a través de las webs docentes), observando y reflexionando sobre la propia acción didáctica-pedagógica y buscando progresivamente mejoras en las actuaciones acordes con las circunstancias (investigación-acción).(5)

Pero el problema con todo esto es que los profesores, en su gran mayoría temen enfrentarse con herramientas educativas en el aula, puesto que muchos no están adecuadamente preparados. De

acuerdo a un estudio realizado en los Estados Unidos de Norteamérica por la Comisión del Presidente para el Aprendizaje Basado En La Web (6) , la inadecuada capacitación del docente es una de las más grandes barreras para la integración de las tecnologías (TIC) al currículo escolar. Este concluye con que "Desafortunadamente, muchos docentes carecen de los conocimientos, habilidades y estrategias necesarias para integrar la tecnología al currículo. Algunos, sencillamente no la utilizan. Los expertos sugieren que muchos de esos profesores evitan hacerlo porque no cuentan con alguien que pueda ayudarlos, asesorarlos sobre cómo lograrlo."

Todo esto conlleva a que los centros desarrollen una profunda reestructuración de todos sus elementos, puesto que las nuevas tecnologías producen un cambio en el entorno y como la escuela lo que pretende es preparar a la gente para este entorno, si éste cambia, la actividad de la escuela tiene que cambiar y con ella la posición de los profesores y de los estudiantes.

1.3 Software educativo. Proceso de desarrollo

El software es un ingrediente indispensable para el funcionamiento del computador. Está formado por una serie de instrucciones y datos, que permiten aprovechar todos los recursos que el computador tiene, de manera que pueda resolver gran cantidad de problemas. Un computador en sí, es sólo un conglomerado de componentes electrónicos; el software le da vida al computador, haciendo que sus componentes funcionen de forma ordenada.

Existen varios tipos de software y entre ellos se encuentra el software de aplicación, que permite a los usuarios llevar a cabo una o varias tareas más específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre otros: aplicaciones de automatización industrial, aplicaciones ofimáticas, software educativo, software médico, bases de datos, videojuegos.

El presente trabajo se centra en realizarle el proceso de análisis y diseño a un software educativo, ¿pero qué es un Software educativo? Varios autores han definido software educativo como "aquel material de

aprendizaje especialmente diseñado para ser utilizado con un computador en los procesos de enseñar y aprender”, según expresa la doctora María Alejandra Torres. (7)

Con lo anterior también concuerda la licenciada Ana M Andrada que dice que “es todo programa para computadora que se desarrolla con la finalidad específica de ser utilizado como recurso didáctico en procesos de enseñanza y de aprendizaje” (8) a lo que la Dra. Begoña Gros agrega que “están pensados para ser utilizados en un proceso normal de aprendizaje y por ese motivo se establece un diseño específico a través del cual se adquieran unos conocimientos, unas habilidades, unos procedimientos, en definitiva, para que un estudiante aprenda”. (9)

En resumen, software educativo constituye todo aquel programa de cómputo que se desarrolle con fines educativos y que esté avalado por un especialista en pedagogía.

Los primeros intentos de desarrollo de este tipo de software se sitúan al final de la década del 60’ con la aparición de los sistemas de instrucción programada, pero el verdadero auge se dio en la década del 80’. En primera instancia con la producción de lenguajes para el aprendizaje, luego con el desarrollo de herramientas de autor para la producción de software educativo y ya más específicamente con la elaboración de programas tutoriales, de ejercitación y práctica, de cálculo, y de simulación.

Para hacer un software se realiza un proceso de desarrollo de software que no es más que, coincidiendo con Pressman, el conjunto de actividades que guían los esfuerzos de las personas implicadas en un proyecto y que tienen la misión de transformar los requerimientos del usuario en un producto, de manera tal que tanto el usuario final como las personas implicadas en el proyecto tengan una visión similar del mismo (10).

Pero este conjunto de actividades no es el mismo puesto que cada software tiene características propias. Es por esto que se han hecho diversos estudios para adecuar cada proceso de desarrollo de software a cada tipo de software, y de estos se han obtenido diferentes metodologías de desarrollo que abarcan desde el análisis hasta el despliegue de la aplicación.

Entre algunas de estas metodologías existentes y utilizadas hoy en día en los procesos de elaboración de aplicaciones que utilizan técnicas multimedia, se encuentran (11):

- MultiMet, que es una metodología cubana que se utiliza para modelar el proceso de creación de una multimedia y está orientada a las etapas de concepción más que a la descripción de la modelación del producto como tal.
- HDM (Hypertext Design Model), que fue creada por los profesores Franca Garzotto, Paolo Paolini y Daniel Schwabe en 1991 y es más que un intento de modelar la estructura del hipertexto-hipermedia, una modelización de las estructuras de navegación.
- OOHDM (Object Oriented Hypermedia Design Model) que es la sucesora de HDM y se basa en la teoría de orientación a objetos.
- RUP (Racional Unified Process) que permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual y posibilita a los grupos de desarrollo producir aplicaciones informáticas más robustas y flexibles que se adecuan a las necesidades de los usuarios.

Todas estas metodologías surgen o fueron adaptadas para crear software educativo ante la apremiante necesidad de tener medios de enseñanza que apoyen la revolución tecnológica que existe en las aulas de hoy en día.

1.4 Tendencias a desarrollar software educativo

En la actualidad cuando se solicita crear software educativo existen preferencias de desarrollar utilidades para la presentación de los contenidos (textos, animaciones, gráficos, vídeos) o de herramientas de comunicación entre profesores y estudiantes (correo electrónico, Chat, foros). Esto puede resumirse en crear aplicaciones que utilicen las técnicas hipertexto, multimedia o hipermedia o en desarrollar aplicaciones Web. Esto es consecuencia de que las tendencias actuales de software

educativo se inclinan hacia estas vertientes debido a la aparición de Internet y a la inclusión de las TICs en el aula que han propiciado un marco ideal para que estas se desarrollen. La educación a distancia y la utilización del computador como recurso didáctico en el aula está muy de moda en la actualidad.

1.4.1 Aplicaciones WEB

Las aplicaciones Web educativas por lo general se centran en brindar información a educadores, ofertándole cursos, eventos, actividades, concursos, admisiones, y herramientas didácticas para apoyar su trabajo en el aula entre otras cosas. En Latinoamérica y en el mundo en los últimos años se ha incrementado el número de aplicaciones de este tipo que brindan servicios a los profesores y actualmente, diversos son los portales Web que están a disposición de todos los que deseen vincular su aprendizaje con las nuevas tecnologías de información y comunicación.

Cuba es un vivo ejemplo que tiene sitios dónde se le brinda información al docente y se ofertan aplicaciones educativas. Ejemplo de esto es el sitio “Software educativo en Guantánamo” el cuál muestra una variedad de productos elaborados para apoyar la educación en Cuba. También está el sitio de CITMATEL con más de 10 años de experiencia que cuenta con una variada oferta de software educativo.

Así mismo, en Latinoamérica tenemos portales de este tipo tal cómo el portal chileno “chile@prende” [<http://www.chileaprende.cl>], “Educar” el portal educativo argentino [<http://www.educ.ar>], “Tiza y Mouse” [<http://www.tizaymouse.com>], “Eduteka” [<http://www.eduteka.org>], “Colombia Aprende” [<http://www.colombiaprende.edu.co>], “Portal Educativo de las Américas” OEA [<http://www.educoas.org>] y otros tantos cuya finalidad es ofertar recursos educativos e informar a los profesores y usuarios interesados.

1.4.2 Hipertexto, Multimedia, e Hipermedia

Con el propósito de incentivar el trabajo de los profesores en el aula y que los estudiantes tengan un proceso de aprendizaje más ameno y productivo hoy en día se elaboran diversos software educativos y

se exponen en la mayoría de las veces en sitios como los antes expuesto. Ejemplo de este software son las aplicaciones que utilizan técnicas multimedia para mostrar la información, que no son más que un conjunto de varias medias tales como texto, sonido, imágenes; o, como se plantea en Microsoft Encarta Biblioteca 2002: "Multimedia, en informática, es la forma de presentar la información que emplea una combinación de texto, sonido, imágenes, animación y vídeo. Una combinación de tres o más de estos elementos con alguna medida de interactividad de usuario se podría considerar normalmente una aplicación multimedia. Entre las aplicaciones informáticas multimedia más corrientes figuran juegos, programas de aprendizaje y material de referencia como la Enciclopedia Encarta".

También existen los Hipertextos que son documentos digitales o no, que se puede leer de manera no secuencial o lineal de acuerdo sea la necesidad. 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.

Por otro lado se encuentra el término Hipermedia, combinación de los conceptos Hipertexto y Multimedia, que 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, vídeo, etc.) que conforman el documento. Un sistema hipermedia es una estructura similar a la del hipertexto donde la información contenida en los nodos es multimedia (textos, imágenes, secuencias de animaciones, sonidos, vídeo). Podemos concluir entonces que la diferencia entre un sistema hipertexto y un sistema hipermedia radica en el tipo de información contenida en sus nodos.

Como respuesta al problema de este trabajo se propone la realización de un sistema hipermedia ya que este no sólo convierte al usuario en un receptor de mensajes elaborados por otros como la mayoría de los medios de comunicación, sino que hace posible que exista un intercambio de información entre el usuario y el mensaje emitido; además puede decidir qué secuencia de información debe seguirse, qué ritmo, qué cantidad y qué profundización de la información. Adicionándole a esto que el hecho de que

sea una combinación del hipertexto con la multimedia posibilita conjugar los beneficios de ambas tecnologías tomando de la multimedia la riqueza de expresión y del hipertexto la geometría que permite que los datos puedan ser explorados y presentados siguiendo diferentes secuencias. Esto trae como ventajas que permite presentar la información poco o nada estructurada, su interfaz de usuario es muy intuitiva. Además constituye un marco idóneo para la autoría en colaboración, al permitir la compartición, distribución y personalización de la información.

1.5 Análisis de otras soluciones existentes

El mejor sustento de toda investigación se centra en los estudios anteriores sobre el caso. Es por esto que continuación se presentan ejemplos de soluciones semejantes a la que este trabajo propone.

➤ Peraza, N y Parra T (2001), “La comprensión lectora en la resolución de problemas matemáticos” elaborado en la Universidad Simón Bolívar Caracas donde las autoras se proponen propiciar en los alumnos de tercer grado de educación básica un estímulo a través de un material educativo computarizado (MEC) enfocado bajo la perspectiva de sistema de ejercitación y práctica para la comprensión de enunciados verbales en problemas matemáticos. Se inscribe en la modalidad de proyecto factible basada en tres fases; diagnóstica, apoyo de investigación documental y elaboración de la propuesta y evaluación de la factibilidad. Este prototipo trata de aprovechar al máximo la capacidad de interacción que brindan los ambientes computarizados, favoreciendo la participación activa del estudiante.

➤ Lage F, y Cataldi , “Una experiencia de Resolución de Problemas a través de Modelos Cooperativos-Colaborativos aplicada a Algoritmia usando nuevas Tecnologías de Comunicación” fue el título del trabajo elaborado en la Universidad de Buenos Aires en la Facultad de Informática para la carrera de Ingeniería de sistemas debido a los problemas detectados en los alumnos evidenciado por los fracasos en los exámenes en los últimos cuatrimestres, mediante foros de discusión con seguimiento de los alumnos en la realización de tareas cooperativas y colaborativas.

➤ Cárdenas F, Castillo N y Daza E. “Editor e Intérprete de Algoritmos Representados en Diagramas de Flujo” desarrollado en la Universidad de Magdalena de Santa Marta Colombia,

diseñado para estudiar algoritmos representados con diagramas de flujo, su interfaz gráfica facilita en gran medida la creación de un software tutor de algoritmos, el cual contendría una amplia explicación de ejemplos y conceptos necesarios para aprender algoritmos, disponibles a través de gráficas, animaciones, sonido y texto.

Las soluciones antes expuestas, a pesar de que fueron diseñadas con el mismo objetivo que tiene el software que aquí se propone, fueron elaboradas para una audiencia muy específica y cerrada y su adaptación a la audiencia para la cual se concibe el producto “Pizarrón Virtual” es bastante compleja pues el nivel de madurez de pensamiento es muy distinto. Además, las condiciones que ofrecen estos productos no satisfacen los requisitos exigidos por estudiantes con las características para las cuales va dirigida la propuesta.

1.6 Metodologías más utilizadas

Para la realización de este trabajo, como se ha justificado anterior, se ha decidido hacer un sistema hipermedia. Para crear este tipo de aplicación, como para la creación otra cualquiera, es preciso seguir una metodología que guíe el trabajo de forma organizada.

Habitualmente, el desarrollo sistemas o de documentos multimedia e hipermedia se lleva a cabo utilizando directamente herramientas de autor, descuidándose el importante proceso previo de análisis y diseño conceptual de los aspectos estructurales, de navegación y de interfaces con el usuario, lo que hace complicado su mantenimiento futuro. (12)

Sin embargo, en los últimos años (13), existe una tendencia a considerar el desarrollo multimedial e hipermedial como un proceso de ingeniería del software, por lo que ya se han propuesto diferentes metodologías, antes mencionadas, como HDM (Hypertext Design Model), OOHDM (Object-Oriented Hypermedia Design Model) o RMM (Relationship Management Methodology), que establecen la necesidad de considerar un diseño previo a la construcción de cualquier sistema o documento multimedia e hipermedia, y ofrecen una serie de técnicas, más o menos formales, para recoger en diferentes modelos abstractos las especificaciones del sistema a desarrollar.

Con el fin de llevar a cabo el desarrollo del producto de manera organizada y bien documentada, se realizó el estudio de las metodologías más afines a al desarrollo de este tipo de aplicación. A continuación se muestra una descripción de cada una de ellas.

- **HDM** (Hypertext Design Model) fue creada con el objetivo de crear un modelo que sea de utilidad para realizar el diseño de una aplicación de hipertexto (o hipermedia). (14)

Más que describir la estructura interna de una aplicación que utilice hipermedia, crea un modelo que sea de utilidad para realizar el diseño de una aplicación a partir de la estructura de datos siendo muy útil para el análisis de aplicaciones ya existente (con el fin de detectar errores en la estructura navegacional) y comparaciones entre ellas. Es un modelo más bien para evaluar aplicaciones (evaluación orientada al diseño).

La unidad básica del modelo es la entidad. Una entidad es la más pequeña parte autónoma de la información, es decir, que no necesita ninguna otra información para tener un sentido total. Las entidades se agrupan en tipos de entidades. Una entidad es una jerarquía de componentes, los cuales están formados a su vez por unidades. El concepto de unidad es similar al de nodo, mientras que un componente es un conjunto de nodos que crean una unidad lógica. (14)

Asimismo hay varios tipos de enlace; los más importantes en la estructura son los que unen componentes dentro de una entidad, y se denominan enlaces de componente o de perspectiva; los enlaces estructurales conectan componentes de la misma entidad; por último los enlaces de aplicación conectan componentes y entidades de distinto tipo, y son independientes de la estructura. (11)

Este modelo no permite que unas entidades sean componentes de otras entidades (y estas tienen complejas estructuras internas) y, aunque los autores no lo recomiendan, diferentes unidades pueden compartir el mismo cuerpo. (Según los autores, esto conlleva a una desorientación para el usuario, aunque por otra parte, fomentaría la reusabilidad). Seguir el modelo HDM es extremadamente complicado cuando el número de entidades involucradas crece. (11)

- **RMM** (Relationship Managment Methodology) fue la primera metodología para el diseño de multimedia. Se trata, probablemente, del único método para hipermedia que parece cubrir todo el ciclo de desarrollo, desde el estudio de factibilidad hasta la evaluación del sistema, aunque sólo propone actividades y productos concretos para las fases de análisis y de diseño.

La clase de aplicaciones para la cual RMM es más adecuada, corresponde a las que presentan una estructura regular para un dominio de interés, en donde hay clases de objetos, relaciones definibles entre éstas clases, y múltiples instancias de objetos dentro de cada clase. Ejemplo de software que satisfacen estos requerimientos son los catálogos de productos, las aplicaciones hipermediales frontales para bases de datos tradicionales o aplicaciones legadas. Considerando que muchas de ellas poseen datos volátiles que requieren actualizaciones frecuentes, se hace necesario disponer de medios que permitan automatizar y agilizar los desarrollos iniciales y los subsecuentes procesos de actualización

El análisis se realiza por medio de un diagrama entidad-relación en el que sólo se permiten relaciones con cardinalidades $1 \rightarrow 1$ o $1 \rightarrow N$ y tampoco es posible establecer relaciones reflexivas.

Con respecto al diseño, una de las características más relevantes de este método es que éste se hace tanto de forma ascendente como descendente, ofreciendo una interesante manera de realizar una verificación.

Para el resto de las fases, estudio de factibilidad, implementación, pruebas y evaluación, no se propone ningún tipo de guías.

RMM constituye una metodología tentadora para el desarrollo del proceso por el desglose de las fases de la producción y la incorporación de diagramas para el diseño de la presentación, el comportamiento dinámico y la estructura de la navegación. No obstante, su uso óptimo se basa en las aplicaciones de catálogo de productos y aplicaciones hipermediales frontales para bases de datos tradicionales o aplicaciones legadas por poseer una alta volatilidad de la información.

✓ Por último se realizó el estudio de **RUP** (Rational Unified Process), que fue la seleccionada. Esta metodología es un proceso de Ingeniería de Software cuyo objetivo es producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecidos, cubriendo el ciclo de vida de desarrollo del software. RUP junto con UML (Unified Modeling Language) constituye el estándar más utilizado para el análisis, implementación y documentación de un software en la industria de hoy en día.

1.7 RUP Y UML

1.7.1 RUP (Proceso Unificado de desarrollo de software)

Se seleccionó RUP porque es un modelo que involucra un análisis de riesgo, cubre todo el ciclo de vida del producto guiándolo a través de los casos de uso, soporta un enfoque de desarrollo iterativo e incremental, las iteraciones tempranas se enfocan en validar y producir una arquitectura de software basada en componentes, el ciclo de desarrollo inicial toma la forma de un prototipo ejecutable que gradualmente evoluciona convirtiéndose en el sistema final y además tiene implícito en su proceso de desarrollo la evaluación continua de la calidad con respecto a los requerimientos de calidad deseados.

El proceso está formado por dos estructuras o dimensiones como muestra la siguiente ilustración:

(Ilustración 1)

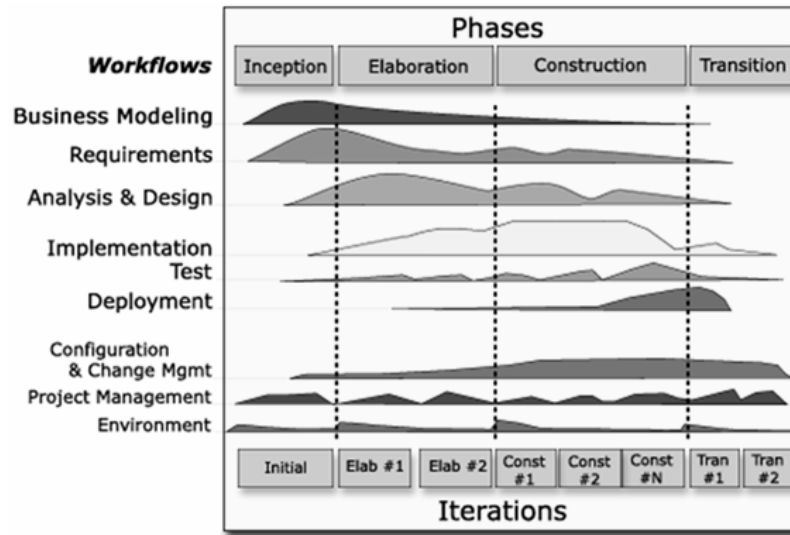


Ilustración 1. Proceso Unificado de Desarrollo de Software

- Un eje horizontal que representa el tiempo y muestra los aspectos del ciclo de vida del proceso.
- El eje vertical representa los flujos de trabajo del proceso, los cuales agrupan actividades de acuerdo a su naturaleza.

La primera dimensión representa la parte dinámica del proceso, y está expresada en términos de ciclos, fases e iteraciones.

La segunda dimensión representa la parte estática del proceso, cómo se describe en términos de componentes, actividades, flujos de trabajo, artefactos y actores.

Además se organiza en 4 fases las cuales terminan cada una con un hito. Estas fases son:

1. **Concepción o Inicio:** La idea, la visión del producto, cómo se enmarca en el negocio, el alcance del proyecto. Esta fase se culmina con los objetivos del ciclo de vida.
2. **Elaboración:** Planificar las actividades necesarias y los recursos requeridos, especificando las características y el diseño de la arquitectura. Esta fase culmina con la arquitectura del ciclo de vida.

3. **Construcción:** Desarrollar el producto y evolucionar la visión; la arquitectura y los planes hasta que el producto en una primera versión esté listo para ser enviado a la comunidad de usuarios. Esta fase culmina con la capacidad inicial de operación.
4. **Transición:** Realizar la transición del producto a los usuarios, lo cual incluye: manufactura, envío, entrenamiento, soporte y mantenimiento del producto hasta que el cliente esté satisfecho. Esta fase culmina con la Versión de Producto, la cual a su vez concluye el ciclo.

La ilustración 2 muestra un típico perfil de proyecto expresando el tamaño relativo de cada una de las fases así como el hito con que finaliza cada una.

(Ilustración 2)

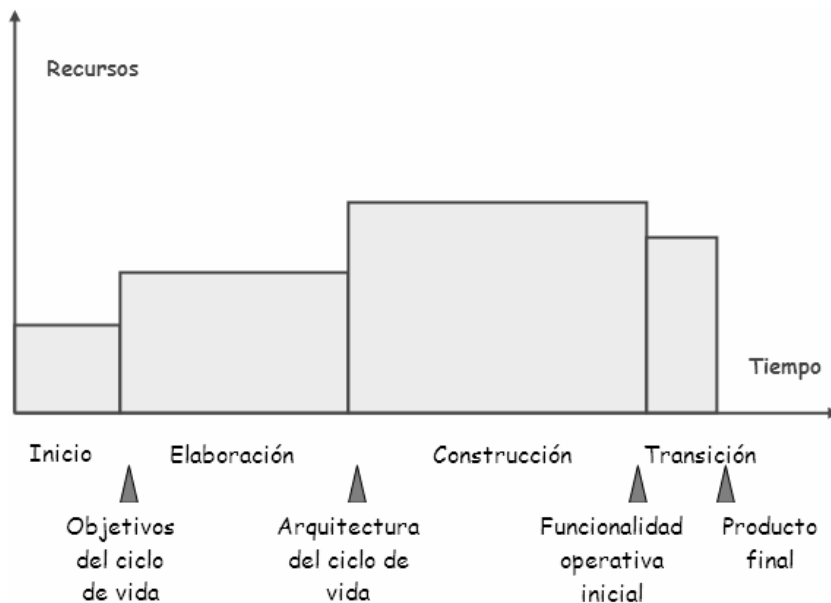


Ilustración 2. Típico perfil de un proyecto

RUP es una metodología que se modela con UML (Unified Modeling Language). La misma constituye una guía de cómo usar UML de forma más efectiva. Ambos están estrechamente relacionados entre sí pues, mientras RUP establece las actividades y los criterios para conducir un sistema desde su máximo

nivel de abstracción (la idea en la cabeza del cliente), hasta su nivel más concreto (un programa ejecutándose en las instalaciones del cliente), el segundo ofrece la notación gráfica necesaria para representar los sucesivos modelos que se obtienen en el proceso de refinamiento.

1.7.2 UML (Lenguaje Unificado de Modelado)

UML constituye un lenguaje estándar de modelado para la especificación, visualización, documentación y construcción de artefactos de sistemas en los que el software juega un papel importante. (15)

Este surge ante la necesidad que tenían los desarrolladores del software de comunicar sus modelos a otros desarrolladores, y no solo entre ellos mismo sino a todos los que estuvieran vinculados con el proyecto e incluso, a desarrolladores de futuras generaciones. Además, también se hacía necesario proporcionar un marco en el que desarrolladores individuales pudieran pensar o analizar y registrar estos modelos en documentos pues no podían retener todo en sus cabezas. (15)

La creación de UML, permitió a los creadores de software visualizar los resultados de su trabajo en esquemas o diagramas estandarizados que capturan sus ideas de forma convencional y fácil de comprender para comunicarlas a otros. (16)

Es de suma importancia aclarar que no es una guía para realizar en análisis y diseño orientado a objetos, o sea, no es un proceso. (17)

Este lenguaje propone un vocabulario que incluye tres categorías: elementos, relaciones y diagramas, y estas a su vez están compuestas por diferentes categorías. En la figura que a continuación se muestra se especifican cada una de ellas.(15)

(Ilustración 3)

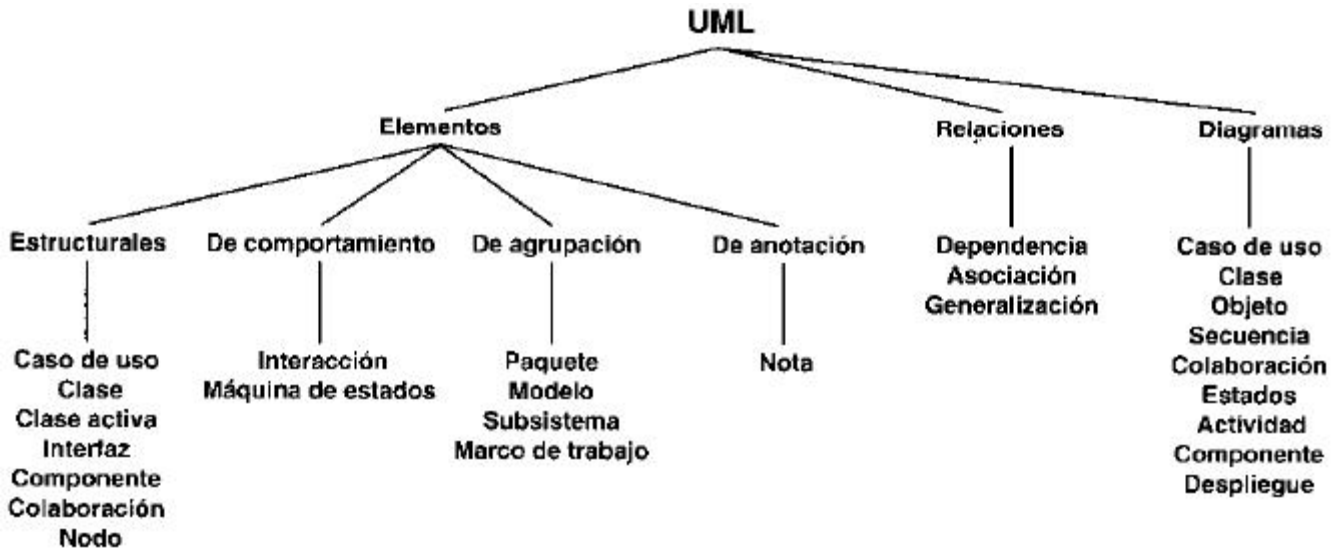


Ilustración 3. El vocabulario de UML

El producto “Pizarrón Virtual” es una aplicación que utiliza técnicas hipertexto por lo que se hace necesario para un mejor desarrollo del mismo incluir algunos aspectos que UML no posee. Estos aspectos están presentes en el lenguaje de modelación OMMMA_L el cual no es más que una extensión de UML.

1.8 OMMMA_L como extensión de UML para aplicaciones con multimedia

El Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA_L) constituye una extensión de UML que favorece el modelado para aplicaciones que utilicen técnicas multimedia.

Una aplicación de este tipo es un sistema de software interactivo con diferentes tipos de medias y que incluye una determinada funcionalidad con una interfaz de usuario. UML, puede representar la funcionalidad de un sistema a través de diagramas de clases y refinarla en el análisis y diseño con los diagramas de comportamiento, pero este lenguaje no ofrece estereotipos para el modelo de interfaz de usuario ni ofrece ningún artefacto que relacione los objetos de tipo media que presenta este tipo de aplicación con las clases de la misma.

OMMMA_L para solucionar lo anterior crea su propia versión de algunos diagramas presentándolos de una manera más cómoda y más acorde al tipo de aplicación a realizar.

Por ejemplo, para elaborar una aplicación que utilice técnicas hipertexto para presentar la información, es necesario tener en cuenta la estructura lógica donde reúne los objetos de dominio de aplicación y los objetos de los medios de comunicación asociados; esta figura también de una presentación espacial (el diseño) y de un comportamiento temporal que abarca los objetos de los medios de comunicación continuos y los requisitos de ejecución en tiempo real; además posee un control interactivo que tiene lugar en el manejo de eventos de interacción con el usuario.

Todo esto OMMMA_L lo representa a través diagramas basándose siempre en su fuente inspiración UML. Este último agrupa sus diagramas ya antes mencionados en cuatro categorías: diagrama de casos de uso, diagramas estructurales, diagramas de comportamiento y diagramas de implementación, siendo el segundo y el tercero quienes interactúan directamente con las descripciones hechas anteriormente.

Para OMMMA – L podemos modelar la estructura lógica a través de diagramas de objetos y clases; la distribución espacial de media puede ser descrita a través de un nuevo artefacto propuesto para el lenguaje, el diagrama de presentación; el comportamiento temporal a través de diagramas de secuencia y por último el control interactivo a través de los diagramas de estado.

Los diagramas de clases son el núcleo de un modelo de aplicación orientado a objeto y consisten en clases y asociaciones que describen la estructura de objeto y las posibles interrelaciones estructurales. Para representar la lógica de aplicación y los tipos de media del modelo estático, se representa un diagrama de clases en dos partes vinculadas: una jerarquía de definición de tipos de media que utiliza la herramienta de autor empleada y que están presentes en la aplicación, y el modelo lógico de esta última que comprende clases y sus asociaciones para describir el modelo del dominio. (Ver anexo 1 (Ilustración 30)).

Además en el Diagrama de clases del modelo de objeto se agregan 3 conceptos; un objeto puede ser de tipo:

- escenario: cuando representa un conjunto de pantallas que muestran una información a través de objetos con similar funcionalidad.
- aplicación: cuando agrupa elementos de media y aúna sus funcionalidades como una entidad.
- media: cuando se hace referencia a sonido, texto, imágenes, animaciones, video, botones, etc.

En cuanto al diagrama de presentación este es una adición al lenguaje y es usado para hacer una descripción intuitiva del esquema del arreglo espacial en la interfaz de usuario de las medias y para representar la estructura de presentación auditiva. Esto último se representa mediante un rectángulo fuera del área de diseño especificando el canal de ejecución cuando el sonido no viaja en los dos canales habituales. La representación de las medias se hace en el área de diseño y de la forma que se muestra en el anexo 1 (Ilustración 31). En el anexo 1 (Ilustración 32) se muestra un ejemplo de un diagrama de presentación.

1.9 Rational Rose

La selección de RUP como metodología para el desarrollo del software y del lenguaje OMMMA_L como extensión de UML, nos guían hacia la utilización de la herramienta Rational Rose.

Rational es la herramienta UML desarrollada por Rose y es posiblemente la más utilizada actualmente en el mundo. Como todos sus demás productos, Rational Rose proporciona un lenguaje común de modelado para el equipo que facilita la creación de software de calidad más rápidamente.

Existen varias razones por las que utilizar Rational Rose a la hora de desarrollar un sistema, algunas de las cuales se citan a continuación:

- Comunicación: Todos los miembros del equipo de trabajo, desde los analistas hasta los programadores, trabajarán con un lenguaje común de Modelado Visual (en este caso OMMMA_L como lenguaje extendido de UML). Esto asegura que todos entenderán los requerimientos del proyecto.

- Administración de la complejidad: El modelado visual de Rational Rose permite a los diseñadores de software navegar fácilmente a través de proyectos complejos logrando, en base a sus necesidades, visualizar desde cuadros de imágenes que muestran el proceso genérico del proyecto, hasta observar diagramas que plasman detalles de bajo nivel en la arquitectura de la aplicación. Al poder ver un anteproyecto desde el inicio surgirán pocos errores en el proyecto.

Rational Rose enmarca los requerimientos y variables del diseño, permitiendo crear un ciclo de desarrollo más pequeño, robusto y adaptable.

- Reusabilidad: Mediante el modelado visual se pueden crear componentes (modelos), que al salvarlos pueden ser compartidos y reutilizados por varios proyectos, permitiendo que los cambios sean fácilmente incorporados a proyectos existentes.
- Desarrollo basado en componentes: El desarrollo basado en componentes se ha convertido en el proceso de diseño más efectivo, los usuarios de Rational Rose pueden modelar sus componentes e interfaces sólo haciendo un “arrastrar y soltar” de los componentes del sistema hacia el diagrama de componentes.

Además, esta herramienta permite hacer análisis, modelado, diseño para el producto de software ayudando a los equipos de desarrollo a construir, integrar, extender, modernizar e implantar software y sistemas basados en software.

1.10 Macromedia Flash 8

Las herramientas de autor son aplicaciones informáticas que posibilitan elaborar sistemas hipermedia a través de un entorno de trabajo que permite una programación basada en iconos, objetos y menús de opciones. Pueden ser utilizadas en el desarrollo de programas educativos ya que permite la creación de aplicaciones en las que, de forma sencilla y rápida se tiene la posibilidad de cambiar el flujo de la información según las necesidades del alumno, relacionar palabras, incluir cuestionarios y marcadores que evalúen los conocimientos alcanzados, activar animaciones y vídeos explicativos, incorporar sonidos y lenguaje hablado, etcétera (18)

Uno de los requisitos que solicitó el cliente para la elaboración del producto es que este se realizase con Macromedia Flash que es la herramienta líder para realizar aplicaciones multimedia interactivas, animaciones y generación de contenidos en los desarrollos Web- utilizando estrategias avanzadas de desarrollo pensadas para profesionales. Esta es una de las herramientas más usadas mundialmente en la actualidad para la producción de sistemas hipermedias en el mundo profesional. Sus diferentes versiones, la facilidad de manejo y la inclusión de gran número de herramientas de producción son las bases fundamentales con las que cuenta este programa. (19)

Las versiones valoradas para proponer la implementación del producto fueron Flash MX 2004 y Flash 8 debido a que estas son de interfaz de fácil aprendizaje y manejo, y los programadores tienen conocimientos de las herramientas y del lenguaje utilizado en éstas. Se falló a favor del Flash 8 pues esta versión es una potente herramienta creada por Macromedia que ha superado las mejores expectativas de sus creadores. Las posibilidades de Flash son extraordinarias pero cada nueva versión ha mejorado a la anterior, y la versión seleccionada no ha sido menos.

(Ilustración 4)



Ilustración 4. Logotipo de la herramienta Macromedia Flash 8

Entre las ventajas fundamentales que ofrece esta herramienta se encuentra:

- **Diseños más atractivos:** permite el uso de efectos visuales que facilitarán la creación de animaciones, presentaciones y formularios más atractivos y profesionales, así mismo, pone a disposición mecanismos para hacer este trabajo más cómodo y rápido, tales como la existencia de filtros y modos de mezcla añadidos en esta versión.
- **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, permite también la selección de configuraciones preestablecidas para textos dinámicos y estáticos.

- **Bibliotecas integradas:** Ahora se buscar rápidamente cualquier objeto existente en las películas creadas, navegando por las bibliotecas de todos los archivos abiertos desde un único panel.
- **Mayor potencia de animación:** 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.
- **Asistente de ActionScript:** El Asistente de ActionScript se había eliminado en la versión anterior, pero se ha vuelto a recuperar, y de forma mejorada, en esta última.

Además, esta versión también posibilita crear efectos de profundidad y de velocidad que hasta ahora solo se podían que generar con Photoshop y otros programas. (20)

Las herramientas que ofrece Macromedia flash tienen un gran potencial, pero insuficiente para lo que reclaman los actuales desarrolladores. A raíz de esto Macromedia decide sacar un código que facilite el trabajo y amplíe los horizontes existentes hasta entonces, es aquí donde nace ActionScript, un lenguaje que permite trabajar dentro de clips de películas. (21)

1.10.1 ActionScript

ActionScript es un lenguaje de programación orientada a objetos nativo de Flash que especifica de forma explícita tipos de datos para variables, parámetros de función y tipos de devolución de funciones. Entre las características que lo distingue se encuentran:

- Las **Acciones**, que son sentencias que ordenan al reproductor ejecutar una acción durante la reproducción de la película;
- Los **Argumentos**, que son parámetros que modifican las acciones de una función;
- Las **Clases**, que son objetos nuevos definidos por el programador. Tienen que ser definidos dentro de una función constructora;
- Los **Constructores**, cuya función constructora sirve para definir un nuevo objeto, con sus propiedades y métodos.

Además, también presenta:

- **Gramática**: comandos en idioma inglés, muy parecidos a los lenguajes BASIC.
- **Sintaxis**: construcción sencilla de comandos, se escriben casi como el inglés.
- **Funciones**: conjunto de comandos que realizan tareas complejas.
- **Programación estructurada**.
- **Orientación a objetos**.

1.11 Soporte educativo

1.11.1 Características de la audiencia

Cuando se desea realizar un producto de alta calidad y que cumpla con los objetivos para los que se concibió, lo primero que se define es hacia quién va dirigido el mismo. Es necesario y muy importante analizar qué características presenta el usuario final que va a consumir el producto que se va a elaborar, puesto que es solamente mediante una detallada comprensión de la audiencia que es posible concebir y construir mensajes y acciones que lleguen a esas personas y que generen en su pensamiento y en sus acciones la respuesta deseada. (22)

Los destinatarios finales para los cuales va dirigido el producto son jóvenes de ambos sexos entre los 15 y 18 años. Presentan un nivel de desarrollo cognitivo concreto avanzado y un nivel educacional: Medio Diversificado y Profesional. Además, poseen conocimientos previos básicos sobre algoritmos,

conocimientos generales de Matemática, Informática Básica y Álgebra así como capacidades generales de lógica matemática y software de aplicación.

Esta audiencia no posee discapacidades y sí intereses educativos y actitudes de aceptación, interés y motivación.

Estos jóvenes no cuentan en la escuela con ningún medio de enseñanza didáctico apoyado en la TICs para su proceso de aprendizaje y la materia de Técnicas de Resolución de Problemas les es muy compleja a la hora de estudiar.

1.11.2 Modelo pedagógico

La concepción del producto “Pizarrón Virtual” se encuentra bajo un enfoque constructivista, o sea, se guía su elaboración sustentando la teoría constructivista.

La misma fue fruto de estudios realizados por Jean Piaget y propone que el aprendizaje se construye por el individuo y es él quien decide su propia arquitectura (26). Es importante observar que el Constructivismo en sí mismo no sugiere un modelo pedagógico determinado (se trata de un modelo pedagógico). De hecho, el Constructivismo describe cómo sucede el aprendizaje, sin importar si el que aprende utiliza sus experiencias para entender una conferencia o intenta diseñar un aeroplano. En ambos casos, la teoría del Constructivismo sugiere que construyen su conocimiento. Para ello es necesario el pensamiento activo y la implicación en la experimentación y la resolución de problemas. Internet y sus recursos amplían la capacidad de interacción personal con estos elementos y hoy en día esta teoría propone un cambio sustancial en la escuela: “un cambio en los objetivos escolares acorde con el elemento innovador que supone el ordenador”.

“Pizarrón Virtual” es un software educativo donde el aprendizaje lo construye el individuo a partir del intercambio de experiencias con el ambiente que le rodea y el conocimiento se adquiere a través de un proceso dinámico e interactivo. Además, el mismo permite el desarrollo de habilidades para la resolución de problemas, presentando situaciones novedosas que estimulan un nuevo aprendizaje

Con este software se pretende brindarle una herramienta a la Escuela Técnica Comercial “Luís Razetti” acerca de técnicas de resolución de problemas que apoye el proceso de enseñanza y aprendizaje de la asignatura “Algorítmica y Programación”.

Además, se espera como resultados del aprendizaje, que los estudiantes:

- Adquieran los conceptos relacionados con los métodos de resolución de problemas.
- Se motiven a resolver problemas propuestos
- Conozcan la importancia del método algorítmico en la resolución de problemas

También, luego del uso del software deben ser capaces de:

- Reconocer los conceptos básicos relacionados con la resolución de problemas
- Comprender la importancia de las técnicas que se requieren para resolver problemas
- Analizar la importancia del uso del software como herramienta que contribuye a la resolución de problemas

Por otro lado se tiene la perspectiva de que el software:

- haga cambios en conductas como:
 - Conocimientos (ya que es un programa que ofrece a través de una interfaz amigable todos los conceptos y herramientas relacionadas con la resolución de problemas algorítmicos).
 - Destreza (pues brinda un entorno gráfico que permite establecer el diálogo y/o comunicación hombre máquina indispensable para desarrollar las habilidades ofrecidas por el programa).
 - Valores (Incentiva la creatividad y motivación en el alumno).
- brinde una posibilidad interactiva:

Tabla 1 Posibilidad interactiva

Tipo de Interactividad	Justificación
Tutor que Facilita la Adquisición de Conocimientos.	Son potentes recursos didácticos que facilitan los aprendizajes de sus usuarios.
Medio para Desarrollar el Razonamiento.	Mediante actividades el alumno desarrolla la capacidad para seleccionar, organizar los conocimientos, aumentar las relaciones entre las ideas.
Medio para la Aplicación del Conocimiento en Situaciones Nuevas	Facilitan un aprendizaje constructivo, asociando los nuevos contenidos a los conocimientos anteriores.

Lo descrito en la tabla anterior posibilitará:

- Facilitar el intercambio de información alumno aprendizaje permitiendo consolidar los conocimientos mediante la formulación de preguntas
- Valorar las acciones del aprendizaje facilitando el auto-aprendizaje individual
- Desarrollar las acciones de los estudiantes logrando solidez en la asimilación de los conocimientos, habilidades y destrezas

Además, toma en cuenta: los objetivos, destinatarios, operaciones mentales porque la naturaleza del contenido esta relacionado con contenidos curriculares del área de informática.

1.11.3 Naturaleza de Actividades Educativas

Tabla 2 Naturaleza de actividades educativas

Tipo de actividad	Justificación

Tipo de actividad	Justificación
Resolución de Problemas	Los alumnos aprenden contenidos y diversos recursos procedimentales así como estrategias auto-reguladoras sobre cómo afrontar diferentes clases de problemas.
Descubrimiento Guiado	Dotar al alumno de conocimientos que le permitan continuar aprendiendo de una forma guiada y por sí mismo.
Descubriendo Experimenta	Guiar al alumno en el descubrimiento de los conocimientos que complementen su trabajo y desarrollar una actitud crítica.

1.12 Conclusiones

Las características propias de los adolescentes que forman la audiencia para la cual está destinado “Pizarrón Virtual”; las formas en que ellos hoy en día reciben la información a través los ordenadores con las bondades que brinda Internet, los teléfonos celulares, la televisión, etc.; y los avances de las TICs en el mundo en los últimos tiempos y su introducción en la educación; indican que es oportuno realizar un software educativo como respuesta al problema planteado, haciendo uso de la hipermedia pues entre sus principales ventajas cuenta con una navegabilidad intuitiva y una interfaz amena y acorde con las perspectivas de estos estudiantes.

Por otro lado para el desarrollo de la propuesta, RUP es la metodología ideal por ser interactivo e incremental, guiado por los casos de uso y centrado en la arquitectura. En este caso es OMMMA_L el lenguaje seleccionado para modelar dicha metodología pues como extensión de UML ofrece sus mismas comodidades añadiéndole o modificándole diagramas que permiten un mejor modelado para este tipo de aplicación.

CAPÍTULO 2

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1 Introducción

En este capítulo se realiza una modelación del entorno donde se usará la aplicación como respuesta al problema planteado. Esto se hace a través del Diagrama del Modelo de Dominio y de la identificación de los principales conceptos que lo integran; además se incluye un mapa de navegación que le brinda una idea al usuario de cómo interactuara la aplicación.

También se enumeran los requisitos funcionales y los no funcionales por los cuales se regirá el sistema propuesto, posibilitando estos hacer una concepción general del mismo e identificar mediante el Diagrama de Casos de Uso las relaciones entre sus actores y los casos de uso.

2.2 Modelo del Dominio

El modelo de dominio es una de las alternativas que brinda RUP para la identificación de requisitos y la comprensión del contexto cuando existe poca estructuración en los procesos de negocio, y con la que se le puede mostrar al usuario de manera visual los principales conceptos que se manejan en el dominio del sistema, sus partes y sus relaciones. Esto les permite a todos los que de alguna manera están involucrados en el proceso de desarrollo del producto manejar un vocabulario común que posibilite el entendimiento del contexto en que se sitúa el sistema. (23)

Este modelo se realiza a través de un diagrama de clases de UML simplificado, en el cual se representan las clases conceptuales que pueden intervenir en el sistema y sus asociaciones preliminares, así como los objetos más importantes en el mismo. Estos objetos del dominio representan “cosas” que existen o los eventos que acontecen en el medio en el que se desenvuelve la aplicación.

(23)

2.2.1 Diagrama del Modelo del Dominio

(Ilustración 5)

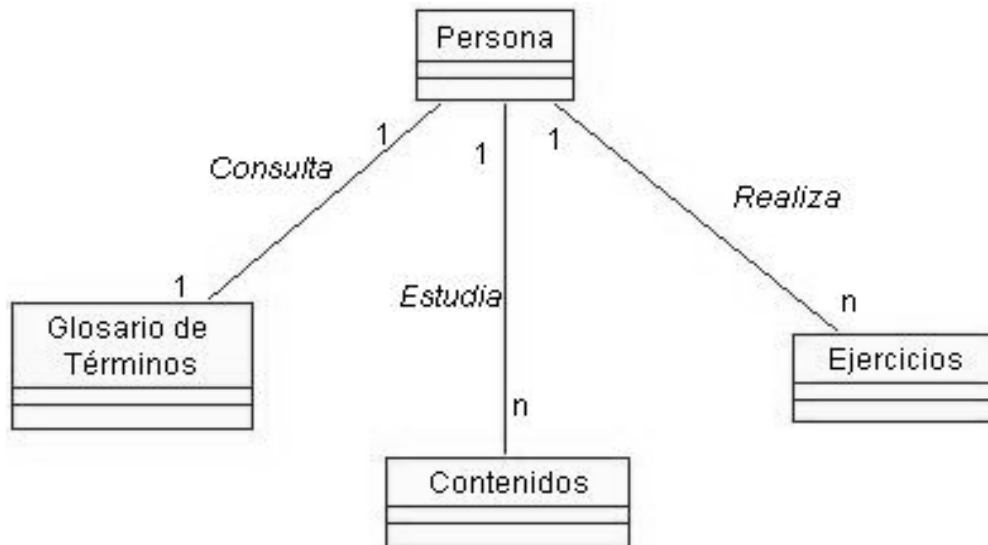


Ilustración 5. Modelo de Dominio

2.2.2 Conceptos

A continuación se presentan los conceptos utilizados en el diagrama del modelo de dominio:

- **Persona**, hace referencia al objeto que interactúa con la aplicación, puede ser un profesor, un estudiante o cualquiera otra.
- Se le denomina **Glosario de Términos** al objeto que contiene un grupo de términos utilizados en la asignatura con su significado.
- **Contenidos** es el objeto que contiene como indica su nombre los contenidos que el usuario debe estudiar. Estos incluyen: Conceptos, Técnicas y Análisis de Datos.
- Se nombra **Ejercicios** al objeto que contiene un conjunto de ejercicios para que el usuario se evalúe en los contenidos estudiados.

2.3 Diagrama de Navegación

El siguiente mapa de navegación es un artefacto creado para brindarle al usuario una idea a grandes rasgos de la forma en que interactuará la aplicación. Este muestra los enlaces entre los distintos contenidos del producto. En este nivel de abstracción, sólo es de interés especificar qué pantallas de forma general conformarán mapa de navegación y desde dónde serán alcanzables. Es válido aclarar que en el sistema presentará una navegación global, o sea, se podrá acceder desde cualquier pantalla a cualquier otra mediante los botones de acceso, incluyendo el glosario de términos y la salida de la aplicación a los cuales se tendrá acceso desde cualquier ubicación en el sistema.

(Ilustración 6)

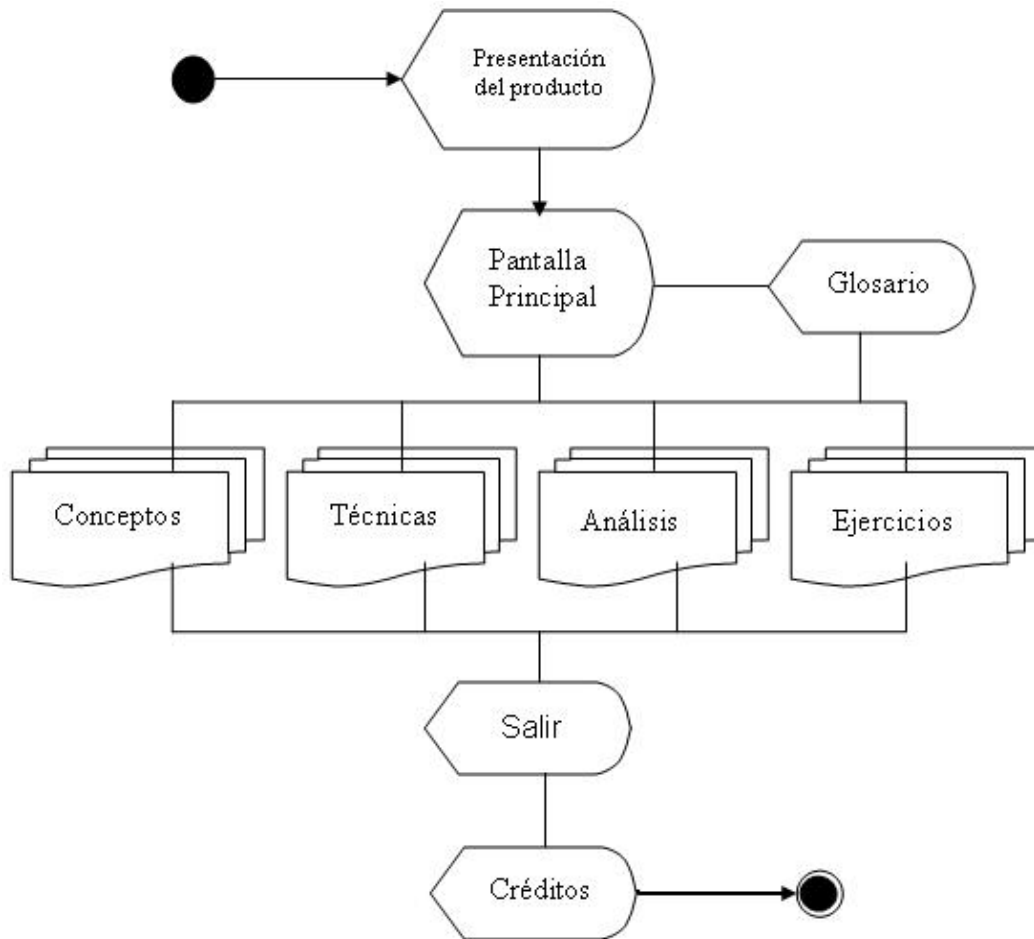


Ilustración 6. Mapa de navegación

2.4 Descripción del sistema propuesto

2.4.1 Descripción de la funcionalidad.

Un proyecto no puede ser exitoso sin una descripción detallada, correcta y exhaustiva de los requerimientos. Estos definen lo que debe hacer un sistema y la forma en que debe hacerlo.

Un requisito (o requerimiento) es una característica de diseño, una propiedad o un comportamiento de un sistema. Los requisitos constituyen la descripción de los deseos o de las necesidades de un producto. (23)

Cuando se enuncian los requisitos de un sistema, se está creando un contrato entre los elementos externos al sistema y el propio sistema, que establece lo que se espera que el sistema haga y las propiedades que tendrá. En otras palabras, los requerimientos son cualidades y capacidades que son especificadas por el cliente y que debe tener el sistema. Estos requerimientos se pueden clasificar en:

- Requerimientos funcionales: representan las funciones que el sistema será capaz de realizar.
- Requerimientos no funcionales: indican las propiedades o características que debe presentar el sistema y que de alguna forma lo limitan.

1. Requisitos funcionales del sistema

2. Permitir al usuario ejecutar la aplicación
3. Permitir controlar el audio del sistema
4. Permitir el acceso al Glosario de Términos
5. Mostrar las palabras del glosario de términos con su significado.
6. Permitir la navegación por toda la aplicación
7. Mostrarle al usuario su ubicación dentro del sistema
8. Permitir la salida del sistema cuando esta sea solicitada
9. Visualizar los créditos del producto a ser confirmada la salida
10. Mostrar el contenido dividido por temas: Conceptos, Técnicas, Análisis de Datos, Ejercicios
11. Mostrar los ejercicios por temas
12. Permitir al usuario interactuar con el ejercicio en curso

1. Requisitos no funcionales del sistema

Apariencia o Interfaz externa

1. Se mostrará el nombre de las instituciones participantes
2. Se mostrará el nombre del producto
3. El texto será de color blanco como escrito con tiza
4. Los textos que identifican interfaces u otros contenidos serán de color amarillo con iluminación en blanco
5. La imagen predominante será un pizarrón de fondo verde con marco marrón
6. Las opciones de servicios además de tener su icono identificador tendrá el texto que muestre la opción en cuestión con su función descrita en una palabra para un reconocimiento rápido por el usuario
7. El vocabulario que se utilizará será el español exclusivamente además de las palabras técnicas propias de la materia
8. Las medias a visualizar siempre se hará utilizando la misma área de la interfaz
9. La voz de la profesora será voz de una mujer joven

Navegación

10. Desde cualquier pantalla se podrá acceder a cualquier módulo
11. Se podrá abandonar el programa desde cualquier pantalla, posterior a una confirmación del usuario
12. Desde cualquier lugar se tendrá acceso al glosario.

Software

13. Plataformas de usuario:
 - Plataforma WS / Windows NT (Es una plataforma formada por uno o varios procesadores Intel, con un sistema operativo de Windows, pero específico para trabajar en red)

- Plataforma PC / Windows
- Plataforma WS / Linux
- Plataforma PC / Linux

Hardware

Los requerimientos que a continuación se muestran son los mínimos que debe tener la Terminal para que se pueda ver la aplicación:

- Procesador 486DX/66 MHz o superior.
- 16 MB de memoria, a más memoria mayor rendimiento.
- Monitor VGA o superior.
- Ratón Microsoft o compatible.

Implementación

14. El lenguaje de programación será ActionScript

15. La herramienta de desarrollo de la aplicación serán: Macromedia Flash

Soporte

16. La Terminal donde se ejecutará la aplicación debe tener entre sus dispositivos una tarjeta de video, una tarjeta de sonido y aditamento para la reproducción del sonido

Rendimiento

17. El tiempo de visualización de las medias no debe exceder los 5s

18. El tiempo de ejecución de un hipervínculo no debe exceder los 5s

Servicios generales

19. El audio, inicio, glosario, salir, fecha; siempre estarán visibles al cliente durante toda la navegación que realice por el software.

2.5 Modelo de Casos de Uso

El modelo de casos de uso describe la funcionalidad propuesta del nuevo sistema y ayuda a los clientes, usuarios y desarrolladores a llegar a un acuerdo sobre cómo utilizar el mismo y las condiciones y posibilidades que debe cumplir. Contiene los actores y casos de uso del sistema y sus relaciones.

Los actores constituyen entidades que representan roles que son desempeñados por personas (pueden ser sistemas) que de alguna manera participan en la historia de un caso de uso o que interactúan con el sistema por decirlo de otra manera.

Los casos de uso representan los requisitos funcionales del sistema y son un conjunto de actividades que un sistema lleva a cabo y que engendra un beneficio para un actor.

Estos actores, casos de uso y sus relaciones se representan en un diagrama de casos de uso el cual modela el comportamiento del sistema en términos de interacciones de agentes externos (actores) que interactúan con el sistema.

2.5.1 Actores

Actor: Usuario

Justificación: Representa a la persona que va a utilizar el sistema en busca de información o de evaluarse en los contenidos expuestos en el mismo.

2.5.2 Casos de Uso del Sistema

Tabla 3 Casos de Uso del sistema

CUS #	Nombre	Prioridad
CUS 1	Cargar _presentación	Secundario
CUS 2	Controlar _locución	Secundario
CUS 3	Consultar _glosario	Crítico
CUS 4	Controlar _navegación	Crítico
CUS 5	Salir _del _sistema	Crítico
CUS 6	Mostrar _contenidos _de _ módulos	Crítico
CUS 7	Interactuar _con_ ejercicio	Crítico

2.5.2.1 Diagrama y descripción textual de los Casos de Uso

(Ilustración 7)



Ilustración 7. Diagrama de Casos de Uso del Sistema Cargar _presentación

Tabla 4 Caso de uso Cargar _ presentación

CUS 1 Cargar _ presentación	
Actor: Usuario (inicia)	
Propósito: Ejecutar la presentación de producto	
Descripción: El usuario accede a la aplicación y la ejecuta para ver el producto. Esta presentación general es de obligatoria visualización por parte del cliente. Este último no podrá interrumpirla con ninguna acción. Cuando termine la presentación el sistema inmediatamente muestra la pantalla principal del software desde donde el usuario operará en la multimedia.	
Precondiciones:	
Poscondiciones: La pantalla de presentación se verá una sola vez puesto que es la inicialización de la aplicación. Se presenta la pantalla principal del software	
Flujo normal de eventos	
<i>Acciones del Actor</i>	<i>Respuesta del sistema</i>
1 Da clic en el icono de la aplicación	1.1 Muestra la presentación del producto 1.2 Muestra la pantalla principal de Caso de Uso
Referencias: R1	
Prototipo de Interfaz de usuario: Ver Anexo 2 (Ilustración 33)	

(Ilustración 8)

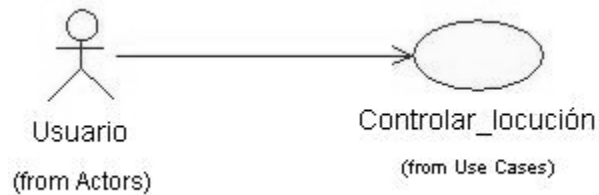


Ilustración 8. Diagrama de Casos de Uso del Sistema Controlar _locución

Tabla 5 Caso de uso Controlar _locución

CUS 2 Controlar _locución	
Actor: Usuario (inicia)	
Propósito: Activar o desactivar la locución del sistema	
Descripción: El caso de uso inicia cuando el usuario oprime el botón del audio. El sistema analiza en que estado se encuentra el mismo y pasa al contrario finalizando así el caso de uso.	
Precondiciones: Locución activa o locución inactiva	
Poscondiciones: Locución inactiva o locución activada	
Flujo normal de eventos	
<i>Acciones del Actor</i>	<i>Respuesta del sistema</i>
1 Da clic en el icono del sonido para detener la locución	1.1 Detiene la locución
Flujo alterno de eventos	
1 Da clic en e icono del sonido para repetir la locución	1.1 Reinicia la locución
Referencias: R2	
Prototipo de Interfaz de usuario: Ver Anexo 2 (Ilustración 34)	

(Ilustración 9)

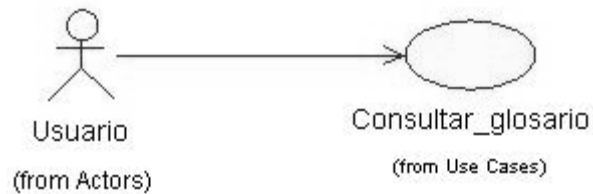


Ilustración 9. Diagrama de Casos de Uso del Sistema Consultar _glosario

Tabla 6 Caso de uso Consultar _glosario

CUS 3 Consultar _glosario	
Actor: Usuario (inicia)	
Propósito: Ver los significados de las palabras que pueden resultar complejas para el usuario.	
Descripción: El usuario selecciona la opción de ver el glosario de términos en busca del significado de una palabra. Selecciona la palabra y el sistema muestra su significado.	
Precondiciones:	
Poscondiciones:	
Flujo normal de eventos	
<i>Acciones del Actor</i>	<i>Respuesta del sistema</i>
1 Da clic en el icono del glosario para hacer una consulta	1.1 Muestra las palabras del glosario de términos
2 Selecciona la palabra deseada	2.1 Muestra el significado de dicha

	palabra
3 Presiona el botón regresar	3.1 Muestra la pantalla en que se encontraba antes de consultar el glosario
Referencias: R3, R4	
Prototipo de Interfaz de usuario: Ver Anexo 2 (Ilustración 35)	

(Ilustración 10)

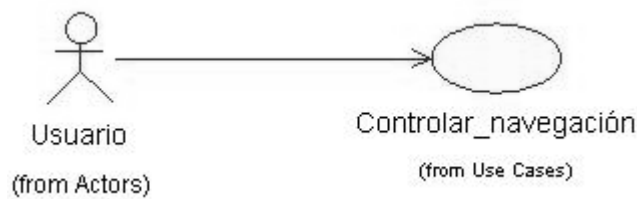


Ilustración 10. Diagrama de Casos de Uso del Sistema Controlar _navegación

Tabla 7 Caso de Uso Controlar _navegación

CUS 4 Controlar _navegación	
Actor: Usuario (inicia)	
Propósito: Permitir la navegación entre pantallas.	
Descripción: El usuario a través de la selección de botones de navegación o menús selecciona hacia donde quiere dirigirse dentro de la aplicación. El sistema muestra dónde se encuentra el usuario dentro de la aplicación. Este caso de uso termina cuando el usuario se encuentra en el lugar seleccionado.	
Precondiciones:	
Poscondiciones: El usuario solo podrá interactuar con la pantalla en que se encuentra.	
Flujo normal de eventos	
<i>Acciones del Actor</i>	<i>Respuesta del sistema</i>
1 Da clic en el icono que identifica la	1.1 Muestra la pantalla deseada

pantalla a la cual desea ir
Referencias: R5, R6
Prototipo de Interfaz de usuario: _____

(Ilustración 11)

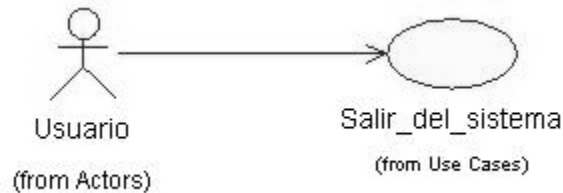


Ilustración 11. Diagrama de Casos de Uso del Sistema Salir _del _sistema

Tabla 8 Caso de Uso Salir _del _sistema

CUS 5 Salir _del _sistema	
Actor: Usuario (inicia)	
Propósito: Permitir que el usuario salga del sistema cuando lo desee.	
Descripción: El usuario selecciona el botón salir del sistema. Inmediatamente se muestra una pantalla de confirmación de salida. En caso de que el usuario confirme su salida se muestran los créditos y se cierra la aplicación. En el caso de que el usuario no confirme la salida del sistema vuelve a la pantalla en que se encontraba.	
Precondiciones:	
Poscondiciones:	
Flujo normal de eventos	
<i>Acciones del Actor</i>	<i>Respuesta del sistema</i>
1 Da clic en el icono de salir para salir de la aplicación	1.1 Muestra una ventana de confirmación de salida

2 Confirma la salida de la aplicación	2.1 Muestra la pantalla de los créditos
	2.2 Cierra la aplicación
Flujo alternativo de eventos	
Acción 2 No confirma la salida	2.1 Muestra la pantalla en que se encontraba
Referencias: R7, R8	
Prototipo de Interfaz de usuario: Ver Anexo 2 (Ilustración 36)	

(Ilustración 12)

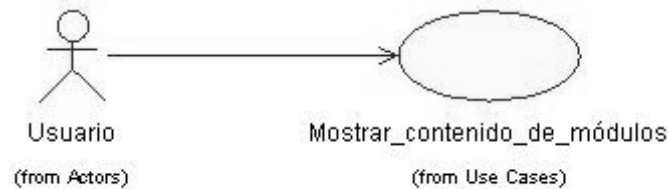


Ilustración 12. Diagrama de Casos de Uso del Sistema Mostrar _contenido _de _módulos

Tabla 9 Caso de Uso Mostrar _contenidos _de _módulos

CUS 6 Mostrar _ contenidos _ de _ módulos
Actor: Usuario (inicia)
Propósito: Mostrar el contenidos de los módulos de la aplicación.
Descripción: El caso de uso inicia cuando el usuario selecciona ver alguno de los módulos que se presentan en la aplicación entre los que se encuentran Conceptos, Análisis, Técnicas y Ejercicios. Luego el sistema se encarga de mostrar el contenido del módulo solicitado.

Precondiciones: El caso de uso cargar presentación tiene que haber concluido.	
Poscondiciones: El usuario se encuentra en el módulo seleccionado y solo podrá interactuar con la pantalla correspondiente al módulo en que se encuentra.	
Flujo normal de eventos	
<i>Acciones del Actor</i>	<i>Respuesta del sistema</i>
Sección "Conceptos"	
1 Selecciona ver el módulo conceptos	1.1 Muestra la pantalla principal del módulo "Conceptos"
2 Selecciona pasar a la siguiente pantalla de este módulo	2.1 Muestra la pantalla dos de este módulo
Sección "Técnicas"	
1 Selecciona ver el módulo Técnicas	1.1 Muestra la pantalla principal del módulo "Técnicas"
2 Selecciona ver "Elementos"	2.1 Muestra los elementos que conforman un algoritmo.
Sección "Análisis"	
1 Selecciona ver el módulo Análisis	1.1 Muestra la pantalla principal del módulo "Análisis"
2 Selecciona ver ejemplo	2.1 Muestra el ejemplo
Sección "Ejercicio"	
1 Selecciona ver el módulo Ejercicios	1.1 Muestra la pantalla principal del módulo Ejercicios donde presentan los temas en que se encuentran agrupados los ejercicios
2 Selecciona el tema "Componentes de un algoritmo"	2.1 Muestra el primer ejercicio de este tema
Flujo alterno de eventos	
Sección "Técnicas"	

Acción 2 Selecciona ver “Expresiones”	2.1 Muestra los tipos de expresiones que presenta un algoritmo
Acción 2 Selecciona ver “Estructuras”	2.1 Muestra las estructuras que presenta un algoritmo
Sección “Ejercicio”	
Acción 2 Selecciona el tema “Tipos de datos”	2.1 Muestra el primer ejercicio de este tema
Acción 2 Selecciona el tema “Contadores y sumadores”	2.1 Muestra el primer ejercicio de este tema
Acción 2 Selecciona el tema “Algoritmos”	2.1 Muestra el primer ejercicio de este tema
Referencias: R9, R10	
Prototipo de Interfaz de usuario: Ver Anexo 2 (Ilustración 37)	

(Ilustración 13)

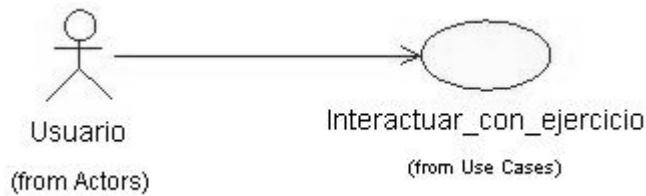


Ilustración 13. Diagrama de Casos de Uso del Sistema Interactuar _con _ejercicio

Tabla 10 Caso de Uso Interactuar _con _ejercicio

CUS 7 Interactuar _con_ ejercicio	
Actor: Usuario (inicia)	
Propósito: Permitir que el usuario interactúe con el ejercicio en curso	
Descripción: El caso de uso inicia cuando el usuario comienza a interactuar con el ejercicio seleccionado. El sistema se encarga entonces de que las opciones brindadas para solucionar el ejercicio sean manipulables. Si el usuario hace correctamente el ejercicio el sistema muestra un mensaje de felicitación y brinda la opción de hacer más ejercicios. En caso contrario indica que hizo mal el ejercicio y brinda algunas posibilidades de repetirlo. De no lograr hacer correctamente el ejercicio en las oportunidades dadas se le indica al usuario que se estudie nuevamente los contenidos.	
Precondiciones: El usuario tiene que haber seleccionado un tipo de ejercicio a realizar. La pantalla muestra el ejercicio seleccionado.	
Poscondiciones:	
Flujo normal de eventos	
<i>Acciones del Actor</i>	<i>Respuesta del sistema</i>
1 Realiza el ejercicio y evalúa	1.1 Felicita por hacer correctamente el ejercicio 1.2 Pregunta si desea realizar otro ejercicio
2 Acepta hacer otro ejercicio	2.1 Muestra otro ejercicio
Flujo alterno de eventos	
Acción 1	1.2 Indica que ha hecho incorrectamente el ejercicio y brinda tres oportunidades más de repetirlo

		De hacerlo bien vuelve a la acción 1.2 De hacerlo incorrectamente orienta estudiarse nuevamente los contenidos y vuelve a la acción 1.2
Acción 1	1.2	Indica que se agotaron los ejercicios de dicho tema
	1.3	Muestra la pantalla principal del módulo ejercicios
Acción 2 Rechaza la oferta	2.1	Muestra la pantalla principal de módulo ejercicios
Referencias: R11		
Prototipo de Interfaz de usuario: _____		

2.5.2.2 Vista Global de los actores y casos de uso del sistema

(Ilustración 14)

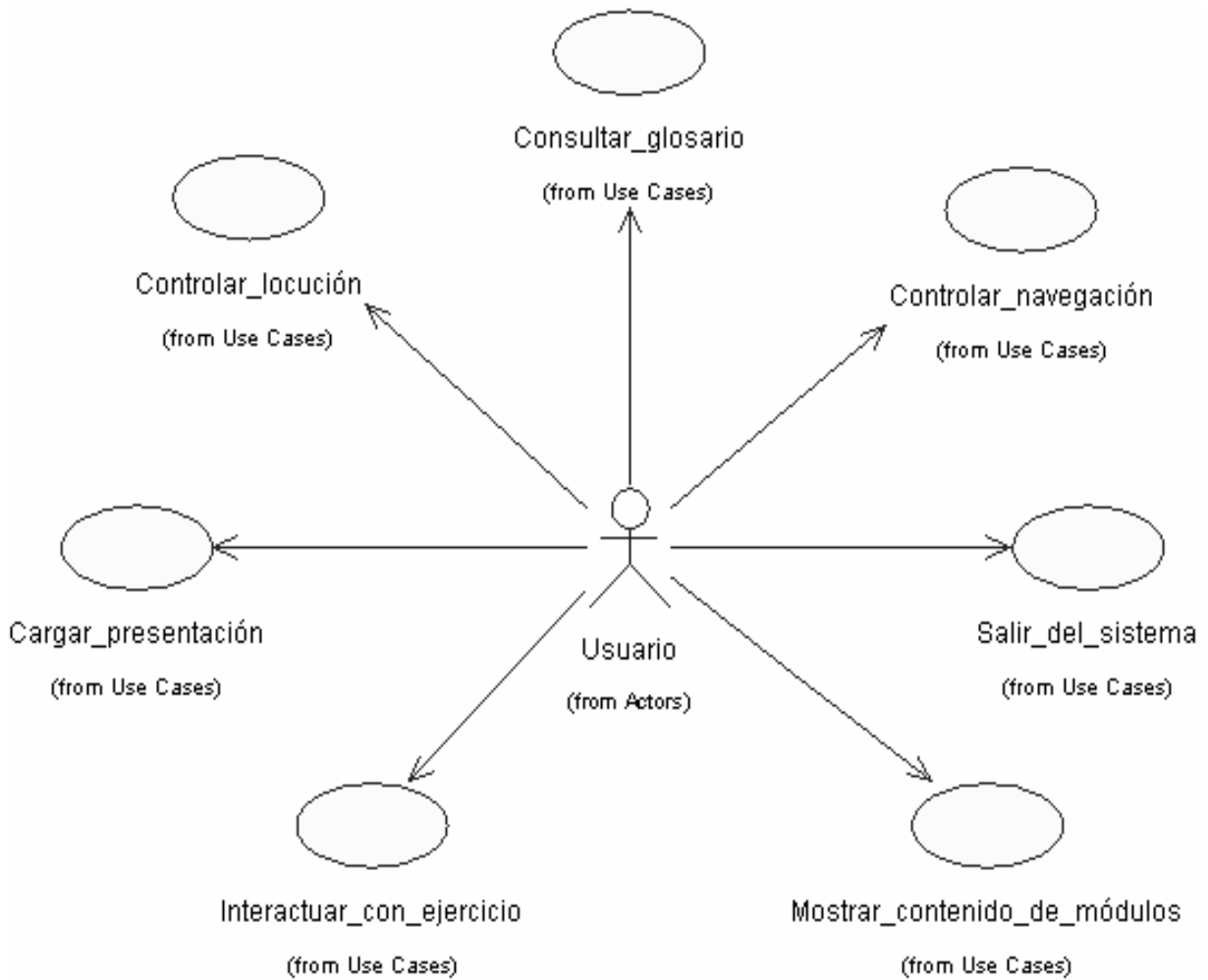


Ilustración 14. Diagrama de Casos de Uso del Sistema

2.6 Conclusiones

“La mayoría de las ideas fundamentales de la ciencia son esencialmente sencillas y, por regla general pueden ser expresadas en un lenguaje comprensible para todos.”

Albert Einstein

Precisamente en este capítulo se estableció un lenguaje común y comprensible para todos los que de una u otra manera están involucrados con el sistema. Esto se hizo mediante la escritura de los requisitos funcionales y no funcionales con los que debe cumplir el producto.

También se realizó un modelo del dominio del entorno donde se presenta el problema que resuelve la aplicación propuesta. En este modelo se presentó el diagrama de dominio, la definición de los 4 conceptos empleados en el mismo y un mapa de navegación que describe los posibles caminos a seguir en la aplicación.

Además, se desarrolló el modelo de casos de uso del sistema a partir de la definición de un actor y siete casos de uso y sus relaciones, representadas en el diagrama de casos de uso del sistema; también se incluyó la descripción textual de estos últimos.

CAPÍTULO 3

CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

3.1 Introducción

A partir de OMMMA_L que fue el lenguaje de modelación seleccionado como extensión de UML para modelar el software “Pizarrón Virtual”, se presentan en este capítulo artefactos que posibilitaron una mejor construcción del sistema.

En él se muestran el diagrama de clases del modelo de objetos y el diagrama de presentación, propuestos por OMMMA_L para el flujo de trabajo de “Diseño”.

Además, también se muestran los diagramas de componentes y de despliegue pertenecientes al modelo de implementación y al modelo de despliegue correspondientemente los cuáles guiarán una mejor implementación del producto.

3.2 Modelo de Diseño

3.2.1 Diagrama de clases del modelo de objetos

El diagrama de clases del modelo de objetos es una de la propuesta de OMMMA_L que utiliza las mismas notaciones que UML pero que además relaciona las medias que se encuentran vinculadas con las clases. Estas medias las sitúan juntas en el diagrama y de forma jerárquica como muestra el ejemplo del anexo 1 (Ilustración 30).

Además, en este diagrama se agregan tres conceptos nuevos, media, aplicación y escenario mencionados y explicados anteriormente en el presente documento.

(Ilustración 15)

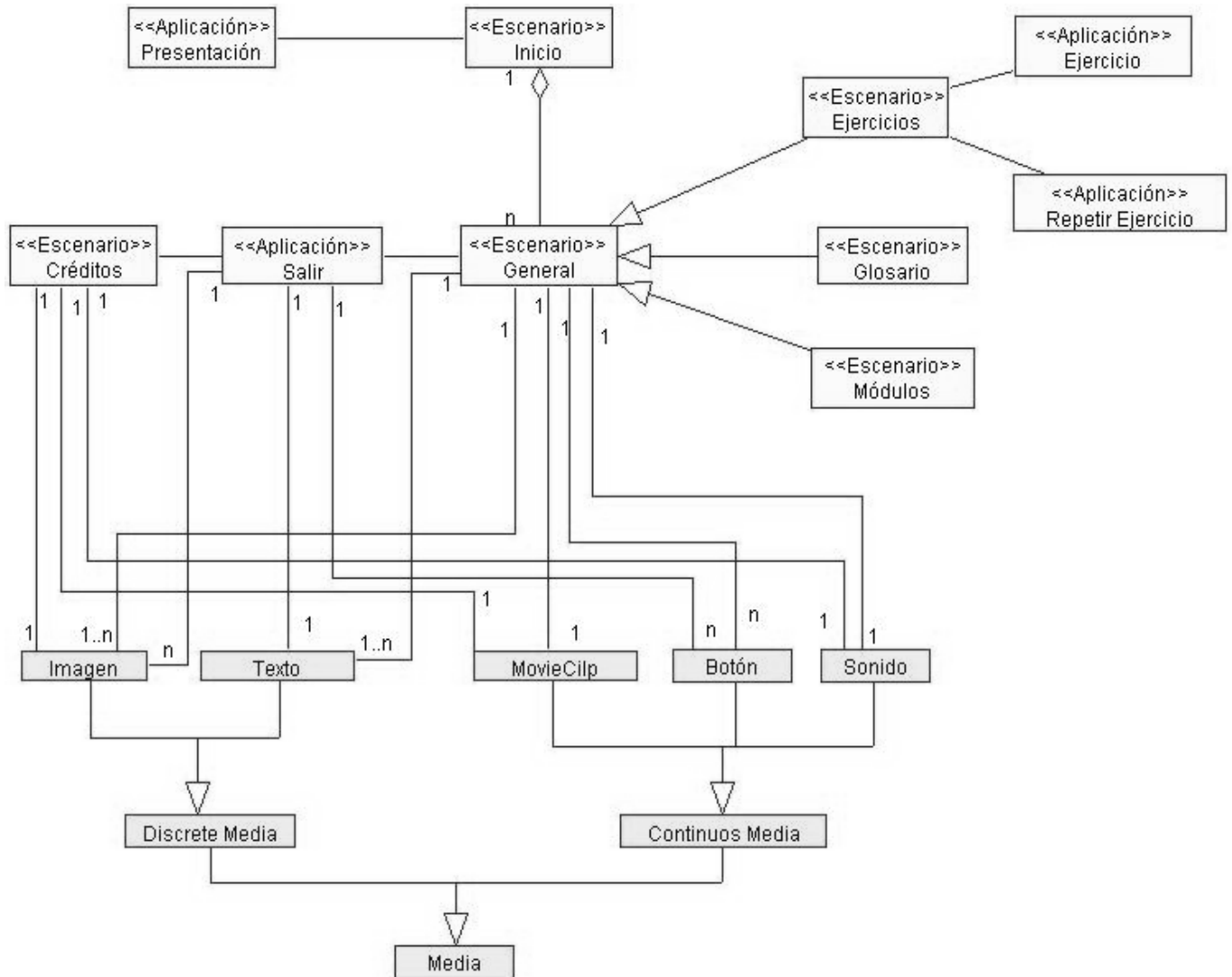


Ilustración 15. Diagrama de Clases del Modelo de Objetos

3.2.2 Diagrama de Presentación

Como anteriormente se había señalado, uno de los diagramas que propone OMMMA_L es el Diagrama de presentación. Este diagrama describe cómo es que se ubican los objetos en la interfaz de la aplicación. Lo principal en este tipo de diagrama es definir el diseño de presentación visual y la estructura de presentación auditiva.

En él se describen dos tipos de objetos:

- Objetos de la visualización: objetos pasivos, usados para presentar el texto, los gráficos, el video, etc.
- Objetos de la interacción: objetos activos que permiten interacciones del usuario y levantan acontecimientos (ej. Botones)

A continuación se muestran los diagramas de presentación que definen las distintas formas en que se va a mostrar la información en las interfaces del software.

(Ilustración 16)

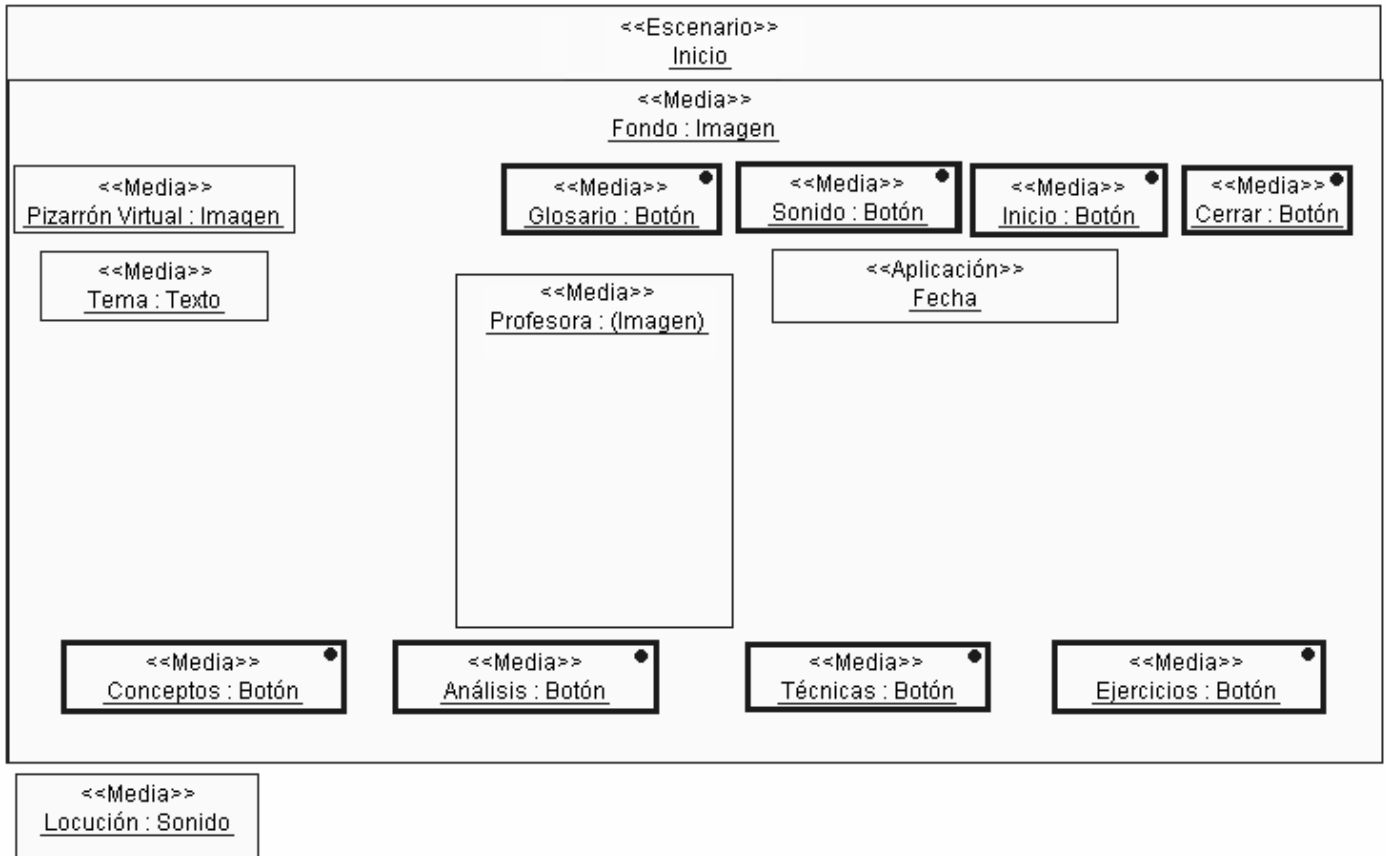


Ilustración 16. Diagrama de presentación Inicio

(Ilustración 17)

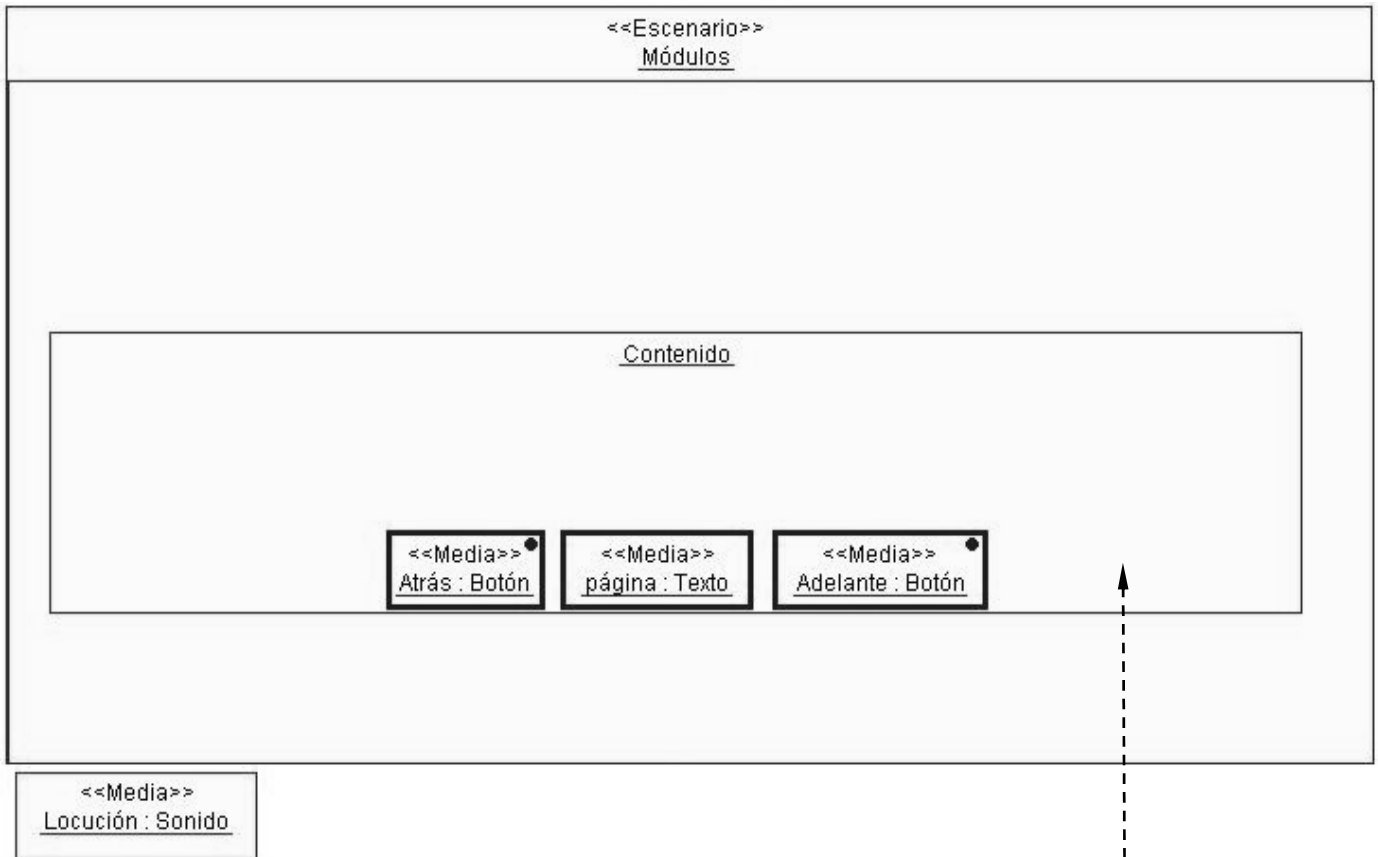


Ilustración 17. Diagrama de presentación Módulos

En esta área se mostrarán todos los contenidos de los módulos mediante textos y palabras calientes. La misma será el área interactiva que es la que va a cambiar en los diagramas de presentación

(

Ilustración 18)

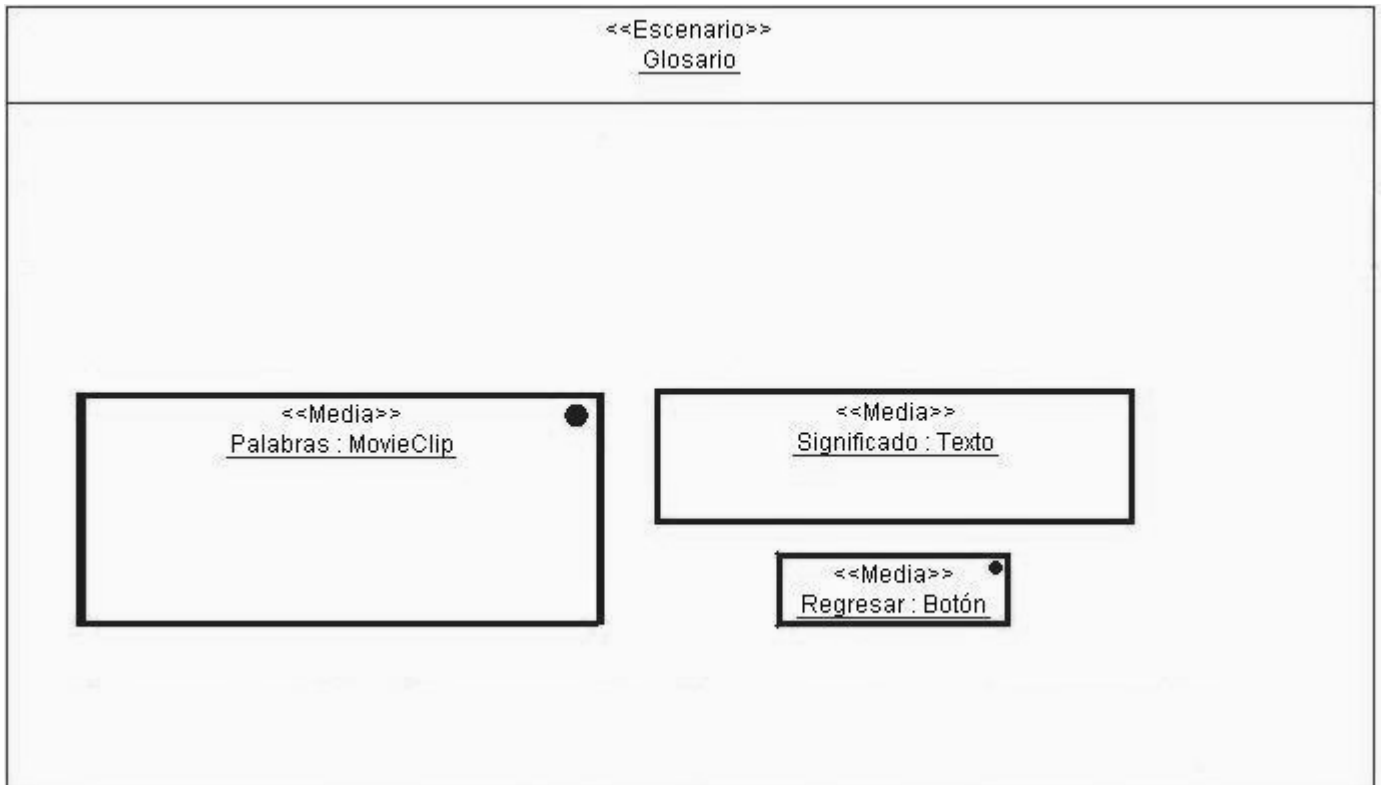


Ilustración 18. Diagrama de presentación Glosario

(Ilustración 19)

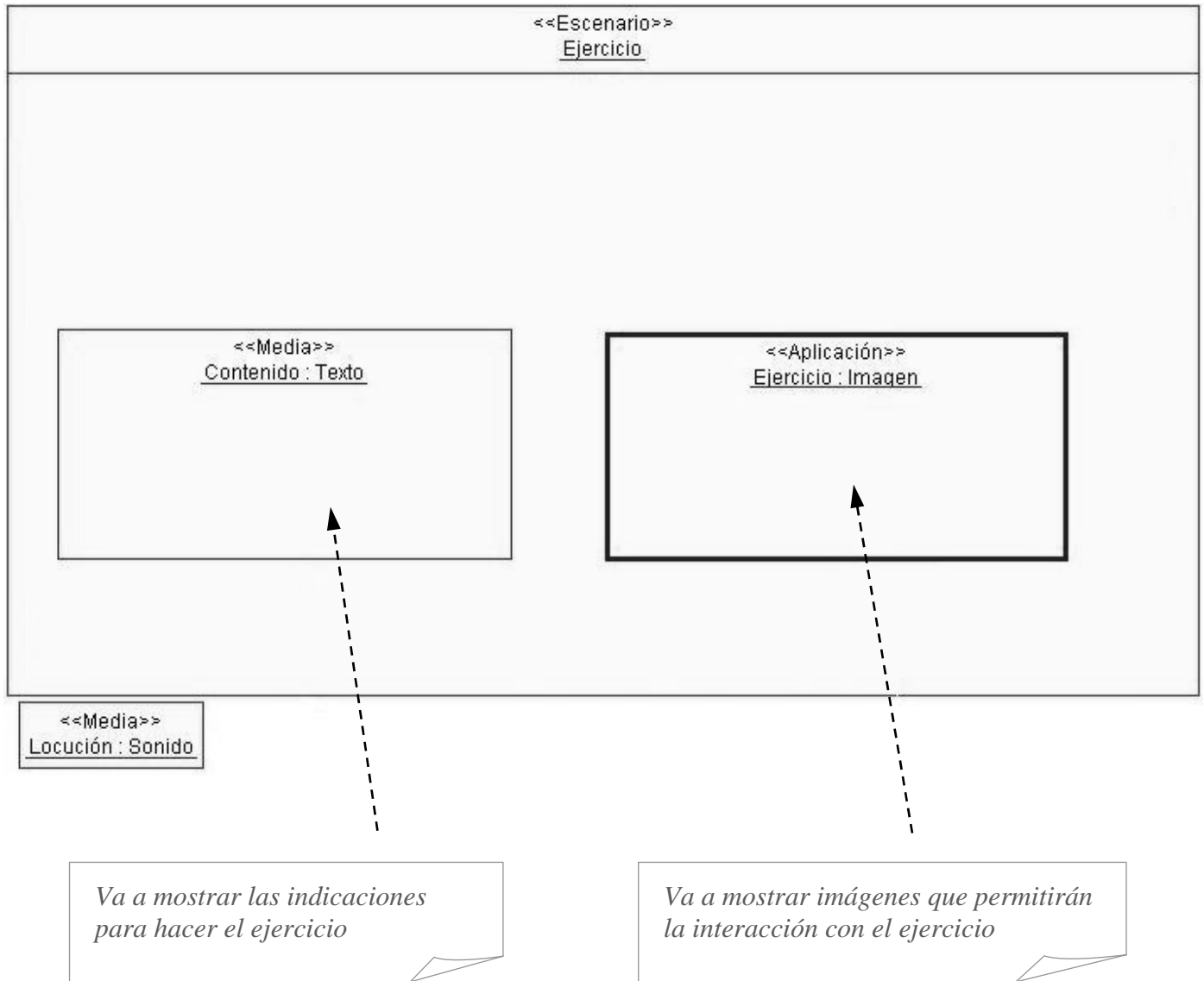


Ilustración 19. Diagrama de presentación Ejercicio

(Ilustración 20)



Ilustración 20. Diagrama de presentación Repetir Ejercicio

(Ilustración 21)



Ilustración 21. Diagrama de presentación Salir

(Ilustración 22)

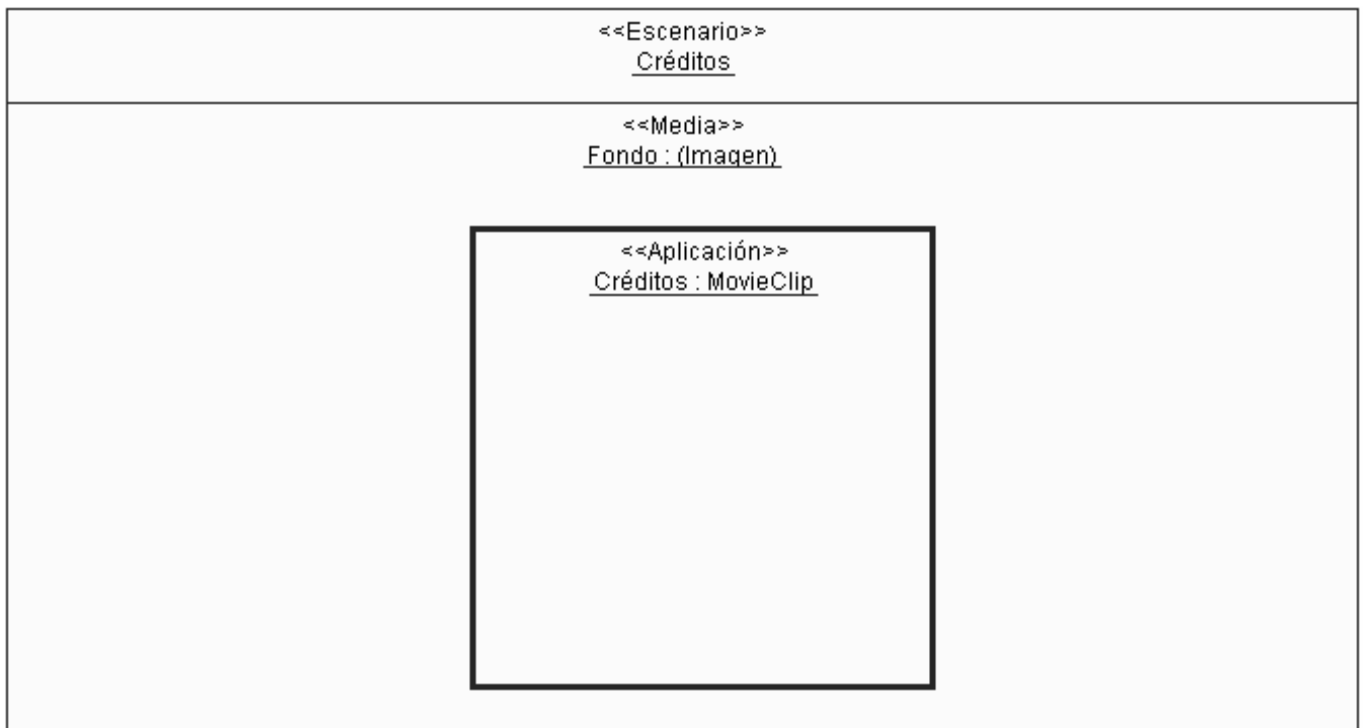


Ilustración 22. Diagrama de presentación Créditos

3.3 Modelo de Implementación

En el modelo de implementación se describe cómo los elementos del modelo de diseño se deben implementar en términos de componentes y cómo se organizan estos de acuerdo con los mecanismos de estructuración disponibles en el entorno de implementación y en el lenguaje de programación utilizado; así como la dependencia entre los componentes (15). Esto se hace a través del diagrama de paquetes y del diagrama de componentes.

El primero constituye la estructuración de los paquetes en los que se pueden agrupar los distintos tipos de componentes y el segundo describe los elementos de software que entran en la fabricación de la aplicación y sus relaciones.

Los componentes pueden ser archivos, paquetes, bibliotecas cargadas dinámicamente, etc. Y para ellos UML tiene definido 5 estereotipos lo cuales no cambia OMMMA_L:

- Ejecutable: Especifica un componente que se puede ejecutar en un nodo.
- Librería: Especifica una biblioteca de objetos estática o dinámica.
- Tabla: Especifica un componente que representa una tabla de una base de datos.
- Archivo: Especifica un componente que representa un documento que contiene código fuente o datos.
- Documento: Especifica un componente que representa un documento

3.3.1 Diagrama de paquetes

(Ilustración 23)

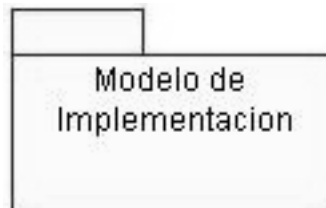


Ilustración 23. Diagrama de paquetes

3.3.2 Diagrama de componentes

En este caso se dividió el diagrama de componentes principal en sub-paquetes para un mejor entendimiento del mismo.

3.3.2.1 Diagrama de componentes general

(Ilustración 24)

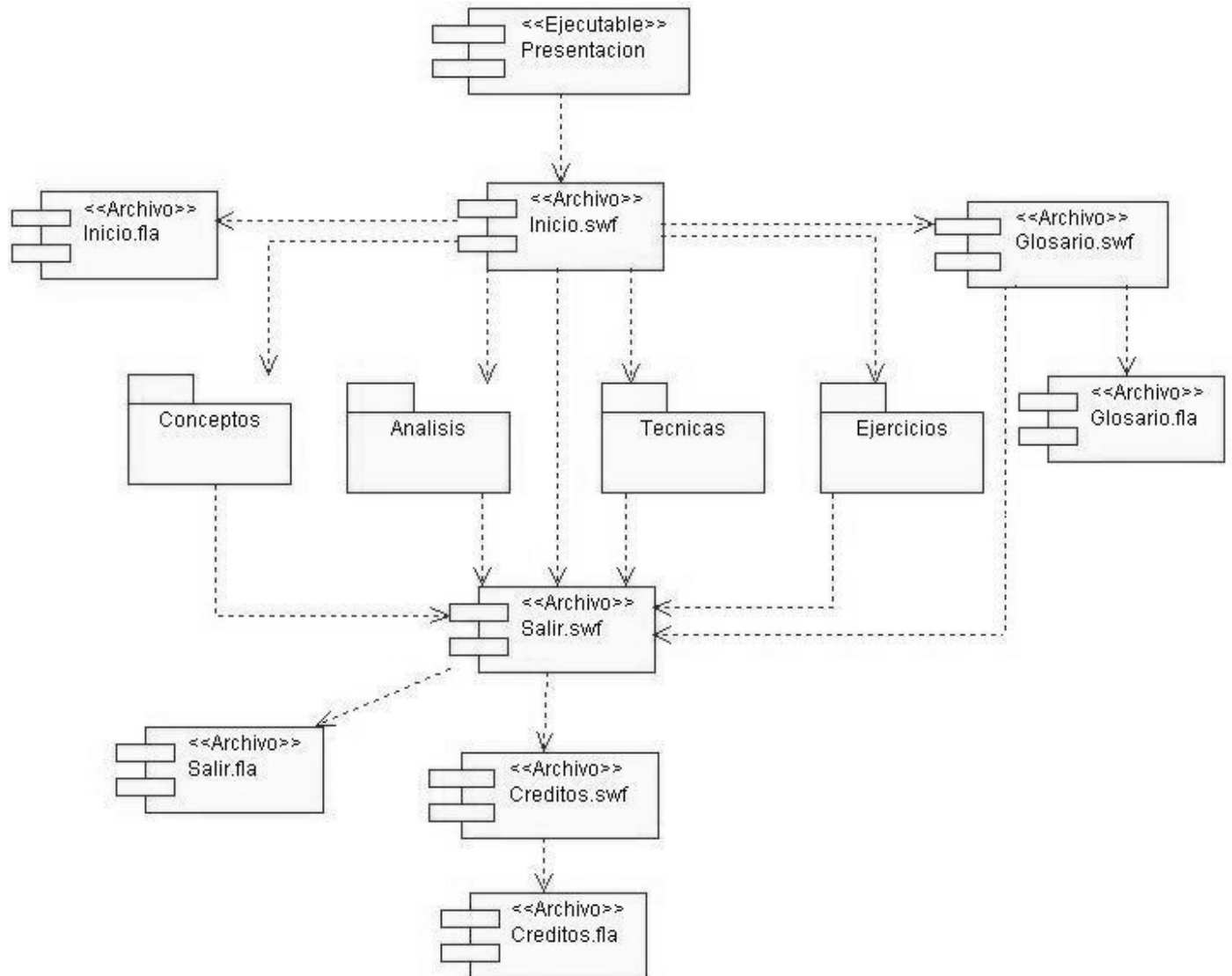


Ilustración 24. Diagrama de componentes general

3.3.2.1.1 Diagrama del paquete "Conceptos"

(Ilustración 25)

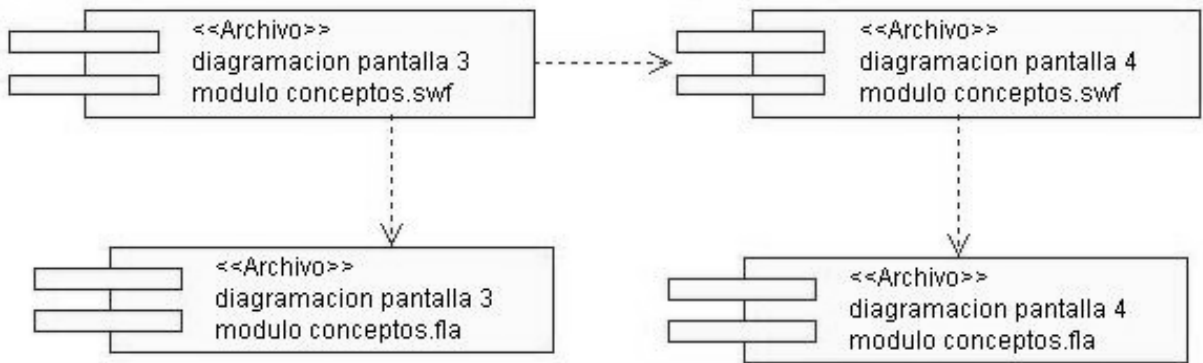


Ilustración 25. Diagrama de componentes del paquete "Conceptos"

3.3.2.1.2 Diagrama del paquete “Técnicas”

(Ilustración 26)

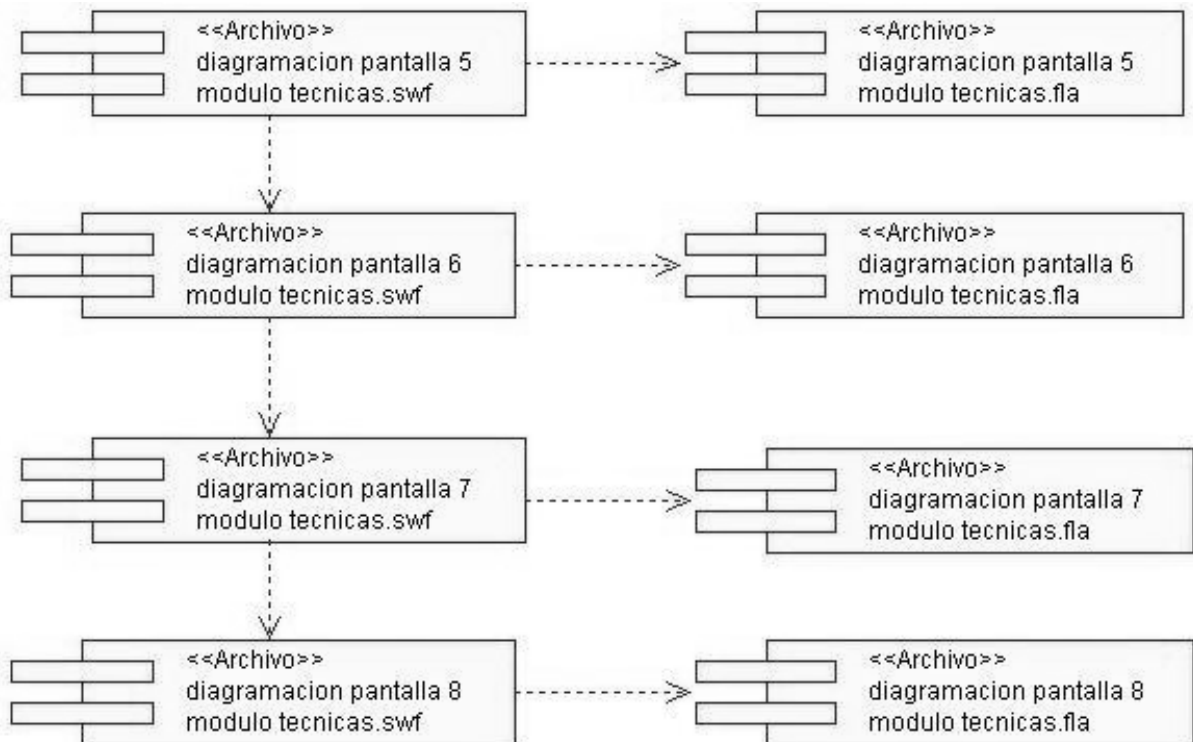


Ilustración 26. Diagrama de componentes del paquete "Técnicas"

3.3.2.1.3 Diagrama del paquete “Análisis”(Ilustración 27)

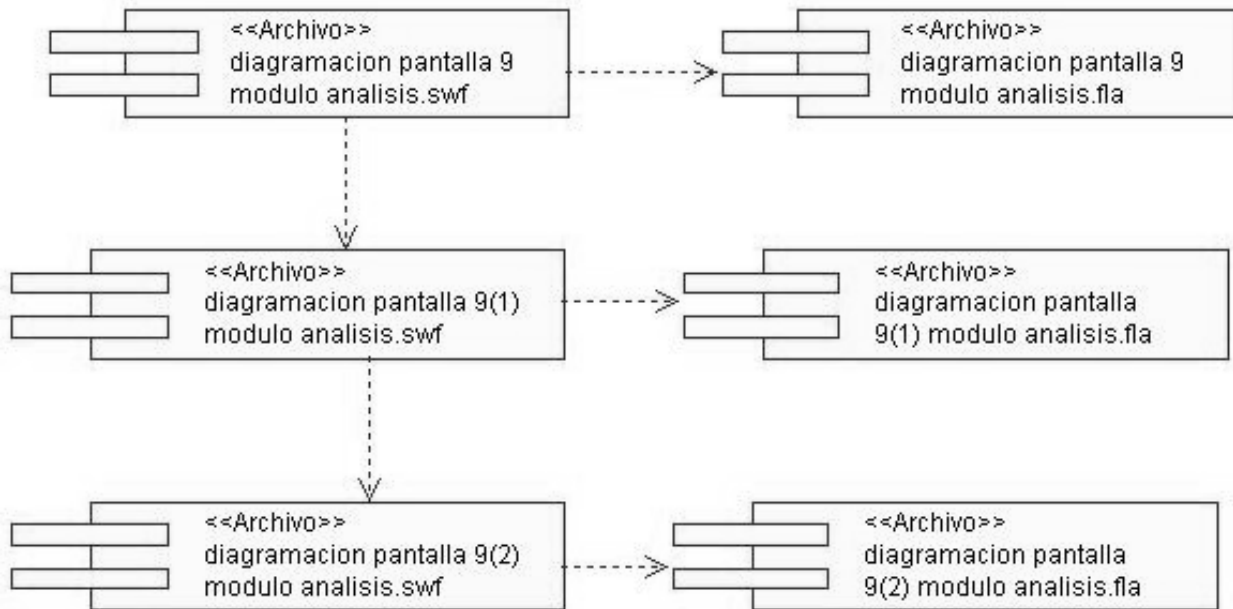


Ilustración 27. Diagrama de componentes del paquete "Análisis"

3.3.2.1.4 Diagrama del paquete “Ejercicios”

(

Ilustración 28)

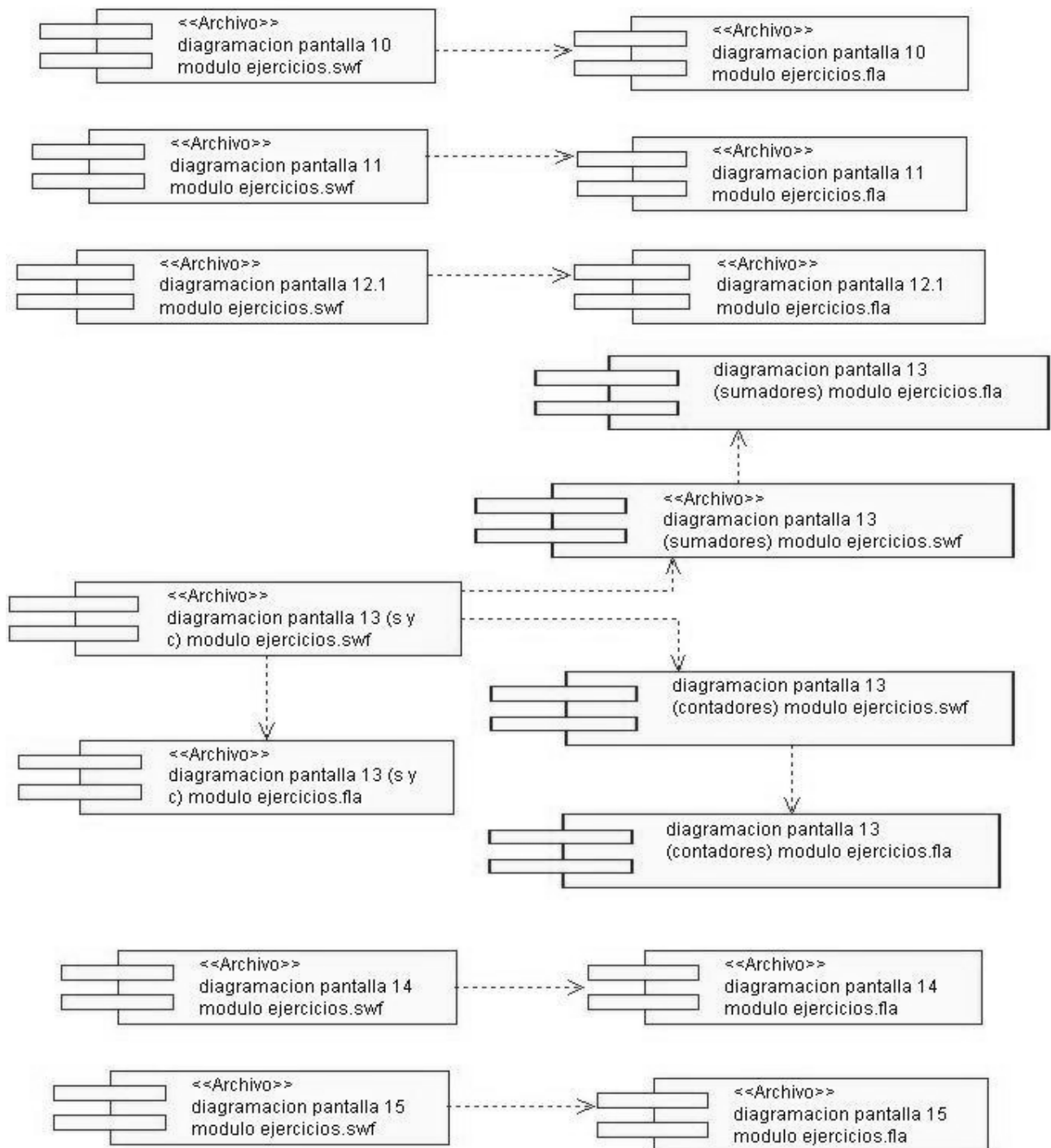


Ilustración 28. Diagrama de componentes del paquete "Ejercicios"

3.4 Modelo de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. (15)

Un Diagrama de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Esto muestra la configuración de los elementos de hardware (nodos) y muestra cómo los elementos y artefactos del software se trazan en esos nodos. Los diagramas de despliegue representan los tipos de nodos del sistema y sus relaciones.

El diagrama de despliegue para el producto “Pizarrón Virtual” corresponde con una Terminal que no necesita estar en red, solo una unidad lectura de CD y además, debe contar con los dispositivos señalados en los requerimientos no funcionales de hardware.

(Ilustración 29)

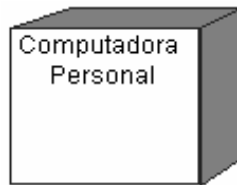


Ilustración 29. Diagrama de despliegue

3.5 Conclusiones

Al concluir este capítulo en el que se utilizaron los diagramas propuestos por OMMMA_L para la fase de diseño: quedaron listos los siete diagramas de presentación que guiaran la modelación de las interfaces del software más el diagrama de clases del modelo de objetos en el cual se presentan las medias relacionadas con cada uno de los escenarios y aplicaciones que posee la herramienta.

Por otro lado, se diseñaron todos los componentes necesarios para la implementación y quedó estructurada la arquitectura en tiempo de ejecución del sistema a través del “diagrama de despliegue” donde se definió que para la utilización del producto por un usuario es necesaria una sola Terminal, la cual entre sus características fundamentales debe tener una unidad de lectura de CD.

CAPÍTULO 4

ESTUDIO DE FACTIBILIDAD

4.1 Introducción

Antes de decidir hacer un proyecto, es necesario analizar si es viable o no desarrollar el mismo. Para esto se realiza un proceso que permita definir su factibilidad en el cual es preciso tener en cuenta algunos aspectos tales como: el esfuerzo humano, el tiempo que se requiere para la ejecución del mismo y su costo.

El presente capítulo muestra el estudio realizado para determinar si es factible o no desarrollar la herramienta que se propone como solución al problema planteado. En el mismo se estimó el esfuerzo mediante casos de usos a través del método "Análisis de Puntos de Casos de Uso"; se realizó una estimación del tiempo de desarrollo para el proyecto y se calculan los costos que esta producción puede contraer.

Además, en el capítulo se muestran los beneficios que el proyecto aportaría de realizarse, y un análisis de estos contra sus costos.

4.2 Planificación

Para analizar si es factible o no hacer este software educativo se realizó la estimación del esfuerzo basado en casos de uso a partir del "Análisis de Puntos de Casos de uso." Este método se utiliza para estimar el tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores. A continuación se expone el mismo a través de sus 4 pasos para hallar el esfuerzo total que se requiere para elaborar el producto.

4.2.1 Estimación del esfuerzo

Paso 1. Identificar los Puntos de casos de uso Desajustados

Tabla 11 Cálculo de Puntos de Casos de Uso sin Ajustar

Variable	Descripción	Fórmula	Valor
UAW	Factor de Peso de los Actores sin ajustar	Ver : Tabla 12	$UAW = 3$
UUCW	Factor de Peso de los Casos de Uso sin ajustar	Ver: Tabla 13	$UUCW = 45$
UUCP	Puntos de Casos de Uso sin ajustar	$uucp = uaw + uucw$	$UUCP = 48$

- Para calcular UAW

Tabla 12 Cálculo del Factor de Peso de los Actores sin ajustar.

Tipo de Actor	Descripción	Factor	Número de actores	Resultado
Simple	Interfaz del sistema	1	0	0
Promedio	La interfaz interactiva o protocolar	2	0	0
Compleja	Interfaz gráfica	3	1	3
Total				3

- Para calcular UUCW

Tabla 13 Cálculo del Factor de Peso de los Casos de Uso sin ajustar.

Tipo de CU	Descripción	Factor	Número de CU	Resultado
Simple	1 - 3 transacciones	5	5	25
Promedio	4 - 7 transacciones	10	2	20
Complejo	8 -... transacciones	15	0	0
Total				45

Paso 2. Ajustar los Puntos de casos de uso

Tabla 14 Obtención de los Puntos de Casos de Uso Ajustados

Variable	Descripción	Fórmula	Valor
UUCP	Puntos de Casos de Uso sin ajustar	Ver : Tabla 11	$UUCP = 48$
TCF	Factor de complejidad técnica	Ver: Tabla 15	$TCF = 0.92$
EF	Factor de ambiente	Ver: Tabla 16	$EF = 0.785$
UCP	Puntos de Casos de Uso ajustados	$UCP = UUCP \times TCF \times EF$	$UCP = 34.67$

- Para Calcular TCF (Ver anexo 3 tabla 19)

Tabla 15 Factores de complejidad técnica (TCF)

$TCF = 0,6 + 0,01 \times \sum \text{Peso}_i \times \text{Valor}_j$ (Donde Valor es un número del 0 al 5)					
Factor	Descripción	Peso	Valor	$\sum \text{Peso}_i \times \text{Valor}_j$	Comentario
T1	Sistema Distribuido	2	0	0	No es un sistema distribuido
T2	Tiempo de la contestación y actuación de los objetivos	1	3	3	Es necesario que el tiempo de contestación y actuación de los objetivos

					correspondan con lo deseado
T3	Eficacia para el usuario final	1	5	5	Es necesario que sea eficiente
T4	Proceso interno complejo	1	1	1	Ningún cálculo
T5	Reusabilidad del código	1	0	0	No
T6	Fácil de instalar	0.5	5	2.5	Debe ser muy fácil de instalar
T7	Fácil de usar	0.5	5	2.5	Debe tener una interfaz amigable
T8	Portabilidad	2	5	10	Compatible con diferentes Sistemas Operativos
T9	Fácil de cambiar	1	4	4	Costo de mantenimiento bajo
T10	Consistente	1	3	3	Necesita algo de consistencia
T11	Incluye objetivos de seguridad especiales	1	0	0	No son necesarios
T12	Mantiene el acceso directo a terceras partes	1	0	0	No
T13	Se requiere facilidades de entrenamiento para usuarios especiales	1	1	1	Sistema de fácil uso
Total (Tfactor):				32	
Luego:					

$$TCF = 0.6 + (0.01 \times T_{factor})$$

$$TCF = 0.6 + (0.01 \times 32)$$

$$TCF = 0.92$$

- Para Calcular EF (Ver anexo 4 tabla 20)

Tabla 16 Factores Ambiente

$EF = 1.4 - 0.03 * \sum \text{Peso}_i \times \text{Valor}_j \text{ (Donde Valor es un número del 0 al 5)}$					
Factor	Descripción	Peso	Valor	Factor	Comentario
E1	Familiaridad con el proyecto que se ejecuta	1.5	1	1.5	Poca familiarización con el proyecto
E2	Experiencia en la aplicación	0.5	1	0.5	Poco tiempo de trabajo con la aplicación
E3	Experiencia en la programación orientada a objetos	1	4	4	El personal posee bastante experiencia en la Programación Orientada a Objetos
E4	Capacidad del analista	0.5	5	2.5	El analista está altamente capacitado
E5	Motivación	1	5	5	Existe una gran motivación por parte del equipo
E6	Requerimientos estables	2	4	8	Los requerimientos no deben cambiar
E7	Personal de media jornada	-1	0	0	Los miembros del equipo trabajan a

					tiempo completo
E8	Grado de dificultad del lenguaje de programación	-1	1	-1	ActionScript
Total (Efactor):				20.5	
Luego:					
$EF = 1.4 + (-0.03 \times Efactor)$ $EF = 1.4 + (-0.03 \times 20.5)$ $EF = 0.785$					

Paso 3. Calcular esfuerzo de FT Implementación

$$E = UCP * CF$$

Donde:

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

- Para calcular CF

Para calcular el factor de conversión necesario para hallar el esfuerzo se utilizó como guía el siguiente criterio:

1. Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.
2. Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- ✓ Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- ✓ Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- ✓ Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

Resumiendo:

CF = 20 horas-hombre (si $Total_{EF} \leq 2$)

CF = 28 horas-hombre (si $Total_{EF} = 3$ ó $Total_{EF} = 4$)

CF = abandonar o cambiar proyecto (si $Total_{EF} \geq 5$)

En el presente caso tres de los factores ambientes que se encuentran de E1-E6 están por debajo del valor medio y ninguno de los que se encuentran entre E7-E8 está por encima del valor medio (3). Es por esto que:

$$Total_{EF} = 3 + 0$$

$$Total_{EF} = 3$$

$$CF = 28 \text{ horas-hombre (porque } Total_{EF} = 3)$$

Luego: $E = UCP * CF$
 $= 34.666 * 28$
 $E = \underline{970.636 \text{ horas-hombres}}$

Paso 4. Calcular esfuerzo de todo el proyecto

Tabla 17 Esfuerzo por flujo de trabajo

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	242.66 horas-hombre
Diseño	20%	485.318 horas-hombre
Implementación	40%	970.636 horas-hombre
Prueba	15%	363.989 horas-hombre
Sobrecarga	15%	363.989 horas-hombre
Total	100%	2426.589 horas-hombre

Como el valor de esfuerzo calculado representa el esfuerzo de la actividad implementación, por comparación salen el resto de los esfuerzos y la suma de ellos es el **esfuerzo total (E_T)**.

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 20 días laborables; la cantidad de horas que puede trabajar una persona en 1 mes es 160 horas por lo tanto,

$$E_T = 15.16 \text{ meses-hombres}$$

4.2.2 Tiempo total estimado para el proyecto y costo

El esfuerzo es la cantidad de horas-hombre que se tienen que emplear para la realización del software. Si esta variable se divide entre 5 que serían las personas vinculadas al proyecto (4 informáticos y un diseñador), el tiempo estimado para el proyecto sería:

CH: Cantidad de Hombres

TE: Tiempo Estimado

E_T : Esfuerzo Total

$$TE = E_T / CH$$

$$= 15.16 \text{ hombres-mes} / 5 \text{ hombres}$$

$$= \underline{\underline{3.032 \text{ meses}}}$$

Para calcular el costo del proyecto, se asume como salario \$225 que es el salario básico que cobra un adiestrado. Por lo tanto, pagándole \$ 225 a cada uno de los 5 desarrolladores de producto, durante los tres meses que se estima que dure la elaboración del software eso daría un costo por concepto de salario de:

$$\$ 225 * 4 = \$ 1125$$

$$\$ 1125 * 3 = \underline{\underline{\$3375}} \quad \text{Este costo convertido equivale a } \underline{\underline{\$135 \text{ CUC}}} \text{ y a } \underline{\underline{\$141.5 \text{ USD}}}$$

A parte de esto, sería necesario, algunos dispositivos para la elaboración del software. Entre ellos se encuentran:

Dispositivo	Cantidad	Justificación	Costo MN	Costo CUC	Costo USD
PC	3	Para que trabajen los 5 desarrolladores	33187	1287.5	1350
Bocinas	3	Necesarias para trabajar con las medias	865	34.6	33
Impresora	1	Para la impresión de materiales de estudio y análisis necesarios para los desarrolladores.	4649.32	185.97	195
Hojas	2 cajas	Para soportar información necesaria para el proyecto	939.90	37.60	39.1
CDRW	10	Para que soporte la herramienta a la hora de entregarla a los negociadores	375	15	15.7
Costo total de útiles necesarios:			40016.22	1560.67	1632.8

En total, sumando el costo por concepto de salario más el costo de los dispositivos necesarios, el costo total del proyecto equivaldría a:

$$CT = CS + CD$$

$$= \$ 3375 + \$ 40016.22$$

$$\boxed{= \$ 43391.22 \text{ MN}}$$

Donde:

Ct: costo total del proyecto

Cs: Costo por concepto de salario

Cd: Costo total de los dispositivos necesarios

Esto equivale a **\$1735.65 CUC** y a **\$1805.07 USD**

4.3 Beneficios tangibles e intangibles

4.3.1 Tangibles

Como principales beneficios **tangibles** asociados al desarrollo del producto “Pizarrón Virtual” se señalan los siguientes:

- Se contará con una aplicación flexible, dinámica y de interfaz agradable que posibilite a los estudiantes perfeccionar su aprendizaje.
- La escuela “Luís Razetti” contará con un medio tecnológico que integrará la educación con las TICs.
- Su comercialización aportará una bonificación monetaria de 10 mil USD.

4.3.2 Intangibles

Como principales beneficios **intangibles** asociados al desarrollo del producto “Pizarrón Virtual” se señalan los siguientes:

- Agilizar el aprendizaje, de manera que los estudiantes puedan resolver todas sus dudas y preguntas sin necesidad de la asistencia en todo momento del profesor de la asignatura, evitando las interrupciones y el retraso generado por esta situación en las horas de clase.
- Mejorar el nivel de rendimiento.
- Incentivar a la comprensión lectora y por ende el análisis de los enunciados de los problemas en el objetivo de técnicas de resolución de problemas algorítmicos.

4.4 Análisis de costos y beneficios

Según los cálculos realizados anteriormente, la elaboración del software que se propone como solución al problema planteado al inicio de este trabajo, se valora en un costo de \$1805.07 USD. Esta suma no llega a los 10 000 dólares que aportaría la venta de este producto. Sin considerar además, el hecho de que 4 de los 5 desarrolladores dispuestos para la elaboración de “Pizarrón Virtual” son estudiantes y no trabajadores, por lo que no cobran un salario mínimo por realizar esta tarea sino un estipendio que a lo sumo es de \$ 100 MN. Por otro lado, a la comisión monetaria se le agregan otra suma de beneficios tangibles y no tangibles antes mencionados que hacen de este software una magnífica oportunidad de desarrollo.

4.5 Conclusiones

El desarrollo del producto “Pizarrón Virtual” como respuesta al problema planteado es un proyecto completamente factible. Los cálculos y estimaciones realizados en este capítulo evidencian esta afirmación con los siguientes resultados:

Tabla 18 Resultados estimados

Parámetros	Valores
Esfuerzo	15.16 hombres/mes
Tiempo de desarrollo	3.032 meses
Cantidad de hombres	5 hombres
Salario	\$ 225
Costo total en MN	\$ 43391.22
Costo total en CUC	\$1735.65
Costo total en USD	\$1805.07

Igualmente los beneficios que alcanza son óptimos despuntando por encima de todo el aporte económico de 10 mil dólares que cubre el costo total del proyecto y mucho más. Además, la suma de todos los beneficios que aportará “Pizarrón Virtual” pesa mucho más que el costo que contrae esta producción.

CONCLUSIONES

- Tras la investigación se concluye que la forma que tienen los jóvenes hoy de crear sus propios conocimientos es distinta a que existía años atrás. El medio televisivo, los videojuegos, Chat e Internet hacen que sus intereses sean otros a la hora de recibir la información. Esto implica que sea conveniente la introducción de las TICs en la escuela “Luis Razetti” para mejorar el proceso de enseñanza y aprendizaje en dicho centro estudiantil.
- La selección de RUP como metodología a seguir para desarrollar dicho producto, acompañado de OMMMA_L como extensión de UML para el modelado permitieron que se diseñasen artefactos que consolidan la comunicación mediante un único lenguaje entre todas las personas implicadas en el proyecto.
- La propuesta que aquí se diseñó, facilitará la implementación del producto, con una interfaz amigable y que propicia un ambiente familiar para la audiencia a la que va dirigido.
- Pizarrón Virtual constituye la respuesta ideal al problema que existe en la escuela “Luis Razetti” en la asignatura “Algorítmica y Programación”. El mismo es un proyecto completamente factible que aportaría un beneficio económico de 10 mil usd cubriendo ampliamente los costos que lleva realizarlo en un tiempo de desarrollo de 3.032 meses por cinco personas.
- Con el presente trabajo se logró obtener la documentación necesaria y precisa para implementar el software “Pizarrón Virtual”, a través de su análisis y diseño quedando así completamente materializado el objetivo general de esta investigación.

RECOMENDACIONES

- Desarrollar la implementación del software “Pizarrón Virtual” que propone el presente trabajo como propuesta para resolver los problemas existentes en la escuela “Luis Razetti” de Caracas, Venezuela.
- Valorar la opción de reducir el número personas para la realización del producto pues el grado de sencillez del mismo no requiere de 5 desarrolladores.

REFERENCIAS BIBLIOGRÁFICAS

1. BENITO, J. J. E. *Sistema multimedia destinado al aprendizaje interdisciplinar de curvas cicloides y fractales*, [Citado el: 28/01/07] Disponible en:
http://www.matematicalia.net/index.php?option=com_content&task=view&id=203&Itemid=149
2. CONDORÍ, C. *Impacto de las nuevas tecnologías en educación*, 2004 [Citado el: 25/01/07]
Disponible en: http://www.ilhn.com/periodismo/archives/cat_trabajo_final.php.
3. SQUIRES y MCDUGALL. *Cómo elegir y utilizar software educativo*. Ediciones Morata, España 2001.
4. OECD. *Are Students Ready for a Technology-Rich World?: What PISA Studies Tell Us.*, 2006 [Citado el: 05/02/07] Disponible en:
http://www.oecd.org/document/14/0,2340,en_2649_201185_36002382_1_1_1_1,00.html
5. LORA, C. *Nuevas herramientas para la enseñanza*, 2005 [Consultado el: 05/02/07] Disponible en:
<http://www.educando.edu.do/Educando/Administracion/Recursos/Articulos/Nuevas+herramientas+para+la+ense%C3%B1anza.htm>.
6. CUEVAS, R. D. R. *¿Qué necesitamos para integrar las TIC a nuestros salones de clases?*, 2006 [Citado el: 05/02/07] Disponible en:
<http://www.educando.edu.do/Educando/Administracion/Recursos/Articulos/Que+necesitamos+para+integrar+las+TIC+a+las+aulas.htm>.

7. TORRES, D. M. A. *Software Educativo*, 2003 [Citado el: 10/02/07] Disponible en:
<http://www.feyalegria.org>.

8. ANDRADA, L. A. M. y ROSSI, D. G. *El desarrollo de software educativo hoy: perspectivas epistemológica, conceptual y tecnológica*, 1997 [Citado el: 27/01/07] Disponible en:
<http://www.horizonteweb.com/jie98/conferen.htm>

9. GROS, D. B. S. *Del software educativo a educar con software*, 2000 [Citado el: 10/02/07]
Disponible en:

<http://www.metacrawler.com/info.metac/search/web/concepto%252Bde%252Bsoftware%252Beducativo>

10. PRESSMAN, R. S. *Ingeniería del Software: Un enfoque práctico*. 5ta ed. LA Habana: Félix Varela, 2005. vol. I, 370 p.

11. JIMÉNEZ VALDÉS, S.. *Propuesta del proceso de producción para el departamento de multimedia educativa de la Universidad de Ciencias Informáticas*. Universidad de las ciencias Informáticas, Ciudad Habana, 2005.

13. MARTÍNEZ, J. M. S. and HILERA, J. R. G. *Modelado de documentación multimedia e hipermedia*, 1997 [Citado el: 15/03/07] Disponible en: <http://www.ucm.es>

14. MARTÍNEZ, M. and ROSSI, G. C. *Estudio de la composición y estructura poblacional de culícidos (Díptera, Culicidae) en zonas representativas de Montevideo*. Facultad de Ciencias. PEDECIBA, 1996 [Citado el: 12/02/07] Disponible en:
<http://www.ugf.br/editora/revistas/entomologia/eyv2003/art6.pdf>

14. NAVARRETE, A. *Una metodología relacional hipermedia. Estudio en casos prácticos*. Tutor: Gimeno., D. J. B. Investigativa, Ingeniería Superior en Informática. 1998.[Citado el: 17/02/07]
Disponible en: http://gti.upf.edu/english/publications/publicacion_english.2006-10-10.4263540830

15. JACOBSON, I.; BOOCH, G., *et al. El proceso unificado de desarrollo de software*. 2004.

16. SCHMULLER, J. *Aprendiendo UML en 24 horas* [Consultado el: 04/03/07] Disponible en: <http://bibliodoc.uci.cu/pdf/reg00004.pdf>.

17. LARMAN, C. *UML y Patrones*.

18. BERROCOSO, J. V. *Herramientas de autor* [Consultado el: 07/03/07] Disponible en: http://www.unex.es/didactica/Tecnologia_Educativa/guion01.htm.

19. WOODS, P. S. *Programación de Macromedia Flash* Editado por: McGraw-Hill, E. 2003 [Consultado el: 07/03/07] Disponible en: www.agapea.com/Programacion-de-Macromedia-Flash-Mx-n8750i.htm - 18k.

20. ANTONI MARIN, C. A. *Animación*. 2005, [Consultado el: 26/04/07] Disponible en: <http://mosaic.uoc.edu/entrevistas/ndemo1105.html>.

22. FRASCARA, J. *Definiendo audiencia* [Consultado el: 27/01/07] Disponible en: http://www.wolkoweb.com.ar/apuntes/textos/definiendo_audiencia.rtf.

23. JACOBSON, I.; BOOCH, G., *et al. El proceso unificado de desarrollo de software*. 2004, vol. 1.

24. PERALTA, M. *Estimación del esfuerzo basada en casos de uso* Disponible en: <http://www.itba.edu.ar/capis/rtis/rtis-6-1/estimacion-del-esfuerzo-basada-en-casos-de-usos.pdf>.

REFERENCIAS BIBLIOGRÁFICAS

26. GUREVICZ, M. and TORO, C. *EL CONSTRUCTIVISMO. Jean Piaget*. 2000 [Citado el: 23/03/07]

Disponible en: <http://www.uba.ar/academicos/uba21/download/materias/psi-act-piaget1.pdf>

BIBLIOGRAFÍA

1. SCHMULLER, Joseph. *Aprendiendo UML en 24 horas* [en línea] PEARSON EDUCACION, México, 2000. Disponible en: <http://bibliodoc.uci.cu/pdf/reg00004.pdf>
2. PERALTA, M.. *Estimación del esfuerzo basada en casos de uso. Centro de Ingeniería del Software e Ingeniería del Conocimiento (CAPIS)*. Instituto Tecnológico de Buenos Aires, 2003 Disponible en: <http://www.itba.edu.ar/capis/webcapis/planma.html>
3. JIMÉNEZ VALDÉS, S.. *Propuesta del proceso de producción para el departamento de multimedia educativa de la Universidad de Ciencias Informáticas*. Universidad de las ciencias Informáticas, Ciudad Habana, 2005.
4. DÍAZ CATALÁ, S. and SOLENZAL FERNÁNDEZ, G.. *Multimedia Auto-Aprende*. Universidad de las ciencias Informáticas, Ciudad Habana, 2006.
5. MARTÍNEZ PÉREZ, Y., DÍAZ DOMÍNGUEZ, A.. *Plantilla para el Montaje Dinámico de los Productos de la Colección Multisaber*. Universidad de las ciencias Informáticas, Ciudad Habana, 2006.
6. JACOBSON, Ivar., BOOCH, Grady, and RUMBAUGH, James. *El Proceso Unificado de Desarrollo de Software*. Series Editors Addison-Wesley
7. COELLO GONZÁLEZ, S. and HERNÁNDEZ LEÓN R, A. *El paradigma cuantitativo de la investigación científica*. Universidad de las Ciencias Informáticas, Ciudad de La Habana, 2002.
8. CIUDAD RICARDO, F, A. and MONROSE, S.. *EMBRIOCIM – Enciclopedia de Embriología Médica – Colección GALENOMEDIA*, Instituto Superior Politécnico “José Antonio Echevarría”, Ciudad de La Habana, 2004.

9. JACOBSON, I.; BOOCH, G., et al. *El proceso unificado de desarrollo de software*. 2004, vol. 1
10. JACOBSON, I.; BOOCH, G., et al. *El proceso unificado de desarrollo de software*. 2004, vol. 2
11. SÁNCHEZ ILABACA, J. *Aprendizaje visible, computador invisible*. Taller Internacional de Software Educativo, 1999. Disponible en : <http://www.c5.cl/tise99/memoriatise99/html/fcharlas.html>
12. VILLANUEVA ARMENTEROS, Y. *Tendencias actuales en la enseñanza- aprendizaje de las matemáticas y la utilización*, 2000 Disponible en:
<http://www.revistaciencias.com/publicaciones/EEpFFuVpZuQiCgbPvy.php>
13. FERNANDO LAGE, J. and CATALDI, Z. *Una experiencia de resolución de problemas a través de modelos cooperativos-colaborativos aplicada a algoritmia usando nuevas tecnologías de comunicación*, 2001 Disponible en: <http://www.fi.uba.ar/laboratorios/lie/Publicaciones/51.pdf>
14. AREA, M. *Los medios, los profesores y el currículo*, 1991 Disponible en:
<http://webpages.ull.es/users/manarea/sendai/CAPT3.pdf>
15. Esperanza Marcos Grupo KYBELE, *Investigación en Ingeniería del Software vs. Desarrollo Software*, Universidad Rey Juan Carlos, 2003 Disponible en:
<http://kybele.escet.urjc.es/MIFISIS2002/Articulos%5CArt11.pdf>
16. CABERO ALMENARA, J. *Líneas y tendencias de investigación en medios de enseñanza*, 2002 Disponible en: <http://tecnologiaedu.us.es/bibliovir/pdf/69.pdf>

Sitios consultados:

<http://www.rational.com.ar>

<http://www.upm.es>

<http://www.sparxsystems.com.ar>

GLOSARIO DE TÉRMINOS

Aplicaciones informáticas: Son programas de ordenador que se compran ya realizadas y listas para usar. Las hay de muy diversos tipos, según para qué propósito se hayan diseñado: procesadores de texto, bases de datos, programas de contabilidad, de facturación, etc.

Aplicaciones ofimáticas: programas informáticos propios para facilitar el trabajo de oficina.

Artefacto: es una pieza tangible que crean, modifican y usan las personas encargadas de realizar determinadas actividades durante el proceso de desarrollo del software.

Avance tecnológico: progreso que está teniendo la tecnología a medida que están pasando los años (o los meses).

Base de datos: es un almacén de datos relacionados con diferentes modos de organización. Representa algunos aspectos del mundo real. Diseña y almacena datos con un propósito específico.

Capacidades cognitivas: habilidades que tiene los sujetos para procesar información a partir de la percepción. Es decir primero se procesa información y después se analiza, se argumenta, y se comprende.

Chat: recurso de Internet que permite la comunicación entre personas en formato texto.

Ciberespacio: el mundo de las redes. Espacio donde navegan las personas por Internet.

Diagrama: es la representación gráfica de un conjunto de elementos, usualmente representado como un grafo conectado de vértices (elementos) y arcos (relaciones).

Entidades: son objetos concretos o abstractos que presentan interés para el sistema.

Hardware: componente físico tecnológico que trabaja o interactúa de algún modo con la computadora.

Herramientas de autor: son herramientas que se utilizan fundamentalmente para la elaboración de contenidos multimediales.

Internet: red informática de comunicación internacional que permite el intercambio de todo tipo de información entre sus usuarios.

Medios de enseñanza: herramientas que se utilizan para apoyar la enseñanza.

Mención informática: carrera que se cursa en el politécnico “Luis Razetti” de Caracas, Venezuela.

Ordenador: Es una máquina o un sistema de tratamiento de la información que realiza operaciones automáticas para las cuales ha sido previamente programada (Ejemplo: una computadora personal).

Palabras calientes: palabras con vínculos a otros textos

Programas tutoriales: programas que guían el aprendizaje

Requisitos: condición o capacidad que debe cumplir un sistema. También se les llama requerimientos.

Sistemas: conjunto de partes o elementos organizados y relacionados que interactúan entre sí para lograr un objetivo. Reciben (entrada) datos, energía o materia del ambiente y proveen (salida) información, energía o materia.

Sistemas de instrucción programada: son programas que desarrollan el auto-estímulo en el uso de los sistemas, la participación activa del estudiante y la realimentación durante el uso de los mismos.

Software: todo programa o aplicación para realizar tareas específicas.

Software de aplicación: programa informático que facilita la realización de un determinado tipo de trabajo. Suele resultar una solución informática para la automatización de ciertas tareas complicadas como puede ser la contabilidad o la gestión de un almacén.

Software educativo: programa con fines educativos.

TICs: tecnologías de la Información y las Comunicaciones. También aparece en los últimos tiempos el término NTICs (Nuevas Tecnologías de la Información y las Comunicaciones).