

UNA METODOLOGÍA ORIENTADA A OBJETOS PARA LA PRODUCCIÓN DE SOFTWARE MULTIMEDIA

AN OBJECT ORIENTED METHODOLOGY FOR MULTIMEDIA SOFTWARE DEVELOPMENT

GLADYS BENIGNI

Coordinación de Informática. Universidad de Oriente. Núcleo Nueva Esparta.

e-mail: gbenigni@ne.udo.edu.ve

RESUMEN

En las últimas décadas, con la aparición del computador, la memoria digital, la multimedia e Internet se ha podido observar un avance no solo continuo sino vertiginoso de las denominadas nuevas tecnologías de la información. El objetivo de esta investigación es presentar una metodología que permita generar de una manera práctica y sencilla, la creación de software hipermedial. Se destaca además, la importancia de la interfaz, a través del buen uso de metáforas para lograr lo que se desea al desarrollar software, esto es una fácil interacción hombre – máquina. Se revisaron diferentes metodologías de Ingeniería del software orientadas a objeto y otras relativas al desarrollo de software multimedia.

PALABRAS CLAVES: Software educativo, hipermedia, interfaz, grafo de navegación, orientación a objeto, Ingeniería de Software.

ABSTRACT

In the last decades, with the introduction of computers, digital memory, multimedia and Internet, there has been an impressively increasing advance on the so called new technologies of information, being informatics their common denominator. The aim of the following investigation is to present a methodology allowing the creation, in a simple and practical way, of hypermedial software. The importance of the interface has also been stressed through the good use of metaphors in order to achieve the aim when developing software, an easy interaction between man and machine. A review of different software engineering methodologies has been done. Some object oriented and others, regarding the development of multimedia software.

KEY WORDS: Educational software, hypermedial, interface, graph of navigation, object oriented, Software Engineering.

INTRODUCCIÓN

Este trabajo se centra en la creación de una metodología para el desarrollo o producción de software multimedia, enfocada en la particularidad de que puede ser utilizado tanto en el ámbito educativo como informativo.

La metodología aquí propuesta ha sido denominada MOOM (Metodología Orientada a Objeto Multimedia), y pretende involucrar estos aspectos tan importantes en el mundo actual de la informática. Esta metodología surge por la experiencia vivida como asesora de Trabajos de Grado en el desarrollo de aplicaciones multimedia, orientadas tanto al ámbito educativo como a propósitos generales, evidenciándose, durante el desarrollo de estos sistemas, la necesidad de contar con un método de desarrollo que permitiera la flexibilidad de abarcar el desarrollo de aplicaciones educativas - informativas y que

mantuviese la filosofía objeto orientado. Entre las metodologías más utilizadas en la Universidad de Oriente, Núcleo Nueva Esparta orientadas hacia la producción de software multimedia, se destacan: Modelo de Referencia Hipermedia propuesta por Montilva (1996); Método para la Producción de Software Educativo (PROSDOS) de Zambrano (1992); Metodología de Producción de Software (Propuesta) referida por el CENAMEC (1995) y en la actualidad denominada Materiales Educativos Computarizados (MEC); MECs propuesta por Galvis, entre otras.

Con MOOM se pretende disponer de una metodología para la producción de software educativo – informativo que permita a los desarrolladores la facilidad de uso, así como minimizar, por ejemplo, la cantidad de diseño de guiones cuando su estructura es la misma, pero el contenido es diferente (en función de texto, imágenes, entre otros). Además, hacer que la misma pueda ser

utilizada para el desarrollo en el Web, complementándola con las extensiones de la notación UML de Rational Rose para diseño de aplicaciones Web, propuesta por Conallen (1999).

Se han revisado diferentes metodologías de Ingeniería del Software Orientada a Objetos, Booch (1996) y Jacobson (1998) entre otros, con el fin de tomar de éstas los aspectos más importantes, que contribuyan a la creación de una metodología actual y de fácil uso para las personas interesadas en el desarrollo de software hipermedial.

MÉTODOS

Se examinaron las metodologías orientadas a objeto OOSE (Object-Oriented Software Engineering) propuesta por Jacobson (1998), y el Análisis y Diseño Orientado a Objetos de Booch (1996). En el área de desarrollo de software hipermedial, se consideraron dos metodologías; la primera, establecida en el texto “Introducción a los Sistemas Multimedia”, propuesto por Montilva (1996), y la segunda, denominada Enseñanza Asistida por Computador y Producción de Software Educativo (PROSDOS), de Zambrano (1995).

METODOLOGÍA MOOM

La presente metodología está subdividida en tres etapas o modelos: modelo de análisis, modelo de diseño y modelo de implantación.

Modelo de Análisis

La base central de MOOM radica en el modelo de análisis, ya que debe diferenciarse, estrictamente, hacia quien va dirigido el software que se pretende desarrollar; en estos momentos hablamos de autoría del software: Educativo o Informativo.

Cuando el software es netamente educativo, refiriéndonos particularmente a un(os) objetivo(s) específico(s) del contenido programático de un curso en particular, se debe cumplir con una serie de pasos involucrados en el proceso de enseñanza-aprendizaje. Es en este momento cuando nos involucramos en la enseñanza asistida por el computador (E.A.C).

La enseñanza asistida por el computador, hace referencia a la interacción humano-máquina o en nuestro caso en particular alumno(a)-máquina. Según Zambrano (1995), la E.A.C comprende todas las aplicaciones que utilizan al computador en el ámbito educativo, a saber: enseñanza, gestión y

administración escolar, evaluación y corrección de pruebas, orientación escolar e investigación pedagógica, entre otras.

Si hablamos de software multimedia informativo, podríamos hacer referencia a un libro electrónico, un atlas, diccionarios, folletos, revistas, entre otros. Con un software de este tipo, el usuario podría educarse obteniendo información particular del contenido del software que utilice, con la salvedad de que éste no cumple con el contenido programático de un curso específico. De este forma, el enfoque de análisis no debe orientarse de manera semejante al de un software netamente educativo.

Hasta los momentos, hemos querido establecer diferencias concretas, en función de quién va a ser el actor o participante en el uso del software. En el caso de la enseñanza asistida por computador, el actor principal puede ser el aprendiz o el docente (dependiendo de la aplicación). En lo que respecta al informativo, el actor principal puede ser cualquier usuario interesado en el tema.

PASOS DEL MODELO DE ANÁLISIS.

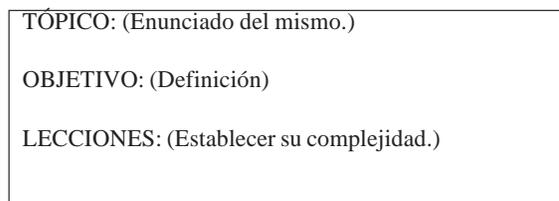
El modelo de análisis está compuesto de las fases siguientes:

- Determinación de los requerimientos del problema.
- Identificación de los objetos.
- Elaboración del mapa de navegación del sistema.

Determinación de los requerimientos del problema.

En esta etapa, el desarrollador del software, se encargará de determinar las necesidades del sistema y su factibilidad. La factibilidad, viene dada por la disponibilidad real en cuanto a los recursos necesarios para el desarrollo del prototipo.

Una vez determinada las necesidades y la factibilidad del mismo, se procede a distinguir hacia quién va dirigido el proyecto. Al tratarse de un *Soporte Didáctico*, deben definirse las lecciones correspondientes al tópico seleccionado. Este tópico debe escogerse con personal especializado en el área (en este caso, maestros(as) o profesores(as), analizando aquellos que demuestren mayor complejidad e importancia (esquema 1) en el proceso enseñanza-aprendizaje y cumplir estrictamente con los objetivos aprobados, por el Ministerio de Educación.



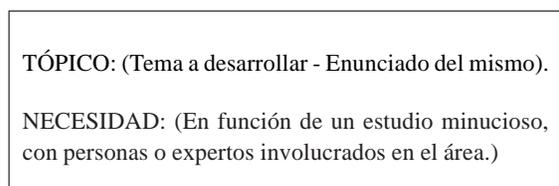
Esquema 1. Complejidad de las lecciones de un Tópico seleccionado.

Una pregunta común es referida a la complejidad. ¿Qué tan compleja es una lección en función de otra?. La respuesta viene dada a través de la importancia que tengan los temas que serán abordados en cada lección en cuanto a la profundidad de ampliación de los mismos, permitiendo a los actores poder navegar más allá en la búsqueda de nueva información.

Por lo tanto, para lograr el éxito se debe tomar en consideración que el aprendizaje por parte de los actores sea significativo. En primer lugar, el contenido debe ser potencialmente significativo, tanto desde el punto de vista de su estructura interna (no debe ser arbitrario, ni confuso), como desde el punto de vista de su asimilación. En segundo lugar, debe tenerse una actitud favorable para aprender significativamente, es decir, el alumno ha de estar motivado por relacionar lo que aprende con lo que sabe, además apoyado en las motivaciones y conocimientos previos del aprendiz.

Con la ayuda del profesor o maestro, se podrá determinar qué tópico en específico debe seleccionarse para su desarrollo. Una vez determinado el mismo en función de la complejidad, se procede a elaborar un boceto en función de las lecciones que formarán parte de este, indicando la complejidad establecida en cada una de ellas.

Si el prototipo a desarrollar es un *software multimedia informativo*, el mismo se analizará en función de sus necesidades, estableciéndose así hacia quién va dirigido y el objetivo del mismo (Esquema 2). Para este tipo de sistemas, no hablaremos de lecciones, sino de unidades de información, que de alguna forma tienen correspondencia hacia un tema en particular.



Esquema 2. Estudio de la necesidad del software.

El definir lecciones o unidades de información, según sea el caso, y una vez determinada la factibilidad del desarrollo del sistema y por último, definidos los objetivos, se procede con la etapa de determinar los objetos presentes en el prototipo de sistema a desarrollar.

Definición de los objetos.

Hasta los momentos, se han determinado las lecciones tratándose de un soporte didáctico o las unidades de información refiriéndonos a un software informativo, correspondientes al prototipo de software multimedia que se desea diseñar. En esta etapa, se definirán los objetos en función de las mismas y la relación existente entre ellas. Es importante señalar, que todavía nos encontramos a un nivel macro del problema, por lo tanto, los objetos serán también definidos al mismo nivel.

La relación existente entre los objetos viene dada a través de las asociaciones entre los mismos. Estas asociaciones representan en los sistemas multimedia los links, hipervínculos o ramificaciones hacia otras lecciones o unidades de información. Con esto no se pretende que los enlaces estén sólo en función de un primer nivel de asociación, sino por el contrario, a medida que se avance en el diseño aparecerán nuevos vínculos o ramificaciones, que en muchos casos pueden representar hipertextos.

Un ejemplo, que puede aclarar lo anteriormente expuesto, sería el considerar un tópico de una asignatura en particular a la que denominaremos T. De este tópico pueden surgir a un nivel general n lecciones, las que llamaremos L_i , con $i = 1...n$. A este punto tendríamos definido los objetos principales del sistema y su representación puede observarse en la Figura 1.

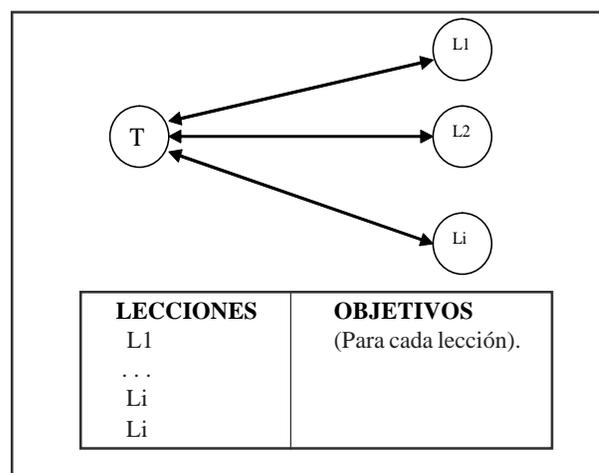


Figura 1. Representación de los objetos principales del sistema

Una vez representado los objetos y sus asociaciones, es importante definir los objetivos del tópico seleccionado, al igual que cada una de las lecciones que se desprenden del análisis detallado y minucioso del mismo. Cuando hacemos referencia a un software educativo, es imprescindible, definir los objetivos específicos de cada lección, para llevar a feliz término el tópico seleccionado o propuesto. Si se trata de un software multimedia informativo (Esquema 3), debe indicarse el objetivo del mismo hacia el tema o tópico seleccionado, así como también definir el propósito de cada unidad de información.

UNIDADES DE INFORMACIÓN	OBJETIVOS (En función del propósito de cada Ui).
U1 U2 ... Ui	

Esquema 3. Forma para indicar los objetivos de las unidades de información

Elaboración del Grafo de Navegación del Sistema.

Es imposible comenzar a diseñar un sistema multimedia, que no haya sido evaluado en la fase de análisis por personal especializado en el área. Para esto, se introduce la noción de elaborar un grafo de navegación, donde estarán todas las lecciones o unidades de información, según sea el caso. Este grafo representará el prototipo del sistema a desarrollar, a través de nodos (objetos) y las asociaciones o enlaces se indicarán según lo asocie el desarrollador del software. Es importante señalar que en este nivel no identificamos qué tipo de elementos multimedia se usarán, pero sí se señalará (de qué tratará cada uno de los objetos indicados) el boceto general del mismo, para obtener la aprobación del usuario o aceptar las sugerencias del mismo. Con este grafo de navegación se sabrá a qué nivel de profundidad serán los enlaces para cada lección o unidad de información. Recuerde que cada nodo representa una lección o unidad de información.

La terminología utilizada para el grafo de navegación se muestra en el Esquema 4, y en la Figura 2 se muestra un ejemplo de grafo de navegación.

LECCIÓN	LECCIONES ASOCIADAS	DESCRIPCIÓN
Ln		Descripción general de cada lección.

(a)

LECCIÓN	LECCIONES ASOCIADAS	DESCRIPCIÓN

(b)

Esquema 4. Terminología para elaborar el Grafo de Navegación: (a) software educativo; (b) software informativo.

EJEMPLO:

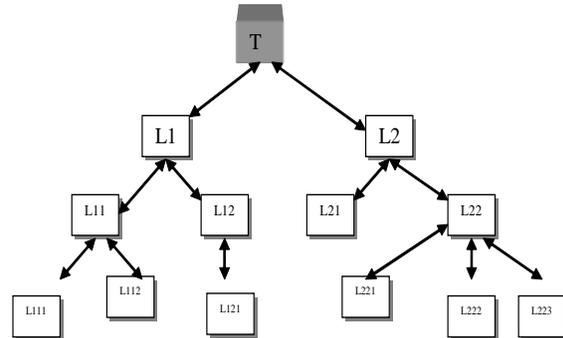


Figura 2. Grafo de Navegación para un software educativo

El ejemplo mostrado indica que para el tópico (T) seleccionado, se desprenden las respectivas lecciones que van a dar origen al sistema que se desea desarrollar. Cada nodo del grafo, representa un bloque de información, que posteriormente, en la fase de diseño, se mostrará detalladamente. Como se puede observar en la Figura 1, los enlaces en el cuadro son bidireccionales, lo que detalla la importancia de los hipervínculos en multimedia. Este ejemplo es bastante sencillo, pero a medida que crece la complejidad de los objetivos involucrados en el tópico aumenta el tamaño del grafo. Es importante destacar que cualquier nodo del grafo puede tener hipervínculo con cualquier nivel del grafo. Además, dada la flexibilidad de esta metodología, los enlaces no necesariamente tienen que ser todos bidireccionales, ya que dependerá directamente del sistema que se desee desarrollar y de sus características propias.

Ejemplificando con respecto a los hipervínculos, podríamos tomar de la Figura 3 el nodo correspondiente a la lección L112 y representar vinculación a otros nodos.

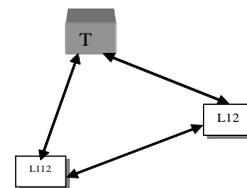


Figura 3. Hipervínculos.

En la figura 3 podemos observar que, a su vez, el nodo L112 mantiene relación bidireccional con T y el nodo L12.

Diseñar un grafo completo puede causar complejidad en el entendimiento del mismo por parte de los usuarios, puede dividirse el grafo inicial (figura 4) en varios para su simplicidad.

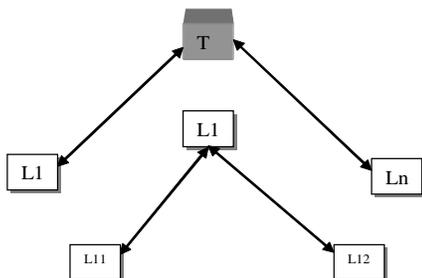


Figura 4. Descomposición del tópico en sus respectivas lecciones.

Esta metodología es abierta y la subdivisión del grafo de navegación, será considerada por el desarrollador del sistema.

MODELO DE DISEÑO.

Ya determinado en el modelo de análisis los objetos presentes en el sistema, y previamente validados por el personal encargado del mismo, se procede a diseñar cada uno en función de los elementos de multimedia que se usarán en el mismo (vídeo, texto, animación, imagen, sonido, entre otros).

Para esto, se tomarán los objetos (lecciones o unidades de información), y se detallarán en función de lo que se requiere en cuanto a los elementos multimedia a ser usados en el mismo, tomando en cuenta la terminología descrita a continuación.

Terminología a utilizar en el Diseño.

La terminología a utilizar en el diseño del sistema se muestra a continuación en la Tabla 1.

Tabla 1. Tabla de Nomenclatura de los elementos multimedia

Elementos Multimedia	Nomenclatura
Texto	T(i) i = 1,...,n
Hipertexto	Ht(i) i = 1,...,n
Animación	A(i) i = 1,...,n
Vídeo	V(i) i = 1,...,n
Botón(es)	B(i), i=1,...,n
Iconos o Iconos Animados	I(i) o IA(i)
Imagen	IM(i) i = 1,...,n
Sonido	S(i) i = 1,...,n
Fondo	F(i) i = 1,...,n
Narración	N(i) i = 1,...,n
Morfismo	M(i) i = 1,...,n

Dado que el diseño de un sistema multimedia, se torna pesado por repeticiones constantes de cada una de las lecciones o unidades de información, se recomienda que aquellas cuya estructura sea similar, indicarla solo una vez haciendo referencia de la unidad o lección correspondiente como veremos posteriormente.

Diseño de los objetos.

En el punto anterior, se describió la nomenclatura que ha sido utilizada por cada elemento de multimedia, en este caso mostraremos, de una manera sencilla, cómo representar las lecciones o cualquier tipo de información.

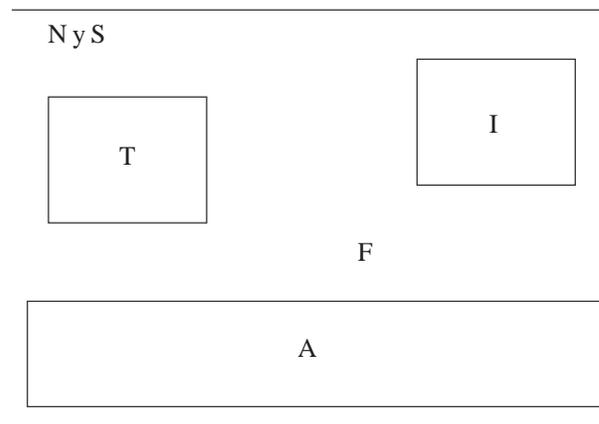
Para esto, se diseñarán tablas en las cuales se mostrará el boceto del sistema que se propone. La tabla 2 refleja la manera de representarlo:

Tabla 2. Elementos utilizados

Nombre de la Lección o Unidad de información:		
Objetivo:		
Medios Utilizados (Marque con X):		
TEXTO	<input type="checkbox"/>	VIDEO <input type="checkbox"/> HIPERTEXTO <input type="checkbox"/>
ANIMACIÓN	<input type="checkbox"/>	SONIDO <input type="checkbox"/> IMÁGENES <input type="checkbox"/>
BOTONES	<input type="checkbox"/>	ÍCONOS <input type="checkbox"/> MORFISMO <input type="checkbox"/>
NARRACIÓN	<input type="checkbox"/>	FONDO <input type="checkbox"/>
Descripción de la lección o unidad		

Además, debe incluirse el boceto, tal como se muestra en la Tabla 3.

Tabla 3. Boceto de la lección o unidad de información.



El diseñador en este momento, puede mostrar a los posibles usuarios potenciales del sistema un prototipo de las posibles pantallas del sistema, para validar hasta qué punto la interfaz, a través de las respectivas metáforas (globales, visuales, entre otras), representan lo que el usuario concibe de cada uno de ellas, permitiendo minimizar considerablemente la carga cognitiva de los mismos. Esto permitirá avanzar con éxito en la programación del sistema, ya que el usuario estará interactuando permanentemente con el mismo.

Generalmente, se dan casos en los cuales la estructura del boceto es la misma, conllevando al diseñador a repeticiones constantes en la estructura más no en el contenido de la misma. Por ejemplo, puede variar el contenido del texto, audio, imágenes, entre otros, pero su ubicación en el contexto es la misma (es lo recomendable). Para tal fin, puede diseñarse una tabla (ver Tabla 3), donde se indique que lecciones o unidades tienen el mismo diseño, pero es importante detallar el contenido. Para esto, tomemos como ejemplo la unidad U1, en la cual existen a su vez otras iguales en su estructura.

Tabla 4. Unidades de Información con un boceto similar.

UNIDAD DE INFORMACIÓN DESCRITA	UNIDADES DE INFORMACIÓN IGUAL BOCETO	VARIACIÓN
U1	U1.2	N, S, T
	U2	S, T, IA.

En la Tabla 4 puede observarse que las unidades U1.2 y U2, tienen la misma distribución en el boceto, pero varían en el caso de U1.2 en el sonido, contenido del texto y narración (audio).

Una vez realizado todos los bocetos, se procede a indicar el tipo de sonido y en forma general el texto, imágenes, entre otras presentes en cada uno de ellos, utilizando para esto lo que denominaremos bibliotecas. A continuación mostraremos el esquema para las mismas.

Biblioteca de Sonido.

Tal como se muestra en la Tabla 5, se indicarán todos los sonidos presentes en el sistema que se esté desarrollando con su respectivo tipo de archivo.

Tabla 5. Biblioteca de Sonido

NOMBRE	ARCHIVO
Ej.: Galerón Margariteño	Gal.mid

Biblioteca de Texto.

En la Tabla 6, podemos observar en que lección o unidad de información, se encuentra el texto descrito con su respectiva extensión.

Tabla 6. Biblioteca de texto

NOMBRE DEL TEXTO	ARCHIVO
EL UNIVERSO (L o Ui)	Univ.txt

Biblioteca de Imágenes.

Como puede verse en la Tabla 7, se especifican las imágenes con su respectiva extensión, así como en donde se encuentra ubicada en el sistema (L o Ui). En el caso de que desee crear bibliotecas para los demás elementos de multimedia, la metodología es flexible para esto y simplemente coloque el nombre a la biblioteca, su secuencia (en caso de animaciones) y su respectiva extensión.

Tabla 7. Biblioteca de Imágenes.

NOMBRE DE LA IMAGEN	ARCHIVO
Atardecer en Juangriego (L o Ui)	ATARD1.jpg

Modelo de Implantación.

En esta etapa, se procede a seleccionar los recursos computacionales necesarios para programar el sistema propuesto, posteriormente debe hacerse una evaluación exhaustiva con el personal que colaboró en el modelo de análisis y diseño, y por último, se elabora el manual de usuario respectivo.

En cuanto a los recursos computacionales, debe seleccionarse minuciosamente los requerimientos mínimos de hardware y especificar claramente él o los software ha ser utilizados y su utilidad.

Una vez programado el sistema, el mismo debe ser evaluado por especialistas no solo del área de Informática, sino de aquellas personas que participaron como expertos en el área de análisis, diseño y desarrollo para efectuar los cambios respectivos (si los hubiere). Si el desarrollador mantiene estrecha relación entre los expertos, potenciales usuarios de la aplicación, diseñadores gráficos, entre otros especialistas desde la fase inicial del proceso, no deben existir en consecuencia cambios importantes en esta última etapa.

Finalmente, debe desarrollar el respectivo manual de usuario, indispensable para cualquier persona que desee conocer e interactuar con el sistema.

CONCLUSIONES

Una vez finalizada la investigación, podríamos concluir que:

- La presente investigación permitió producir un nuevo método para desarrollar aplicaciones hipermediales destinadas al área educativo o informativa bajo el enfoque objeto orientado.
- MOOM, es considerada una herramienta sencilla en el desarrollo de aplicaciones educativas o informativas, ya que ésta ha sido y está siendo utilizada como método de desarrollo en la Universidad de Oriente – Núcleo Nueva Esparta y son estas las conclusiones emitidas por los diferentes desarrolladores de aplicaciones.

RECOMENDACIONES

Es recomendable ampliar el método tomando en consideración la arquitectura de modelado en el Web para ampliar el uso de la misma.

REFERENCIAS BIBLIOGRÁFICAS

- BOOCH, G. 1996. *Análisis y Diseño Orientado a Objetos con Aplicaciones*. México. Editorial Addison – Wesley Iberoamericana, S. A. Segunda Edición. 38-49, 120-146.
- CONALLEN, J., RATIONAL SOFTWARE. 1999. *Modeling Web Application Architectures with UML* [Documento en línea]. Disponible: http://www.rational.com/media/uml/resources/documentation/27662_webapps.pdf [Consulta: 2003, Marzo 19]. 1-9.
- JACOBSON, I. 1998. *Object-Oriented Software Engineering. A Use Case Driven Approach*. United States of America. Editorial Addison-Wesley. Cuarta Edición. 69-83; 109-130.
- MONTILVA, JONÁS. 1995. *Introducción a los Sistemas Multimedia*. Universidad de los Andes. Departamento de Computación.
- _____ 1996. Aplicando Modelos de Procesos de Software al Desarrollo de Aplicaciones Hipermedia. XXII Conferencia Latinoamericana de Informática. CLEI (Panel '96), p. 870-881.
- ZAMBRANO, JESÚS. 1995. *Enseñanza Asistida por Computador y Producción de Software Educativo (PROSDOS)*. Imprenta Universitaria de la Universidad Central de Venezuela. Caracas. 68-183.
- ### BIBLIOGRAFÍA CONSULTADA
- BIANCHINI, A. 1992. *Metodología para el Desarrollo de Aplicaciones Educativas en Ambiente Multimedia*. Trabajo de ascenso no publicado. Universidad Metropolitana. Ingeniería de Sistemas. Caracas.
- KEYES, J. 1997. *The Ultimate Multimedia Handbook*. Editorial McGraw-Hill. New York – United States of America. Segunda Edición. 18.13-18.16.
- DE WINN, ROSCH. 1996. *Todo el Poder de Multimedia*. México. Editorial Prentice Hall Hispanoamericana. Edición Premier. 84-88.
- DÍAZ, P., CATENAZZI, N., AEDO, I. 1996. *De la Multimedia a la Hipermedia*. Madrid- España. RA-MA Editorial. 130-137, 146-169.
- MARQUÉS, P. 1999. *El Software Educativo*. [Página Web en línea]. Universidad Autónoma de Barcelona. España. Disponible: <http://www.xtec.es/~pmarques/edusoft>. [Consulta: 2000, Marzo 14].