

**Universidad de las Ciencias Informáticas**  
**Facultad 8**



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

**Introducción de ApEM-L 1.0 en proyectos**  
**productivos del Ministerio de Educación.**  
**Valoración de resultados.**

**Autor:** Lisbel Rodríguez Coto.

**Tutor:** Ing. Jorge Hernández Roselló

**Co-tutor:** Msc. Febe Ángel Ciudad Ricardo

**Ciudad de la Habana, 2008.**

**“Año 50 del la Revolución”.**

## DECLARACIÓN DE AUTORÍA

Declaramos ser autores de la presente tesis y reconocemos 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 \_\_\_\_\_ del año \_\_\_\_\_.

Lisbel Rodríguez Coto

Ing. Jorge Hernández Roselló

\_\_\_\_\_  
Firma del Autor

\_\_\_\_\_  
Firma del Tutor

### AGRADECIMIENTOS

*A mi mamá y mi papá, por su apoyo incondicional y confianza; sin ellos no hubiese podido llegar hasta donde estoy, por estar siempre a mi lado apoyándome y dándome aliento para salir adelante y por lo mucho que se han esforzado para darme todo lo que he querido y tengo.*

*A mi hermano por todo su apoyo y estar siempre pendiente de mis cosas.*

*A todos mis familiares por el cariño que me han dado y por preocuparse siempre por mí.*

*A mi novia linda por su cariño, su amor, su comprensión, por todo el apoyo que me ha brindado y por hacerme feliz.*

*A todos mis compañeros durante estos 5 años, por ser mis amigos, por apoyarme y aconsejarme, por compartir tantos momentos maravillosos y a todos los que de una forma u otra han contribuido a que este sueño se haga realidad.*

*A Fidel y la Revolución por darme la oportunidad de formarme como profesional en esta maravillosa universidad donde han transcurrido los mejores años de mi vida.*

*A todos les doy las gracias...*

*...porque sin ustedes no hubiese podido ser lo que soy.*

### DEDICATORIA

*A mi mamá, mi papá y mi hermano, por ser los seres más importantes en mi vida.*

*A todos los que han confiado en mí y a todas aquellas personas que de una forma u otra han hecho posible que este sueño se haga realidad.*

**RESUMEN**

En el Ministerio de Educación (MINED) los trabajadores emplean determinados métodos de modelación en la producción de software que no son los más óptimos, por lo que se hace necesario la utilización de un lenguaje notacional que logre representar en modelos: la estructura lógica, el comportamiento y las funciones del futuro software a desarrollar. Conociendo un lenguaje de modelado (ApEM-L: Lenguaje de Modelado para Aplicaciones Educativas) creado específicamente para las aplicaciones educativas con las características objetivas y la finalidad de facilitar el proceso de producción de las mismas, en este trabajo se hace un estudio de los factores que influyen en la introducción de éste en los proyectos productivos de software educativos (SWE) del Ministerio de Educación. Se obtiene como resultado un conjunto de factores, algunos elementos a modificar de la versión ApEM-L 1.0, así como otros que no están representados y se determina el nivel de aplicación del lenguaje. Además, se propone una estrategia para la aplicación de este lenguaje de modelado en los proyectos productivos del Ministerio de Educación y se evalúa por un conjunto de expertos, lo que resulta que la misma es aceptable y tiene cierta probabilidad de éxito.

TABLA DE CONTENIDOS

<b>Introducción</b>	<b>1</b>
<b>Capítulo 1: Las notaciones y los lenguajes para la modelación de aplicaciones educativas</b>	<b>8</b>
<b>1.1 La informática y el software en la educación</b>	<b>8</b>
1.1.1 La tecnología multimedia	8
1.1.2 La tecnología hipermedia	10
1.1.3 Aplicaciones o Software educativos	10
<b>1.2 El modelamiento para la producción de software educativo</b>	<b>13</b>
1.2.1 Modelos	13
1.2.2 La modelación del software	14
1.2.3 Los paradigmas de programación	16
<b>1.3 Las notaciones en la producción de aplicaciones educativas</b>	<b>17</b>
1.3.1 Los guiones de las aplicaciones educativas	17
1.3.2 Los estereotipos y su clasificación	19
1.3.3 Las notaciones en la producción de aplicaciones educativas en Cuba	20
<b>1.4 La documentación en el software educativo</b>	<b>21</b>
1.4.1 Objetivos de la documentación	22
1.4.2 Tipos de documentación	22
1.4.3 Importancia de la documentación del software educativo	24
<b>1.5 El entorno tecnológico y científico en la representación del software educativo</b>	<b>25</b>
1.5.1 RMM: Metodología de Administración de Relaciones	25
1.5.2 OOHDM: Metodología de Diseño Hipermedia Orientada a Objetos	26
1.5.3 UML: Lenguaje Unificado de Modelado	26
1.5.4 OMMMA – L: Lenguaje para la modelación Orientada a Objetos de Aplicaciones	27
<b>1.6 Lenguaje de Modelado para Aplicaciones Educativas (ApEM-L)</b>	<b>27</b>
1.6.1 Objetivos de ApEM-L	28
<b>Conclusiones parciales</b>	<b>29</b>
<b>Capítulo 2: Un acercamiento a la utilización de ApEM-L 1.0 en el Ministerio de Educación</b>	<b>30</b>
<b>2.1 La modelación antes de la introducción de ApEM-L</b>	<b>30</b>
<b>2.2 Importancia de la utilización de ApEM-L 1.0 en el MINED</b>	<b>31</b>
<b>2.3 Factores que dificultan la introducción de ApEM-L 1.0 en los proyectos productivos</b>	<b>32</b>
<b>2.4 ApEM-L 1.0 en el Ministerio de Educación</b>	<b>33</b>
2.4.1 Elementos de ApEM-L 1.0 a modificar según los proyectos productivos del MINED	37
2.4.2 Elementos no representados por ApEM-L 1.0 en los proyectos productivos del MINED	38
<b>2.5 Nivel de aplicación de ApEM-L 1.0 en los proyectos productivos del MINED</b>	<b>38</b>
<b>Conclusiones parciales</b>	<b>38</b>
<b>Capítulo 3: Propuesta de aplicación de ApEM-L en los proyectos productivos de software educativo del Ministerio de Educación</b>	<b>40</b>

## TABLA DE CONTENIDOS

---

<b>Conclusiones parciales</b>	<b>45</b>
<b>Conclusiones Generales</b>	<b>46</b>
<b>Recomendaciones</b>	<b>47</b>

## INTRODUCCIÓN

En los últimos años en Cuba, así como en el mundo, se ha promovido el uso masivo de las tecnologías de la información y las telecomunicaciones, insertándolas en los sectores principales como la Economía, los Servicios, la Salud y la Educación. Como pasos para el cumplimiento de estos propósitos el gobierno cubano ha ejecutado acciones como: inserción en las enseñanzas primaria, secundaria y media superior de tecnologías (computadoras personales, televisores y reproductores de video) que potencien el uso de las Tecnologías de la Información y las Comunicaciones (TIC); aumento del número de instalaciones Joven Club de Computación al doble en todo el país y la creación en septiembre del 2002, de la Universidad de las Ciencias Informáticas (UCI); esta última, encaminada a promover la Industria Cubana del Software (InCuSoft), lo cual la convierte en el enlace entre el Sistema de Centros de Educación Superior y de la Ciencia con la Industria propiciando la transferencia tecnológica, porque es donde existen mejores condiciones creadas para unificar esfuerzos y conocimientos existentes, en el desarrollo de software educativos, impulsando la inserción continua de mejoras en el proceso de creación de este tipo de aplicaciones y en especial la modelación de las mismas; así como sus análisis y sus diseños. (Ciudad Ricardo, 2007)

En el terreno educativo los avances, particularmente relativos al desarrollo de los ambientes gráficos, la animación, el audio y el video, le han dado una nueva dimensión a los medios instruccionales. (Aguilar, 1997) En consecuencia se está de acuerdo con Giraldo y Muñoz al plantear que el software educativo, es el medio que más rápidamente ha asimilado estos cambios e inclusiones, convirtiéndose en una nueva alternativa válida para los procesos de enseñanza.

Producto de ello, ha dado surgimiento a la Informática Educativa y con ella a conceptos como Software Educativo, que se hace muy común en nuestros días. El propio desarrollo de este tipo de software ha traído como consecuencia variaciones importantes en los flujos de los procesos productivos y en los artefactos de modelación para generar su documentación en el uso de las tecnologías multimedia e hipermedia.

La definición de software educativo es expresada como: “programas de ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje”. (Marqués, 1996) El uso del software educativo en el proceso de enseñanza-aprendizaje permite mejorar en el estudiante las destrezas cognitivas. Este tipo de software fomenta el análisis de problemas, facilita el trabajo en grupo, provee soporte en actividades docentes,



en el sentido más amplio, mejora las habilidades del pensamiento y la resolución de problemas (Díaz-Antón, 2002). Para lograr todo esto, el software debe ser de calidad. Cuando se hace referencia a la calidad del software educativo, “se requiere un producto que satisfaga tanto las expectativas de los docentes como de los alumnos, a un menor costo, libre de errores y cumpliendo con ciertas especificaciones instruccionales y tecnológicas”. (Pressman, 2002)

“En Cuba se acumulan más de 20 años de experiencia en la elaboración de software educativo. En estos resultados han tenido una importante participación los Ministerios de Educación y Educación Superior por medio de los Centros de estudio de software y las Universidades respectivamente, así como la organización de Joven Club de conjunto con el sector empresarial”. (MIC, 2003)

El personal dedicado a esta actividad eran profesionales del área pedagógica que por las necesidades de la enseñanza comenzaron a incursionar en el desarrollo de software educativos como iniciativa para amenizar el aprendizaje y dado ello era lógico que no se dedicara el suficiente tiempo a la modelación de este tipo de producto y la documentación de estos.

A lo largo del tiempo mencionado aparecieron en la industria internacional de software metodologías como la *Relationship Management Methodology* (RMM) para la guía de procesos de desarrollo de aplicaciones con tecnología multimedia; la técnica más utilizada en el desarrollo de aplicaciones de corte educativo, que anexaban una notación para el modelado de este tipo de software. Cumplió su objetivo durante un buen tiempo, pero tecnológicamente no logra responder a todas nuestras necesidades actuales en la representación computacional de las soluciones informáticas educativas. De manera similar, en Cuba se trabajó en nuestras necesidades surgiendo la metodología MultiMet, que igualmente traía asociada un lenguaje de representación de este tipo de software. Al mismo tiempo, como ya fue mencionado, evolucionaron las técnicas y lenguajes de programación, y con estas mejoraron las formas de programar el software educativo, llegando en la actualidad a utilizar en mayor medida técnicas orientadas a eventos y en sus mejores exponentes orientados a objetos completamente. (Ciudad Ricardo, 2007)

La presentación en el año 1998 por sus autores: I. Jacobson, G. Booch y J. Rumbaugh de la primera versión del Lenguaje Unificado de Modelado (UML) para la modelación de software de propósito general orientado a objetos, fue de un gran impacto en la industria internacional y tuvo una excelente aceptación, al punto de convertirse en un estándar mundial en este sentido. (Ciudad Ricardo, 2007)

“Desafortunadamente, UML no soporta todos los aspectos de las aplicaciones multimedia de una forma adecuada e intuitiva. Especialmente, las características del lenguaje para modelar los aspectos de la interfaz de usuario, no se aplican explícitamente en los entornos multimedia. Otros conceptos de UML no son lo formalmente aplicables a la multimedia y de ser utilizados tal y como han sido planteados complicarían la modelación de este tipo de aplicaciones. Por estas razones, y gracias a las facilidades de extensión, si bien permitidas en UML, y he aquí su riqueza como lenguaje de modelado, es que sus principales conceptos y notaciones son aplicables a los entornos multimedia, más se hizo necesario el desarrollo de una extensión para este tipo de aplicaciones denominada Lenguaje Orientado a Objetos para la Modelación de Aplicaciones Multimedia (OMMMA-L), que facilita el modelado de un gran rango de aspectos de aplicaciones multimedia interactivas de una forma integrada y comprensiva.” (Sauer, 2001)

El software educativo cubano, como respuesta y representante del sistema educacional de la isla, posee un conjunto de características que lo diferencian sustancialmente del que se produce en el resto del mundo; potenciando áreas como: (Ciudad Ricardo, 2007)

1. Procesamiento de volúmenes de información considerables en varios formatos de presentación (animaciones, videos, imágenes, sonidos, textos e hipertextos).
2. Utilización de técnicas de Inteligencia Artificial (IA) para el tratamiento de entornos educacionales a la medida del usuario.
3. Concepción de seguimientos pedagógicos o desarrollo de la traza de utilización y avance cognitivo del usuario, lo que implica la utilización de bases de datos.
4. Posibilidad de retroalimentación del tutor o profesor guía del proceso docente educativo donde se utiliza el software.
5. Posibilidad del profesor de adecuar el funcionamiento de la aplicación a las características de cada uno de los estudiantes o grupos de estos.
6. Posibilidad del profesor de adicionar elementos multimedia al software para un mejoramiento y actualización del mismo.

Si señalamos además que de los diferentes tipos de software educativos existentes según la clasificación ofrecida por Marqués, 1996: tutoriales, bases de datos, simuladores, constructores y programas herramienta; el software educativo cubano es un producto que fusiona elementos representativos de cada uno de los tipos específicos mencionados anteriormente; aumentando considerablemente la complejidad del tipo de software y su modelación, hecho en Cuba.

El ámbito informático actual para el desarrollo de software educativo, establece grandes retos a las metodologías, procesos de desarrollo y lenguajes notacionales de los software existentes, ambos con una aceptación generalizada; entre los que se encuentra la representación de las características pedagógicas y funcionales de este tipo de aplicaciones, máxime con las características y los pilares de la pedagogía cubana. (Ciudad Ricardo, 2007)

“Durante la actividad de evaluación y síntesis de la solución, el analista crea modelos del sistema en un esfuerzo de entender mejor el flujo de datos y de control, el tratamiento funcional y el comportamiento operativo y el contenido de la información. El modelo sirve como fundamento para el diseño del software y como base para la creación de una especificación del software.” (Pressman, 2002)

Estas funciones descritas por Pressman en el párrafo anterior, son las que corresponden a los especialistas: convertir en modelos ingenieriles informáticos, con capacidad de diseño e implementación, las necesidades y requerimientos del software educativo solicitados por las diferentes instituciones nacionales e internacionales. Para ellos se hace necesario la utilización de un lenguaje notacional que logre representar en modelos: la estructura lógica, el comportamiento y las funciones del futuro software a desarrollar enmarcadas en un determinado contexto pedagógico. (Ciudad Ricardo, 2007)

Teniendo en cuenta que la institución que ocupa el centro de la investigación es el Ministerio de Educación; que las principales funciones de sus especialistas a la hora de modelar en el proceso de producción son: convertir en modelos ingenieriles informáticos con capacidad de diseño e implementación, las necesidades y requerimientos de los software actuales y que estos trabajadores emplean determinadas métodos de modelación para la producción de las aplicaciones que no son los más óptimos en el proceso productivo, porque no les permite representar la totalidad de los elementos y artefactos que se necesitan para la elaboración de sus software educativos de una forma adecuada. También preocupa la importancia que tienen dichos productos y su proceso de producción por lo que representan para nuestro sistema educacional. Entonces, se hace necesario la utilización de un lenguaje notacional que logre representar en modelos: la estructura lógica, el comportamiento y las funciones del software a desarrollar. Conociendo un lenguaje de modelado creado específicamente para la modelación de aplicaciones educativas con las características objetivas y la finalidad de facilitar el proceso de producción de las mismas, se identificó como **Problema científico**: ¿Qué factores dificultan la aplicación de ApEM-L 1.0 en los proyectos productivos de software educativo del Ministerio de Educación?

El problema descrito anteriormente genera como **Objeto de estudio** de esta investigación: Los proyectos productivos de software educativo del Ministerio de Educación y como **Campo de acción**: El proceso de producción de software educativo en el Ministerio de Educación.

Para darle solución al problema científico mencionado se planteó como **Objetivo general**: Identificar los factores que dificultan la aplicación de ApEM-L 1.0 en los proyectos productivos del Ministerio de Educación.

Se traza como **Idea a defender**: La aceptación de ApEM-L 1.0 está frenada por los factores que limitan su introducción en los proyectos productivos de software educativos del Ministerio de Educación.

Al mismo tiempo se plantearon los siguientes **Objetivos específicos**:

1. Analizar el proceso de modelación de los proyectos productivos de software educativo en el Ministerio de Educación antes y después de la introducción de ApEM-L.
2. Identificar los elementos no representados aún por ApEM-L 1.0 en los proyectos productivos de software educativo del Ministerio de Educación.
3. Determinar los elementos de ApEM-L 1.0 a modificar a partir de las necesidades de los proyectos productivos de software educativo del Ministerio de Educación.
4. Determinar el nivel de aplicación (bajo, intermedio, alto) de ApEM-L 1.0 en los proyectos productivos de software educativo del Ministerio de Educación donde se haya introducido.
5. Proponer una estrategia para realizar el proceso de introducción de ApEM-L en los proyectos productivos de software educativos del Ministerio de Educación.

La investigación científica se guió por el conjunto de **Tareas** que se describen a continuación para darle cumplimiento a los objetivos tanto generales como específicos planteados:

1. Realizar un estudio de la documentación y el factor humano que interviene en el proceso de producción de software educativo del Ministerio de Educación.
2. Realizar un estudio de la metodología de trabajo de los proyectos productivos de software educativo del Ministerio de Educación.
3. Realizar la identificación de los elementos faltantes así como los modificables en ApEM-L 1.0 teniendo en cuenta las necesidades de los proyectos productivos de software educativo del Ministerio de Educación.
4. Evaluar y caracterizar la aplicación de ApEM-L 1.0 en los proyectos productivos del Ministerio de Educación.

En el desarrollo de la investigación científica se han utilizado un conjunto de Métodos Científicos para la obtención, procesamiento y llegada a conclusiones. Dentro de estos podemos mencionar los siguientes:

Métodos Empíricos:

1. Observación: para obtener el registro del avance de los proyectos productivos que sirvieron de muestra, en la clasificación y consignación de los acontecimientos pertinentes para la investigación.
2. Entrevista: utilizado para obtener información verbal y real de distintos miembros internos y externos a la actividad de modelación en los proyectos, así como a los especialistas pedagógicos inmersos en la investigación.
3. Encuesta: para conformar las estadísticas de la situación actual con las notaciones existentes, así como las de impacto de la solución propuesta en la investigación.

Esta investigación tiene un carácter dialéctico materialista y vincula la metodología cualitativa con la metodología cuantitativa, basada fundamentalmente en métodos empíricos, que buscan recoger el estado de opinión y sugerencias de los implicados en el proceso de producción del software educativo en el Ministerio de Educación.

Este estudio fue realizado tomando como **Población**: el conjunto de proyectos productivos del Ministerio de Educación dedicados al desarrollo de software educativo. La **Muestra** estuvo compuesta por tres proyectos productivos del Ministerio de Educación que han trabajado con los software educativos: “Español para no hispanohablantes”, “A jugar” y “Multisaber”.

El trabajo está estructurado de la siguiente manera:

**Introducción:** donde se hace una valoración de la necesidad social del tema, se enuncia la situación problemática, problema científico, objetivos, objeto de estudio, campo de acción y métodos aplicados.

**Capítulo 1: Las notaciones y los lenguajes para la modelación de aplicaciones educativas.**

Se hace una conceptualización de lo que se entiende por Software Educativo, se estudia el estado del tema a nivel internacional, se hace mención a las características y la aplicación en el mundo y en Cuba. Se realiza además una descripción de los elementos relacionados con las notaciones y los lenguajes para la modelación de estas aplicaciones y el entorno científico tecnológico.

**Capítulo 2: Un acercamiento a la utilización de ApEM-L 1.0 en el Ministerio de Educación.**

En este capítulo se realiza una breve descripción de la modelación que se realiza en el proceso de producción de software educativo en el Ministerio de educación y se hace el análisis de los factores que dificultan la introducción de este lenguaje de modelado. También se identifican algunos elementos a modificar y otros que no están representados por esta versión, determinando finalmente el nivel de aplicación de la notación.

### **Capítulo 3: Propuesta de aplicación de ApEM-L en los proyectos productivos de software educativo del Ministerio de Educación.**

En este capítulo se propone una estrategia para la introducción de ApEM-L en los proyectos productivos de software educativos del Ministerio de Educación y se realiza una valoración de la misma apoyada en los criterios expresados por algunos expertos.

**Conclusiones, recomendaciones, referencias bibliográficas, bibliografía y anexos.**

## CAPÍTULO 1: LAS NOTACIONES Y LOS LENGUAJES PARA LA MODELACIÓN DE APLICACIONES EDUCATIVAS

La Ingeniería de software designa un conjunto de técnicas destinadas a la producción de un programa de computadora, más allá de la actividad de programación, que son tan importantes como la mencionada. Además requiere llevar a cabo otras tareas, de las cuales, una es la Documentación, donde se generan documentos que abordarán los elementos que sirven de ayuda al interactuar con los trabajos realizados. En este capítulo se da un fundamento teórico a la investigación presentada. Conceptos y teorías como Software Educativo, multimedia, hipermedia, modelos, notación, estereotipos, lenguajes de modelación y herramientas serán trabajados con detenimiento para servir como fundamentación científica de lo investigado.

### ***1.1 La informática y el software en la educación***

En la actualidad el uso de la tecnología es imprescindible para múltiples tareas. La educación lo ha implementado desde hace un tiempo como herramienta pedagógica y siendo bien utilizada por alumnos, docentes y directivos, puede ser muy eficaz y hasta sobre cumplir con los objetivos fundamentales de la institución; sólo hay que ser cuidadoso y estar correctamente preparado. Dado todo esto han surgido conceptos como Informática Educativa y Software Educativos.

La informática resulta una herramienta pedagógica muy útil para enfrentar el mundo que nos rodea, pero debe ser considerada con el merecido compromiso por cualquier institución. El uso del software como parte de la enseñanza, como herramienta de trabajo de alumnos y de docentes, como recurso administrativo o como medio de comunicación masivo es una manera de aceptar y aprovechar estos cambios sociales. Este desarrollo ha traído como consecuencia variaciones importantes en los flujos de los procesos productivos y en los artefactos de modelación, los que generan la documentación en el uso de las tecnologías multimedia e hipermedia.

#### ***1.1.1 La tecnología multimedia***

La tecnología multimedia aparece en el año 1984, cuando la Apple Computer lanza la primera variante de Macintosh, la cual tenía amplias capacidades de reproducción de sonidos; apoyándose en su sistema operativo, propicio para el diseño gráfico y la edición. En 1987, con los videojuegos, los avances en las Tecnologías de la Información y las Comunicaciones (TIC) y el desarrollo por la Philips

del Disco Compacto (CD), aumentan las posibilidades para el desarrollo de la tecnología multimedia. En 1992, se presenta en la feria “Consumer Electronic Show” (CES) de Las Vegas, el CD multiusos, que luego dio paso a la televisión interactiva. En 1993, el nuevo término de multimedia, obliga a una revisión de los conceptos que permiten el funcionamiento de la nueva tecnología, buscando el desarrollo de estándares para lograr la uniformidad en el avance tecnológico. Hoy se ha desarrollado una nueva técnica que surgió a partir de la multimedia: la hipermedia, que tiene su base en el hipertexto, permitiendo al usuario un recorrido o navegación por distintos archivos o partes del programa de acuerdo a sus intereses personales.

Para Héctor Schwabe, investigador de la UAM multimedia ha existido como concepto desde hace 40 años aunque como realidad sólo desde 1989. Multimedia se define como la interacción de medios múltiples: sonido, texto, voz, video y gráficos.

La multimedia tiene varias aplicaciones entre las cuales se pueden mencionar las siguientes: (Corrales Díaz, 1994)

1. En la diversión y el entretenimiento: Multimedia es la base de los juegos de video, pero también tiene aplicaciones en pasatiempos de tipo cultural como cuentos infantiles interactivos, exploración de museos y ciudades a manera de visitas digitales interactivas.
2. Multimedia en los negocios: Las principales aplicaciones se dan en la inducción, capacitación y adiestramiento de personal, la disposición rápida, accesible y procesamiento de altos volúmenes de información.
3. En publicidad y marketing: Sus ejemplares son; la presentación multimedia de negocios, de productos y servicios, la oferta y difusión de los productos y servicios a través de los kioscos de información.
4. En la difusión del saber y conocimiento: La característica de la interactividad de multimedia, que permite navegar por el programa y buscar la información sin tener que recorrerlo todo, logra que la tecnología se aplique en los nuevos medios de modos diferentes y se use de forma alternativa. Y por último entre los muchos beneficios que ofrecen la tecnología multimedia se puede mencionar: La flexibilidad, ya que el material digital puede ser fácil y rápidamente actualizado y presentado a través de innumerables medios.
5. En la administración: Permite revisar y analizar reportes de clientes realizados por video, de manera más rápida y efectiva.

Por lo que se concluye con que es una tecnología digital de comunicación, constituida por la suma de Hardware y Software, con el objetivo de humanizar la máquina; integra medios múltiples por medio de



la computadora: sonido, texto, voz, video y gráficas; propicia la interacción con la máquina y los programas de cómputo a partir de aplicaciones concretas que requieren de tal integración.

### **1.1.2 La tecnología hipermedia**

Vannevar Bush, en la década de los 40, y Theodor Holme Nelson, en los 60, se consideran los artífices de la estructuración no lineal y de la interconexión de la información, asuntos que constituyen conceptos claves para el desarrollo de la interactividad informática aplicada a la comunicación. El hipertexto no es más que un conjunto de bloques de texto interconectados por nexos, que forman diferentes itinerarios para el usuario, donde estos nexos “electrónicos” unen fragmentos de texto internos o externos a la obra (soportes cerrados – off line – o abiertos – on line – respectivamente), creando un texto que el lector experimenta como no lineal o, mejor dicho, como multilineal o multisequencial.

La hipermedia surge como resultado de la fusión de dos tecnologías, el hipertexto y la multimedia. La tecnología multimedia es la que permite integrar diferentes medios (sonido, imágenes, secuencias...) en una misma presentación.

La hipermedia, por tanto, es la tecnología que permite estructurar la información de una manera no-secuencial, a través de nodos interconectados por enlaces sobre los diferentes formatos de información.

Estas hipermedias y multimedias pretenden resolver el problema del procesamiento lineal de la información por el receptor, como ocurre en el libro de texto. Por el contrario, la información se puede construir desde diferentes trayectorias y alternativas, y con diferentes tipos de códigos. Estas trayectorias pueden limitarse por el autor del programa, para evitar problemas de desorientación en el usuario (Pastor, 1997), pudiendo soportar la autoría de documentos complejos y el procesamiento de ideas, especialmente en entornos de trabajo colaborativos. Para la aplicación de esta facilidad los alumnos necesitan la capacidad de anotar nodos de información y ser capaces de colaborar con otros estudiantes y con el profesor sobre unidades específicas de información. Aún más importante que esto es que deben ser capaces de construir su propio sistema de conocimiento a través de la investigación, abstracción, readaptación y adición de conocimientos a una base de datos existentes.

### **1.1.3 Aplicaciones o Software educativos**

Según Cataldi el software educativo puede definirse como: “los programas de computación realizados con la finalidad de ser utilizados como facilitadores del proceso de enseñanza y consecuentemente de

aprendizaje, con algunas características particulares tales como: la facilidad de uso, la interactividad y la posibilidad de personalización de la velocidad de los aprendizajes.” (Cataldi, 2000)

Pere Marqués plantea que el software educativo, son:”programas educativos y programas didácticos como sinónimos para designar genéricamente los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje”. (Marqués, 1996)

Por lo tanto, sintetizando, el Software Educativo es una herramienta destinada a facilitar y mejorar el proceso de enseñanza y aprendizaje en los centros educacionales, además permite desarrollar habilidades cognitivas en los alumnos.

“Todas las formas de software educativos han sido absorbidas por esta tecnología, lo cual no es pura casualidad, sino el resultado de un proceso histórico que ha pretendido combinar los diferentes métodos para transmitir la información, en esperanza de una mayor calidad del propio proceso de adquisición de conocimientos.” (Lee, 2000) “Las tecnologías de la información y la comunicación ofrecen grandes posibilidades al mundo de la educación. Pueden facilitar el aprendizaje de conceptos y materias, ayudar a resolver problemas y contribuir a desarrollar las habilidades cognitivas. Las áreas de aplicación de todas estas técnicas, englobadas en lo que normalmente se denominan informática educativa, son tanto la enseñanza reglada, comúnmente denominada curricular, como la formación en todos los ámbitos posibles. De esta manera, se nos presenta la posibilidad de aprovechar la tecnología para crear situaciones de aprendizaje y enseñanza novedosas.” (Díaz-Antón, 2002)

Una de las clasificaciones más conocida fue dada por los norteamericanos Stephen M. Alessi y Stanley Trollip, cuando plantearon que el uso de las computadoras en la educación podía dividirse en: (Pérez Fernández, 1999)

1. Uso administrativo.
2. Enseñanza sobre computadoras.
3. Enseñanza con computadoras, dentro de la cual se enmarca mayoritariamente el área de la informática educativa.

Según Raúl Rodríguez Lamas, la Informática Educativa se puede definir como la parte de la ciencia de la Informática encargada de dirigir, en el sentido más amplio, todo el proceso de selección, elaboración, diseño y explotación de los recursos informáticos dirigido a la gestión docente,

entendiéndose por éste la enseñanza asistida por computadora y la administración docente. (Rodríguez, 2000)

La verdadera revolución se ha producido a raíz de la generalización de los CD-ROMs como soporte de información para las aplicaciones multimedia y al mismo tiempo una auténtica revolución en el software, sobre todo en la interfaz de usuario. Vinculando la enseñanza por computadora con las nuevas tecnologías multimedia, surge lo que conocemos como los software multimedia educativos; herramientas poderosas dentro del contexto de la Informática Educativa.

“Un software multimedia educativo es una aplicación informática, soportada sobre una bien definida estrategia pedagógica, y que apoya directamente el proceso de enseñanza-aprendizaje constituyendo un efectivo instrumento para el desarrollo educacional del hombre del próximo siglo.” (Pérez Fernández, 1999)

Los software multimedia educativos, permiten agrupar una serie de factores presentes en otros medios, pero a la vez agregar otros hasta ahora inalcanzables: (Pérez Fernández, 1999) (Fernández, 2000)

1. Permite la interactividad con los estudiantes, retroalimentándolos y evaluando lo aprendido
2. Facilita las representaciones animadas.
3. Incide en el desarrollo de las habilidades a través de la ejercitación. Permite simular procesos complejos.
4. Reduce el tiempo que se dispone para impartir gran cantidad de conocimientos facilitando un trabajo diferenciado, introduciendo al estudiante en el trabajo con los medios computarizados.
5. Facilita el trabajo independiente y a la vez un tratamiento individual de las diferencias.
6. Permite al usuario (estudiante) introducirse en las técnicas más avanzadas.
7. Posibilidades de estudiar procesos que no es posible observar directamente.
8. Autocontrol del ritmo de aprendizaje.

Concluyendo, los buenos software multimedia formativos dentro del marco de la Informática Educativa, son eficaces y facilitan el logro de sus objetivos, debido al supuesto buen uso por parte de los estudiantes y profesores, a una serie de características que atienden a diversos aspectos funcionales, técnicos y pedagógicos. A continuación se mencionan algunos: (Autores, 1998)

1. Facilidad de uso e instalación.
2. Versatilidad (adaptación a diversos contextos).
3. Calidad del entorno audiovisual.

4. La calidad en los contenidos (bases de datos).
5. Navegación e interacción.
6. Originalidad y uso de tecnología avanzada.
7. Capacidad de motivación.
8. La documentación.

## **1.2 El modelamiento para la producción de software educativo**

En cualquier proyecto de ingeniería como la construcción de un gran edificio, un avión, la construcción de un procesador de textos o un software de comunicaciones para Internet, requieren de etapas de modelamiento que permitan experimentar y visualizar el sistema que se construirá.

En todas las disciplinas de la Ingeniería se hace evidente la importancia de los modelos ya que describen el aspecto y la conducta de algo que puede existir, estar en un estado de desarrollo o estar en un estado de planeación. Es en este momento cuando los diseñadores del modelo deben investigar los requerimientos del producto y dichos requerimientos pueden incluir áreas tales como: funcionalidad o comportamiento y confiabilidad. Además, a menudo, el modelo es dividido en un número de vistas, cada una de las cuales describe un aspecto específico del producto o sistema en construcción.

El modelado es muy útil para los grandes sistemas, aún en aplicaciones de pequeño tamaño también se obtienen beneficios, sin embargo es un hecho que entre más grande y complejo es el sistema más importante es el papel que juega el modelado porque permite una mejor comprensión, ya que tales son muy difíciles de entender en su totalidad.

### **1.2.1 Modelos**

Un modelo es una representación, en cierto medio, de algo en él mismo u otro medio. El modelo capta los aspectos importantes de lo que estamos modelando, desde cierto punto de vista, y simplifica u omite el resto. La ingeniería, la arquitectura y muchos otros campos creativos usan modelos. El modelo tiene semántica y notación y puede adoptar varios formatos que incluyen texto y gráficos. Es la representación más formal para las divisiones que propicien el entendimiento y la administración de la complejidad de estos. (Booch, 2000) (Rumbaugh, 1997)

En el ámbito de la Ingeniería de Software (ISW) desarrollar modelos, implica representar los sistemas de software en cada una de sus partes, desde diferentes vistas: lógicas, estructurales, de comportamiento, de arquitectura, de ensamblaje; para de esta manera permitir el consenso en el

lenguaje, el entendimiento del problema y las soluciones y la toma de decisiones de los equipos de desarrollo y los clientes. (Ciudad Ricardo, 2007)

Los modelos, así como la modelación de los sistemas; permiten propósitos como los que plantean tres de los investigadores más autorizados en la actualidad a definir estos términos: Booch, Jacobson y Rumbaugh; en su texto *“El Lenguaje Unificado de Modelado. Manual de Referencia”*. (Booch, 2000)

1. Para capturar y enumerar exhaustivamente los requisitos y el dominio de conocimiento, de forma que todos los implicados puedan entenderlos y estar de acuerdo con ellos.
2. Para pensar en el diseño de un sistema.
3. Para capturar decisiones del diseño en una forma mutable a partir de los requisitos.
4. Para generar productos aprovechables para el trabajo.
5. Para organizar, encontrar, filtrar, recuperar, examinar y corregir la información en grandes sistemas.
6. Para explorar económicamente grandes soluciones.
7. Para domesticar los sistemas complejos.

## **1.2.2 La modelación del software**

Similar al concepto de modelo, el término modelación al ser una conjugación, debemos estudiarlo a través del verbo *“modelar”*, en términos de informática esto significaría: representar sistemas, ajustándose a un conjunto de normas y signo de representación determinados, según la notación correspondiente, precisando los modelos de importancia para el entendimiento de las partes y el todo del software y su entorno.

La acción de modelar un software, implica en su hacer el tener en cuenta al construir modelos aspectos como: (Ciudad Ricardo, 2007)

- Arquitectura del sistema que se construye: se hace necesario la observación de la arquitectura que se utiliza o utilizará en la solución, para lograr una estructura sólida y sostenible de la aplicación en el tiempo, así como un entendimiento del o los principios de funcionamiento de la aplicación que se desarrolla.
- Disponibilidad del hardware para la solución propuesta: no todas las soluciones pueden convertirse en la óptima que se desea, sino que al construir nuestros sistemas no podemos olvidar la tecnología con la que se cuenta o aquella que se pueda adquirir en el futuro inmediato para sustentar la propuesta de solución.

- Relación Costo/Beneficio económico de la solución: factor este de gran importancia en las soluciones que se modelen; pues desde el momento de la representación de la solución, es donde se deben valorar las posibles soluciones desde el punto de vista económico, para permitir seleccionar la de mejor relación en cada caso al discernir entre los componentes materiales y humanos a intervenir en las soluciones.
- Posibilidades de cambios futuros y asimilación de nuevas tecnologías o soluciones informáticas: los modelos de representación de un sistema, tienen que estar preparados o permitir, sin tocar el programa o código fuente del mismo, hacer trabajo de mesa para el análisis de cambios, asimilación de nuevos requerimientos o de nueva tecnología. Esto hace que los modelos tengan que ser entendibles y exactos en la representación de elementos de esta naturaleza.
- Posibilidad de reutilización de partes o componentes de otros sistemas ya desarrollados anteriormente: hoy día, al desarrollar sistemas cada vez más grandes e intrincados, así como existiendo problemáticas similares en muchos lugares del planeta, no es ventajoso en momento alguno repetir soluciones en distintas partes del mundo, sino todo lo contrario, trabajar en colectivo, para la generación de mejores soluciones entre todos. Eso hace que la solución sea la reutilización de partes ya existentes, pero que deben de representarse igualmente en los modelos, no importa si como cajas negras, desconocidas para los desarrolladores o como componentes ampliamente representados para su entendimiento en la solución.

En los últimos años, la construcción de modelos orientados a objetivos se ha convertido en una herramienta habitual en distintos ámbitos tales como el de la ingeniería de requisitos y el del modelado de procesos en organizaciones. El *Lenguaje de Modelado de Objetos* (Object Constraints Language) es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos.

Algunas organizaciones los usan extensivamente en combinación con una metodología de desarrollo de software para avanzar de una especificación inicial a un plan de implementación y para comunicar dicho plan a todo un equipo de desarrolladores.

Algunos metodólogos del software orientado a objetos distinguen tres grandes "generaciones" cronológicas de técnicas de modelado de objetos. (Pressman, 2002)

- En la primera generación, tecnólogos aislados y grupos pequeños desarrollaban técnicas que resolvían problemas que se encontraban de primera mano en los proyectos de desarrollo orientado a

objetos. En esta generación se incluye a autores y técnicas como Rumbaugh, Jacobson, Booch, los métodos formales, Shlaer-Mellor y Yourdon-Coad.

- En la segunda generación se reconoció que muchas de las mejores prácticas pertenecían a diferentes métodos del fragmentado terreno de la metodología orientada a objetos. Se realizaron múltiples intentos para integrar dichas técnicas en marcos coherentes tales como FUSION. En cualquier caso, la comunidad del software orientado a objetos empezaba a reconocer los beneficios que la estandarización de las técnicas conllevaría: abandono de las "buenas" formas de hacer las cosas en favor de "la" manera adecuada, que permitiría un lenguaje y unas prácticas comunes entre los diferentes desarrolladores.
- La tercera generación consiste en intentos creíbles de crear dicho lenguaje unificado por la industria, cuyo mejor ejemplo en la actualidad es UML.

### **1.2.3 Los paradigmas de programación**

A lo largo de la historia, la Ingeniería de Software ha estado determinada por varias etapas marcadas por diferentes paradigmas de programación, porque ayudan en la conceptualización y representación del entorno software. Dado que están fuertemente relacionados con las notaciones y los lenguajes y estos se identifican con el establecimiento de un conjunto de signos, reglas, normas y semántica para la representación de la estructura y el comportamiento de los sistemas informáticos, permitiendo de esta forma la homogenización de los términos utilizados por el equipo de desarrollo y la construcción de modelos de representación del futuro software.

A lo largo de la historia de la computación, el desarrollar lenguajes que permitan expresar la lógica del funcionamiento humano a través de programas entendibles para la computadora, ha transitado por varios paradigmas o formas, dentro de las cuales los más significativos son:

- Paradigma estructurado.
- Paradigma Orientado a Eventos. (OE)
- Paradigma Orientado a Objetos. (OO)

Existiendo cada uno de estos paradigmas anteriores se presentaron metodologías de desarrollo de software que anexaban a estas notaciones, simbologías o lenguajes para la representación y descripción de las soluciones informáticas; lo que atentó contra el entendimiento entre algunos desarrolladores de diferentes regiones del mundo y las soluciones ofrecidas por otros. Además, producto a la complejidad de los lenguajes de programación de computadoras, no eran abundantes los

desarrolladores de sistemas informáticos, lo que hacía poco necesario la documentación de las aplicaciones.

Con el desarrollo de la tecnología de la electrónica, las comunicaciones y la telefonía, el tratamiento de la información cambió vertiginosamente. Se produjo un salto tecnológico muy grande que trajo como consecuencia la necesidad de aunar el trabajo de especialistas de diferentes campos y con diferentes lenguajes formales y de representación de los sistemas. Se hizo necesario el comienzo del desarrollo de lenguajes notacionales, que permitieran representar los paradigmas de las programaciones existentes, para aunar los esfuerzos y disminuir los costos del desarrollo de sistemas.

Las notaciones han estado siempre muy ligadas a los paradigmas de programación. Tanto es así que los lenguajes notacionales, como los humanos, solo nos sirven para representar determinados conceptos de un determinado paradigma, no lo contrario. De esta manera se llegó a lo largo de cuatro décadas a notaciones para la modelación estructurada, para la modelación orientada a eventos y para la modelación orientada a objetos en la actualidad. (Ciudad Ricardo, 2007)

### ***1.3 Las notaciones en la producción de aplicaciones educativas***

Con la inserción de las tecnologías de la información y las comunicaciones en la sociedad, ha surgido un interés irrevocable acerca del conocimiento de las mismas, causando cambios visibles en la mayor parte de los sectores sociales.

Como consecuencia de este avance vertiginoso, el gobierno cubano ha llevado adelante numerosos programas de la revolución dirigidos a la esfera de la informática; para un escolar hasta el más de los humildes ciudadanos. Uno de los proyectos de mayor repercusión ha sido el desarrollo de software educativo para la escuela cubana, desde la enseñanza primaria hasta la universitaria. Son ya numerosos los paquetes de software producidos en el país en los pasados años.

#### ***1.3.1 Los guiones de las aplicaciones educativas***

Las nuevas revoluciones tecnológicas van generando nuevos lenguajes que hay que aprender a articular correctamente con el fin de utilizar del modo más eficiente posible esa tecnología que se va poniendo a nuestro alcance:

"La multimedia se convierte así en una nueva cota comunicativa, donde se mejoran, perfeccionan e integran los elementos audiovisuales creando un nuevo lenguaje rico en matices, expresiones y



formas, con el añadido de una interacción presumible entre hombre y relato (vehiculado mediante la máquina) que rompe la unidireccionalidad de la mayoría de los media (medios de comunicación) conocidos" (Pérez Huertas, 1998)

Muchos autores hacen una clasificación más sencilla siguiendo el esquema clásico de las producciones audiovisuales. A diferencia de éstas, en el guión multimedia:

"(...) se recogerán de manera exhaustiva los elementos que han de intervenir en cada una de las pantallas o secuencias, las acciones que se desarrollarán, el grafismo utilizado y el "tempo" que mantendrán, es decir, el orden de representación y el tiempo del mismo. También se debe describir detalladamente cuándo sucederá, bien por la intervención directa del usuario bien por otro tipo de causas, como por ejemplo la ausencia de acciones por parte de éste". (Pérez Huertas, 1998)

Bou con su propuesta: El guión multimedia: "Las aplicaciones multimedia normalmente son diseños con perfil de destino, es decir, se conciben para ser utilizadas por un tipo determinado de usuario. Por ello, al enfrentarse al diseño de una aplicación, deberá indagar sobre las características de su "cliente". (Bou, 1997)

Da paso al establecimiento de una serie de parámetros para transmitir la información que va a ir contenida en la multimedia:

- **Guión de contenido:** Indica el material textual que se va a utilizar en las diferentes secuencias y la manera en la que se va relacionando mediante una jerarquización conceptual que irá de lo más importante o lo más general o específico y que deberá transmitirse en forma muy clara en el guión.
- **Guión narrativo:** Establece cómo se va a presentar esa información. Se corresponde a lo que conocemos también como guión literario, indicando el punto de vista y el estilo. Por tanto, se dará forma al contenido establecido en el paso anterior.
- **Guión icónico:** Marca las imágenes que se tiene disponibles, sean gráficos, fotos, figuras, cuadros, imágenes de video o animación, y en qué momento de la narración serán utilizadas. Para que resulte más fácil, suele usarse un código que identifica cada imagen y en qué secuencia va.
- **Guión de sonido:** Se debe desarrollar en forma sincrónica con el guión narrativo. Los registros de sonido deberán ser secuenciales, y esta secuencialidad se indicará mediante un número de orden. Los registros de sonido pueden ser directos o indirectos según la fuente de la que se haya tomado (un

registro directo es por ejemplo, la grabación en off de una voz que realiza un relato y, uno indirecto, cualquier música empleada).

- **Guión técnico:** Es elaborado por el profesional informático a medida que va comprendiendo la idea del docente. Consiste en definir las bases de la realización, la metodología, los programas a utilizar, los formatos de presentación, diseño de pantalla, los efectos a utilizar en cada parte.

Es de notar con claridad como en la descripción ofrecida sobre los guiones multimedia, se distingue la recarga considerable del mismo en los aspectos que desde el punto de vista visual o de interés para el usuario en una aplicación educativa existen. Sin embargo es de difícil consecuencia la generación de una documentación de interés ingenieril a partir de un guión de este tipo. En la última década se ha avanzado en este campo, logrando en la modelación de entornos educativos, la incorporación de lenguajes notacionales de propósito general como UML, o sus extensiones como OMMMA – L, ambos descritos en este mismo capítulo. No obstante estos lenguajes no logran denotar todos los elementos, tanto ingenieriles como pedagógicos de las aplicaciones educativas cubanas en sus gráficos y semánticas utilizadas. (Ciudad Ricardo, 2007)

### **1.3.2 Los estereotipos y su clasificación**

“Un estereotipo en un lenguaje de modelación es un mecanismo bien formado para expresar extensiones definibles por el usuario, refinamientos o redefiniciones de elementos del lenguaje sin (directamente) modificar el meta – modelo del lenguaje.” (Berner, 2001)

“Los estereotipos clasifican los objetos de acuerdo a su uso, independientemente de su primera clasificación por clases y jerarquía de clases. Esta clasificación ayuda a organizar mejor un modelo y mejora el entendimiento del modelo.” (Berner, 2001)

Según la clasificación propuesta por Berner y otros autores en el año 2001, los estereotipos pertenecen a alguna de las siguientes cuatro categorías:

1. Estereotipos decorativos.
2. Estereotipos descriptivos.
3. Estereotipos restrictivos.
4. Estereotipos redefinitorios.

Comúnmente los cambios o extensiones que se producen en un lenguaje son cambios del tipo gráfico, lo que hace que se estén modificando aspectos decorativos de los mismos, por lo que no sufre el

lenguaje base grandes cambios. Al contrario ocurre con modificaciones en estereotipos restrictivos, los cuales producen cambios del tipo semánticos en los lenguajes bases y por consiguientes en cada uno de sus subconjuntos contenidos en él. Es válido señalar por último que “Los estereotipos son rasgos o características muy potentes. Permiten un rango de modificaciones desde ligeros cambios notacionales hasta redefiniciones del lenguaje base. Sin embargo, el poder de los estereotipos supone riesgos. Estereotipos mal diseñados pueden dañar un lenguaje de modelación.” (Berner, 2001)

### **1.3.3 Las notaciones en la producción de aplicaciones educativas en Cuba**

Hasta ahora este trabajo se realiza manualmente, esto implica comenzar a desarrollar cada software y realizar su implementación a partir de cero a veces sin el conocimiento necesario para ello, lo que además de consumir grandes recursos humanos y de tiempo, afecta la calidad de los productos que se realizan. (Lorente Rodríguez, 2006)

Estando de acuerdo con lo que plantea el especialista citado, cabe mencionar que hoy en Cuba se utilizan para la modelación de las aplicaciones lo siguiente:

- Guiones de contenido y técnicos.
- Árboles o mapas de navegación.
- Análisis de posibles arquitecturas para los productos software, máxime cuando estos son colecciones.

Recordando las seis características distintivas del software educativo cubano, mencionadas en la introducción de este trabajo, es válido señalar que hoy en el país no llega a concretarse con eficiencia la documentación de este tipo de aplicaciones que posibilite lo siguiente:

- Durante la ejecución de los proyectos productivos:
  1. Entendimiento sin ambigüedades (utilización de iguales términos y conceptos semánticos) de los diferentes roles (especialistas pedagógicos, diseñadores gráficos, analistas, programadores y arquitectos) participantes en la elaboración del producto.
  2. Generación de una documentación técnica que posibilite la programación orientada a objetos de las aplicaciones de forma directa, que incorpore las mejores técnicas y prácticas internacionales en la industria del software.
- Al finalizar los mismos y en su mantenimiento:
  1. Utilización de la documentación del proyecto para generar pruebas de unidad y sistema a las aplicaciones, basadas en los algoritmos y flujos de funcionamiento del software educativo.

2. Utilización de la documentación para el mantenimiento de las aplicaciones y la posibilidad de eliminación de errores a partir de los modelos que del mismo se dispongan.
- Generación de nuevas versiones o portabilidad a otros entornos:
    1. Utilización de la documentación ingenieril del producto para el análisis de nuevas soluciones a partir de modelos de diseños horizontales en el desarrollo, que permitan general a partir de él nuevas versiones en otros lenguajes, plataformas o arquitecturas de solución; determinando la razón costo/beneficio de un cambio de este tipo sin incurrir en grandes gastos económicos.
    2. Utilización de la documentación para la determinación de posibilidades de migración a nuevas plataformas o sistemas operativos, determinando la razón costo/beneficio de un cambio de este tipo sin incurrir en grandes gastos económicos o necesidad de hacerlo in situ en laboratorios de prueba, sino desde la concepción ingenieril basada en los modelos existentes.

Las dificultades mencionadas anteriormente, sustentan la necesidad de un lenguaje notacional que permita adecuar los mejores lenguajes existentes en este ámbito al contexto productivo cubano y trabajar sobre la representación de los siguientes elementos:

1. Dominio de información del software.
2. Requerimientos pedagógicos asociados al software.
3. Características de las interfaces de comunicación con el usuario, punto neurálgico de este tipo de aplicaciones.
4. Conjunto de clases y entidades que sustentan la lógica de funcionamiento de las aplicaciones y sus relaciones.
5. Navegabilidad dentro de las aplicaciones.
6. Comportamiento de los elementos componentes del software.
7. Arquitectura que sostiene el modelo de desarrollo de las soluciones.
8. Posibilidad de división de las soluciones en unidades de trabajo más pequeñas desde el punto de vista funcional y visual.
9. Composición física de la solución generada.
10. Despliegue físico de la solución construida.

### **1.4 La documentación en el software educativo**

El término Documentación se refiere al texto escrito que acompaña al software generado en un proyecto. Es el conjunto de documentos, plantillas, esquemas, gráficos, especificaciones que detallan, describen y representan cada una de las partes que integran un producto de software, sus

componentes, dependencias y relaciones, así como lo descrito en todo el proceso de desarrollo y la etapa posterior a la culminación, en conjunto con la documentación para el usuario. La documentación debe ser breve, concisa y fácil de entender.

La documentación de los productos educativos que se utilizan, son en gran medida de difícil entendimiento, lo que requiere obtener un conjunto de documentación, plantillas o modelos que esclarezcan cómo implementar, mantener o darle soporte a una aplicación multimedia. Centrando el estudio en aras de contar con la misma y poder llegar a resultados eficientes en el menor tiempo posible. Además, a lo largo de todo su ciclo de vida, un producto siempre debe ir acompañado de documentación técnica de calidad; empezando por la documentación del proceso de desarrollo, pasando por la información y manuales de producto.

La puesta en funcionamiento de cualquier producto debe ser fácil y comprensible para el ingeniero de desarrollo u otro especialista que en algún momento lleve a cabo la tarea de mantener y mejorar funcionalidades del sistema, así como el usuario final. Para alcanzar esta meta son imprescindibles una correcta investigación, análisis y estructuración de la información. La cual debe estar organizada por niveles o módulos, categorías, paquetes o subsistemas que faciliten su búsqueda o localización y posteriormente una actualización del contenido ya documentado.

### **1.4.1 Objetivos de la documentación**

El objetivo de la documentación es simplificar la incorporación de nuevo personal (debe ser la herramienta empleada para ponerse al día rápidamente) y que sirva para que la gente que ya trabajaba en el proyecto tenga acceso a la historia de eventos importantes en el proyecto (cambios de requerimientos, decisiones de diseño, etc.).

La creación de productos de software sin el desarrollo de ingeniería se torna muy costosa e ineficiente si no se hace uso de sus potencialidades y las ventajas que ofrece. La principal tarea para lograr aplicaciones con calidad ha sido preparar personal calificado profesionalmente con conocimientos necesarios para la actividad de crear y de documentar lo creado usando modelos, la aplicación de teorías, técnicas de desarrollo que respalden y agilicen el proceso de producción.

### **1.4.2 Tipos de documentación**

En función del público objetivo y el tipo de contenido podemos distinguir varios tipos de documentación:

- Documentación de arquitectura y diseño: Son documentos que proporcionan una visión general de cómo se va a desarrollar el proyecto y de por qué se va a hacer de ese modo; estos documentos se suelen generar en las fases iniciales y deben ser revisados cuando se producen cambios. La idea básica es disponer de una descripción de alto nivel del sistema donde se enumeran los componentes que se van a emplear, la justificación de su elección, la funcionalidad esperada y las relaciones entre ellos. En los documentos de diseño se pueden incluir notas sobre implementación (tipos de datos, algoritmos, etc.), aunque la mayor parte de ellas se pueden dejar para la documentación técnica.
- Documentación técnica: Documentación del código, algoritmo, interfaces, etc. Es más detallada y debe ser escrita mientras se implementa; usualmente lo más cómodo es mantenerla junto al código fuente, al menos la que está relacionada con la interfaz y el programador.

Existen dos aproximaciones principales para mantener el código y su documentación de modo conjunto: escribir el código usando un formato analizable por un generador de documentación que funcione con múltiples lenguajes o que esté ligado a un lenguaje específico; estas herramientas analizan el código fuente y generan documentos en uno o varios formatos de salida (HTML, pdf, etc.) a partir de las construcciones del lenguaje de programación y de comentarios marcados de modo especial.

Usar el modelo conocido como programación literaria, una técnica propuesta inicialmente por Donald Knuth con su sistema WEB (la versión actual es CWEB, una reescritura de WEB para documentar programas en C, C++ y Java). En este sistema el código fuente está incluido en el texto descriptivo en lugar de al contrario; la idea básica es considerar un programa como un texto para los seres humanos en lugar de un conjunto de instrucciones para un ordenador. Para generar código compilable, se extraen los fragmentos de programa del documento empleando herramientas especializadas.

Además de la documentación del código, siempre hay documentos que no se deben redactar junto a éste, como por ejemplo las directrices para los desarrolladores del proyecto, que pueden incluir documentos que expliquen como usar el sistema de control de versiones, como se deben organizar, comentar y documentar el código.

- Documentación de usuario final: Es la documentación que se entrega al usuario final, tanto usuarios avanzados (administradores del sistema, personal de soporte) como no especializados.

Este tipo de documentación no suele tener relación con el código fuente, solo describe como usar los programas producidos en el proyecto, por lo que puede ser redactado por personas que no hayan estado involucradas en el desarrollo de los mismos.

### **1.4.3 Importancia de la documentación del software educativo**

Existen multitud de razones válidas para escribir documentación en un proyecto de software:

1. Es un requisito en un proyecto comercial, donde el cliente solicita que se documenten distintos componentes o fases del proyecto.
2. Se emplea para definir un modelo contractual que especifica como interactúan dos subsistemas diferentes; este tipo de modelos suele ser necesario cuando hay varios grupos trabajando en distintos componentes de un mismo sistema.
3. Si se une este modelo la documentación debe mantenerse al día para que los cambios queden reflejados y se pueda validar de qué modo deben interactuar los subsistemas en cada momento.
4. Para permitir la comunicación con equipos externos; cuando se trabaja con personas geográficamente dispersas debe existir un medio para comunicarlas y la documentación compartida es en muchas ocasiones parte de la solución, pero siempre combinada con otros canales como discusiones cara a cara, teleconferencias, correo electrónico o el uso de herramientas colaborativas.
5. Para pensar algo en profundidad. Mucha gente escribe documentación para aclarar cosas discutidas en grupo o aumentar su comprensión de un problema. El escribir ideas en papel puede ayudar a consolidarlas y descubrir posibles pegos en los razonamientos.

De hecho en el modelo de desarrollo tradicional la documentación es muy importante en todas las fases del ciclo de vida del software:

1. Ayuda a definir los problemas que se intentan resolver y a llegar a acuerdos con las personas que desean que estos sean resueltos.
2. Los requisitos deber escribirse en algún sitio, dado que ese es el único modo de validarlos y tener una referencia cuando se desarrolla.
3. Los codificadores, testadores y grupos externos necesitan la documentación producida en la fase de análisis, diseño y desarrollo, ya que es la base de su trabajo. La práctica demuestra que generalmente es necesario revisar y refinar la documentación a medida que el proyecto evoluciona, ya que los requisitos y expectativas cambian durante la vida del proyecto, forzando un modo de trabajo iterativo en el que se repiten fases anteriores para adaptarse a los cambios.

4. Las personas que trabajan en el mantenimiento de un producto necesitan de una buena comprensión de cómo se ha diseñado e implementado un sistema y la documentación producida en las fases anteriores deberían ser la fuente de información principal para obtenerla. Además, dependiendo del tipo de proyecto, los responsables de mantenimiento también necesitan documentar los cambios que se realizan y las tareas que hacen o necesitan hacer de modo regular.

### **1.5 El entorno tecnológico y científico en la representación del software educativo**

A lo largo de las últimas dos décadas se han desarrollado en la industria mundial y nacional un conjunto de soluciones informáticas para el tratamiento de entornos educativos modelados en aplicaciones, pero que no permiten la representación eficiente de todos los elementos que desde el punto de vista pedagógico son abordados en el software educativo cubano. Se dará en este epígrafe una descripción lo más breve posible de cada uno de estas soluciones para introducir en lo actual, por el interés futuro en establecer comparaciones con lo que nos ocupa, el Lenguaje de Modelado para Aplicaciones Educativas.

“Desde alrededor de 1990, una amplia variedad de lenguajes de modelación han sido desarrollados. Estos lenguajes son utilizados para describir los requerimientos y el diseño de un sistema software. Desde 1996, varias tentativas han sido ejecutadas para unificar diferentes métodos y lenguajes. Como resultado de este esfuerzo, un lenguaje fue desarrollado: el Lenguaje Unificado de Modelado (UML). UML introduce una nueva característica distintiva: permite a los usuarios extender o incluso modificar el lenguaje base para adaptar el lenguaje a situaciones o necesidades específicas. La construcción del lenguaje que es utilizada para implementar esta característica es llamada: estereotipo.” (Berner, 2001)

#### **1.5.1 RMM: Metodología de Administración de Relaciones**

La metodología RMM, fue desarrollada por T. Isakowitz, en la Universidad de Nueva York en el año 1995. Se realiza la modelación de las aplicaciones a través de RMDM (*Relationship Management Data Model*), basado en el modelo Entidad – Relación y posee una herramienta CASE denominada: *Relationship Management Case Tool* – RMCASE. “RMM (*Relationship Management Methodology*) contiene el diseño y la construcción de aplicaciones hipermedia en un proceso de siete pasos. Es al mismo tiempo un enfoque “top down” y “bottom up”. Durante la fase del diseño Entidad – Relación, entidades y relaciones son identificadas las cuales se convertirán en nodos y enlaces en la hipermedia resultante. El segundo paso, diseño de cortes (slices), involucra el agrupamiento de atributos de



entidades para la presentación. Los cortes (slices) son “unidades de presentación” que aparecen como páginas de una aplicación hipermedia. La separación del contenido y los aspectos de la presentación no son satisfechos en este paso. RMM especifica la navegación con primitivas de acceso, como enlaces (links), agrupamiento (menus), índices (index) y recorridos guiados (guided tours). La técnica propuesta para el diseño de la interfaz de usuario es la elaboración de maquetas y prototipos.” (Baumeistier, 2001)

### **1.5.2 OOHDM: Metodología de Diseño Hipermedia Orientada a Objetos**

Desarrollada por Schwabe y Rossi, en la Universidad de Rio de Janeiro, Brasil y Universidad Nacional de la Plata, Buenos Aires respectivamente en el año 1996. Adopta la notación y los mecanismos de abstracción de la Programación Orientada a Objetos (POO) y consta de cuatro pasos para su ejecución: diseño conceptual, diseño navegacional, diseño de interfaz abstracta e implementación. Trabaja la representación a través de los siguientes modelos: Esquema de clases, esquema de navegación, esquema contextual de navegación y vista abstracta de datos. “El Modelo de Diseño de Hipermedias Orientado a Objetos: OOHDM (*Object – Oriented Hypermedia Design Model*) comprende cuatro actividades; estas son modelo conceptual, diseño de navegación, diseño de interfaces abstractas e implementación. Estas actividades son ejecutadas en un estilo de desarrollo mixto a partir de los modelos incremental, iterativo y basado en prototipos. Este método trata a la aplicación como una vista superior al modelo conceptual. El concepto de contexto de navegación es introducido para describir la estructura de navegación. Es un concepto potente que permite diferentes agrupamientos de objetos de navegación con el propósito de navegar en ellos en diferentes contextos. Una notación especial es utilizada para la representación de la estructura de navegación. En trabajos tempranos de investigación, OMT se propuso como la notación para el esquema conceptual; un poco más tarde los trabajos ya utilizan UML. Sin embargo, los diagramas de OOHDM no obedecen los patrones UML, sino que utilizan una notación propia para la perspectiva de los atributos en los diagramas de clase y proponen otros tipos de diagramas para el diseño de la navegación y de las interfaces de usuarios abstractas.” (Baumeistier, 2001)

### **1.5.3 UML: Lenguaje Unificado de Modelado**

El Lenguaje Unificado de Modelado (UML) fue desarrollado por Ivar Jacobson, Grady Booch y James Rumbaugh, en la Corporación “*Rational Software*”, en el año 1996 en su primera versión. Mejorado en la actualidad a una versión superior 2.0 y desde sus inicios respetó los elementos estándares del lenguaje OCL, desarrollado inicialmente por la IBM. Consta de varias áreas conceptuales como son: estructura estática, comportamiento dinámico, construcciones de implementación, organización del

modelo y los mecanismos de extensión. A su vez, consta de ocho vistas: estática, de casos de uso, de implementación, de despliegue, de máquinas de estado, de actividad, de interacción y de gestión del modelo; para la modelación de los productos, a través de un conjunto de diagramas distribuidos por cada una de estas vistas. Tiene como base teórica lo siguiente: Unified Method (Grady Booch, 1994), OMT (Grady Booch y James Rumbaugh, 1995) así como el Método OOSE (Ivar Jacobson, 1996). Posee varias herramientas CASE de modelado como son Rational Rose Enterprise Edition y Visual Paradigm. Actualmente en un lenguaje estandarizado por la OMG a partir del año 1997 y ha servido como base para posteriores desarrollos de lenguajes a partir de sus posibilidades de extensión. (Booch, 2000)

#### **1.5.4 OMMMA – L: Lenguaje para la modelación Orientada a Objetos de Aplicaciones**

Fue desarrollado por Stefan Sauer y Gregor Engels, en la Universidad de Paderborn, Alemania, en el año 2001, tomando como base el lenguaje UML. Consta de cuatro vistas fundamentales en su modelación: vista lógica, vista de presentación espacial, vista de comportamiento temporal predefinido y vista de control interactivo. Modifica los diagramas originales de UML de: clases, secuencia y estado. Añade como parte de la vista de presentación espacial un nuevo diagrama: el diagrama de presentación, para la representación espacial de los elementos visuales del futuro software multimedia. Basa su descripción en el patrón de arquitectura MVCMM. (Sauer, 2001)

#### **1.6 Lenguaje de Modelado para Aplicaciones Educativas (ApEM-L)**

Sabiendo que el Leguaje Unificado de Modelado no soporta con sus vistas y diagramas todos los aspectos que son necesarios modelar en el software educativo cubano, se hace necesario introducir un conjunto de estereotipos descriptivos y restrictivos para la extensión de la semántica de los establecido por el lenguaje base y de esta forma permitir incorporar los conceptos manejados en el contexto del software educativo. También, que se respeta el lenguaje base UML, así como lo establecido por el lenguaje OCL como estándar en la modelación orientada a objetos, se creó el ApEM-L. El cual incorpora la vista de presentación como aporte fundamental a la modelación de este tipo de aplicaciones con dos diagramas que permiten definir la estructura para la navegación y la estructura para la presentación.

ApEM – L se presenta como una extensión de UML, tomando como bases teóricas principales OMMMA – L (2001) y OCL – 2.0 (2003), lo que produce las siguientes ventajas: (Ciudad Ricardo, 2007)

- Puede utilizar para su representación todas las herramientas CASE que existen actualmente para la modelación de UML.
- Es un lenguaje que utiliza el estándar internacional OCL, para la modelación de la Programación Orientada a Objetos.
- No modifica la semántica del lenguaje base UML, sino que trabaja en estereotipos restrictivos, por lo que a su vez produce modificaciones descriptivas y decorativas en la representación de los componentes del lenguaje base.

## **1.6.1 Objetivos de ApEM-L**

La concepción del Lenguaje para la Modelación de Aplicaciones Educativas tuvo los siguientes objetivos:

1. Desarrollar una extensión del lenguaje de modelado UML, tomándolo como base e incorporando a éste, a través de sus mecanismos de extensión, los elementos fundamentales del proceso productivo UCI. De esta forma se produjo un lenguaje de propósito particular para la modelación de aplicaciones educativas.
2. Incorporar los elementos más significativos de extensiones anteriores como OMMMA – L (2001) y a su vez respetar lo establecido por el estándar OCL (2003), para de esta forma lograr una extensión consistente y escalable en el tiempo.
3. No complicar ApEM – L con elementos que lo convirtieran o abarcaran un método de desarrollo de aplicaciones educativas, sino solo el área de la representación y la documentación de este tipo de aplicaciones.
4. No circunscribir ApEM – L a un proceso de desarrollo en específico, sino expresarlo de manera tal que pueda ser utilizado con cualquiera de los existentes, aunque se sugiere la utilización de procesos de desarrollo (iterativos, incrementales y basados en prototipos) que permitan la modelación de sistemas orientados a objetos.

Tanto ApEM–L como UML, no establecen ninguna línea entre los diferentes conceptos y construcciones del lenguaje, pero por conveniencia del primero, éste se ha dividido en varias vistas, modelando cada una de estas construcciones que representan un aspecto del sistema. La división ha sido sobre la base de las áreas conceptuales, que son: Estructura Lógica, Comportamiento Dinámico y Gestión del Modelo. En la primera conceptual están la Vista Estática que está compuesta por dos diagramas, el Diagrama de Clases y el Diagrama de Casos de Uso y la Vista de Arquitectura que está compuesta por el Diagrama de Componentes y el Diagrama de Despliegue. En la segunda área está la Vista de Comportamiento la cual está compuesta por cuatro Diagramas: de Actividades, de Estado, de

Secuencia y de Colaboración. En la otra área está la Vista de Presentación con el Diagrama de Estructura de Navegación y el Diagrama de Estructura de Presentación.

### ***Conclusiones parciales***

En este capítulo se han ofrecido los elementos teóricos que sirven de sustento científico a la investigación, así como conceptos que se deben dominar. El análisis de cada uno de estos aspectos mencionados permite arribar a las siguientes conclusiones:

- La modelación de los elementos de la ingeniería son necesarios para una correcta implementación, mantenimiento, reingeniería, reutilización de código y portabilidad de los sistemas en desarrollo.
- Existen un conjunto de notaciones y lenguajes para el desarrollo de aplicaciones pero no permiten la representación eficiente de todos los elementos que desde el punto de vista pedagógico son abordados en el software educativo cubano.
- El Lenguaje de Modelado para Aplicaciones Educativas representa un paso de avance en el modelamiento en el proceso de producción de software educativo.

## CAPÍTULO 2: UN ACERCAMIENTO A LA UTILIZACIÓN DE APEM-L 1.0 EN EL MINISTERIO DE EDUCACIÓN

El Departamento Nacional de Software Educativo del Ministerio de Educación que forma parte de la Dirección de Computación Educativa tiene dentro de sus funciones dirigir, coordinar, organizar y controlar la producción e introducción en la práctica de software educativo para la escuela cubana, desde el ámbito de la red de Centros de Estudio de Software Educativo de los Institutos Superiores Pedagógicos, bajo un esquema de investigación-producción. Con el fin de implementar todas estas funciones se han creado proyectos más avanzados de software en los Institutos Superiores Pedagógicos con personal calificado como Másteres, Doctores en Ciencias y Licenciados junto a programadores y técnicos en Ciencias de la Informática, entre otros especialistas, con el objetivo de satisfacer las necesidades de los software educativos.

En los últimos años se han desarrollado gran cantidad de productos y otros que están en el proceso de creación. Actualmente las escuelas cuentan con una diversidad de software que los alumnos y maestros utilizan de forma satisfactoria en su preparación. Por lo cual se puede afirmar que se ha obtenido cierta experiencia en la producción de software educativo.

### ***2.1 La modelación antes de la introducción de ApEM-L***

En el Ministerio de Educación existe un gran interés por el desarrollo de multimedia educativas, lo que se ha visto reflejado en el incremento de la producción en estos últimos años. La modelación del software es algo muy útil y de implementarse correctamente trae grandes beneficios; pero hasta ahora, en el MINED, este trabajo se realiza manualmente y en algunos casos ni se realiza, esto implica comenzar a desarrollar cada software y realizar su implementación sin apoyo e independientemente a otros desarrollados anteriormente. Se puede observar además que no emplean muchos recursos ni tiempo en este proceso tan importante, que les podría ayudar a economizar tiempo y esfuerzos en cada uno de los nuevos proyectos que se tracen.

En los proyectos donde se realiza modelamiento en el proceso de producción de las aplicaciones educativas, se utilizan los artefactos siguientes:

- Guiones de contenido y técnicos.
- Árboles o mapas de navegación.

Estos no son lo suficientemente eficientes y conllevan en muchos casos a errores de razonamiento de funcionalidad, ambigüedades de varias categorías, y con ellos solamente no se puede tener una visión exacta del sistema, darle mantenimiento, o chequear de forma más detallada, si no que provoca muchos problemas de entendimiento por parte de guionistas, gestores de media, diseño y programadores. No permite en un futuro reutilizar código o funcionalidades ya implementadas que agilicen este proceso, ni se logra definir una arquitectura que sustente las bases de productos de este tipo para la integración de todas sus partes, así como llegar a tener en cierto momento un patrón que posibilite estimar o calcular los costos de ciertas funciones o servicios.

Sería de gran ayuda la introducción de algunos métodos más eficientes, así como algún lenguaje de modelado que ayude a explicar, representar y en la comunicación de los diferentes roles del proceso de producción. Es por ello que se ha propuesto la utilización de ApEM-L, dado que representa una herramienta muy útil e importante.

### ***2.2 Importancia de la utilización de ApEM-L 1.0 en el MINED***

El lenguaje que se propone a utilizar, ApEM-L, es presentado como una extensión de UML, tomando como bases teóricas principales OMMMA – L (2001) y OCL – 2.0 (2003). Dicho lenguaje facilita el modelado y la interacción entre los diferentes roles que están involucrados en el proceso de producción, ya que trabaja en profundidad los aspectos relacionados con las medias, añadiendo especificaciones de las mismas en las descripciones en los casos de uso y los aspectos pedagógicos establecidos, así como cada una de las vistas para representar el software desde diferentes perspectivas.

Por las características mencionadas en el epígrafe anterior de cómo se desarrolla el software educativo en el Ministerio de Educación y que el Leguaje Unificado de Modelado no soporta con sus vistas, diagramas y todos los aspectos que son necesarios para modelar este tipo de software, se puede visualizar la importancia del uso de ApEM-L en las modificaciones que fueron propuestas en el momento de creación de la notación y en las múltiples ventajas que presenta, todo esto expuesto anteriormente. También, los objetivos trazados en la concepción de dicho lenguaje por su creador el Msc. Febe Ángel Ciudad Ricardo, dejan ver con claridad el avance que representa la utilización del mismo.

### **2.3 Factores que dificultan la introducción de ApEM-L 1.0 en los proyectos productivos**

De la entrevista realizada a la vicedecana de producción de la Facultad 8: Licet Gutierrez Mompié, se obtuvo que el principal factor que influye negativamente en la introducción de ApEM-L en los proyectos productivos de software educativos es el desconocimiento del mismo, producto a lo novedoso que es este lenguaje de modelado.

De las entrevistas y encuestas realizadas al líder del proyecto “A jugar”: Diony Suares Yanes, se obtuvo los siguientes factores que influyen negativamente en la introducción de ApEM-L en este proyecto productivo:

1. ApEM-L es algo novedoso.
2. Falta de personal calificado para impartir la capacitación, lo que es muy necesario.
3. Falta de experiencia para la aplicación del mismo.
4. La aplicación del lenguaje trae consigo algunos cambios que influyen en todos los aspectos del proceso de producción que tiene el proyecto.

De las entrevistas y encuestas realizadas al líder del proyecto “Multisaber”: Liana Isabel Araujo Pérez, se obtuvo los siguientes factores que influyen negativamente en la introducción de ApEM-L en este proyecto productivo:

1. Falta de personal capacitado para impartir la capacitación.
2. Falta de experiencia para la aplicación del mismo.

De las entrevistas y encuestas realizadas al líder del proyecto “Español para no hispanohablantes”: Yuneiry Barroso Pedroso, se obtuvo los siguientes factores que influyen negativamente en la introducción de ApEM-L en este proyecto productivo:

1. ApEM-L es algo novedoso.
2. La aplicación del lenguaje trae consigo algunos cambios que influyen en todos los aspectos del proceso de producción que tiene el proyecto.
3. Descontento con la Introducción de ApEM-L.
4. Falta de experiencia para la aplicación del mismo.

Aunque se logró identificar estos factores, la repuesta principal del compañero fue que en este año por los planes que tenía y el atraso que le representaba empezar a utilizar ApEM-L no lo introduciría en el proyecto.

De la encuesta realizada al Msc. Febe Ángel Ciudad Ricardo, se obtuvo los siguientes factores que influyen negativamente en la introducción de ApEM-L en los proyectos productivos:

1. Falta de personal calificado para impartir la capacitación, lo que es muy necesario.
2. Falta de interés por la capacitación para la aplicación de ApEM-L.
3. No disponibilidad de los medios y locales para la capacitación.
4. Falta de experiencia para la aplicación del mismo.
5. La aplicación del lenguaje trae consigo algunos cambios que influyen en todos los aspectos del proceso de producción que tienen cada uno de los proyectos.
6. Desconocimiento de elementos generales de la Ingeniería de Software.

Concluyendo, muchos de los problemas que tienen que ver con la introducción de ApEM-L 1.0 en los proyectos productivos del Ministerio de Educación están relacionados con factores como: la falta de personal capacitado para impartir los cursos de adiestramiento con dicho lenguaje, dado esto, gran porcentaje de los entrevistados consideran que la capacitación es insuficiente. Además, la falta de interés por la capacitación de los integrantes de los equipos de desarrollo, así como la no disponibilidad de los medios y locales para ello. Por otro parte, algo que influye fuertemente y es una de las cosas principales a tomar en cuenta es que ApEM-L es muy novedoso todavía, puesto que fue concebido hace solo un año, lo que trae consigo desconocimiento para la utilización del mismo y además que exista solo un pequeño grupo de proyectos productivos aplicándolo actualmente. También, la aplicación del lenguaje conlleva a muchos cambios que influyen en todos los aspectos del proceso de producción que tienen cada uno de los proyectos productivos, generando esto descontento y desactualización de los últimos. Otro factor, de igual manera influyente, es la inexperiencia en el modelamiento de estas aplicaciones por parte de los integrantes de los proyectos y la falta de conocimientos de los elementos generales de la Ingeniería de Software.

### **2.4 ApEM-L 1.0 en el Ministerio de Educación**

Para conocer el impacto que ha tenido ApEM-L 1.0 en el Ministerio de Educación se realizó una encuesta, la cual fue aplicada a los integrantes de los proyectos “A jugar” y “Multisaber” que están utilizando y se han familiarizado con este lenguaje de modelado. Para emitir una valoración se realizó



un análisis estadístico sencillo por cada una de las preguntas teniendo en cuenta el criterio de cada encuestado.

**Pregunta 1:** ¿Considera usted ventajoso la aplicación de una extensión como ApEM – L para la modelación de aplicaciones educativas?

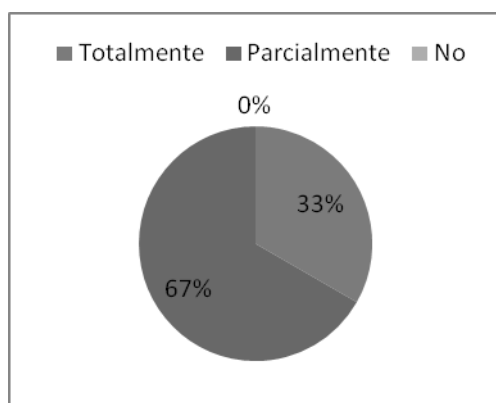


Figura 1.

**Valoración:** A pesar de que hay personas que opinan que la aplicación de una extensión como ApEM-L es totalmente ventajoso, existe un grupo mayor que considera que es ventajoso pero solo parcialmente.

**Pregunta 2:** ¿Se siente usted cómodo al trabajar con la notación ApEM – L en sus distintas vistas?

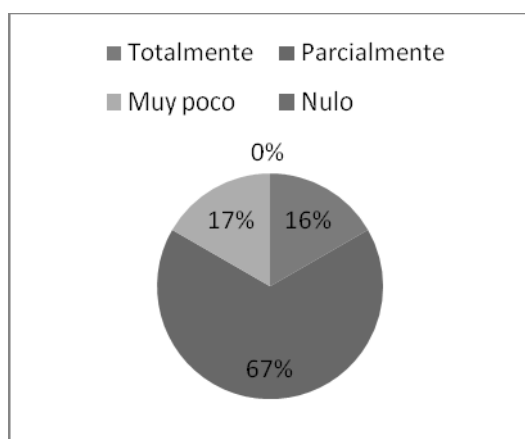
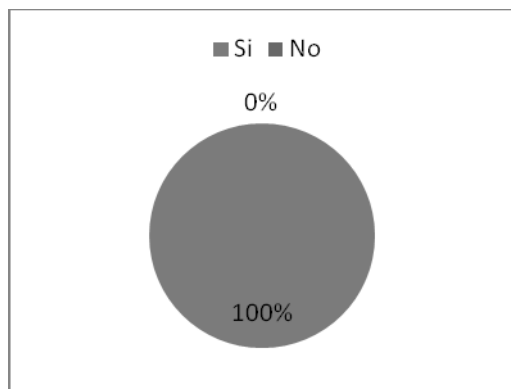


Figura 2.

**Valoración:** Las respuestas varían desde totalmente hasta muy poco, pero el mayor número se conjuga al opinar que es de forma parcial que el trabajo con la notación ApEM-L en sus distintas vistas, les brinda comodidad.

**Pregunta 3:** ¿Considera usted que es sencilla, fácil de entender y aplicar la notación con unos pocos conocimientos de programación?



**Figura 3.**

**Valoración:** No es necesario contar con muchos conocimientos de programación para aplicar la notación y además, todos consideran que es sencilla y fácil de entender.

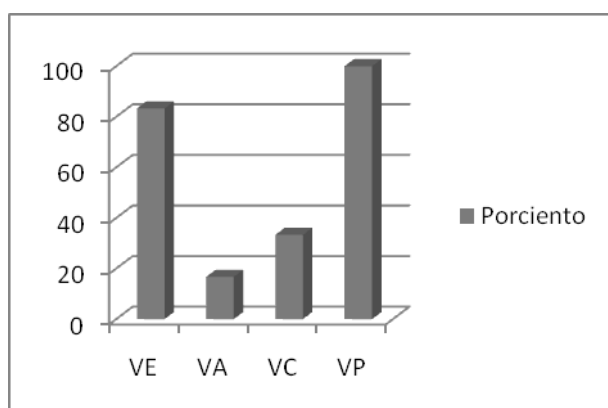
**Pregunta 4:** Marque con una X las vistas y los diagramas que considera mejor trabajados en ApEM – L para la modelación de aplicaciones educativas.

VE: Vista Estática

VC: Vista de Comportamiento

VA: Vista de Arquitectura

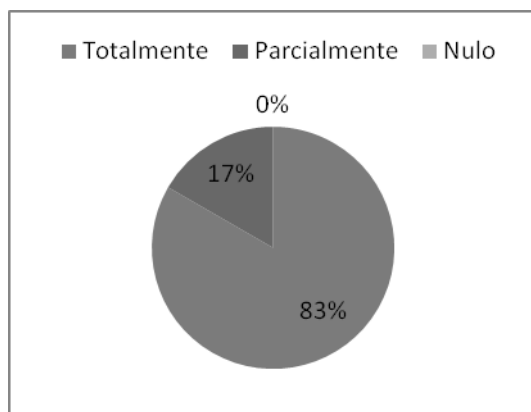
VP: Vista de Presentación



**Figura 4.**

**Valoración:** Según el alto porcentaje de criterios se definen como las vistas y diagramas mejor trabajados a la Vista de Presentación y la Vista Estática. La Vista de Arquitectura y la Vista de comportamiento han sido trabajadas en menor medida.

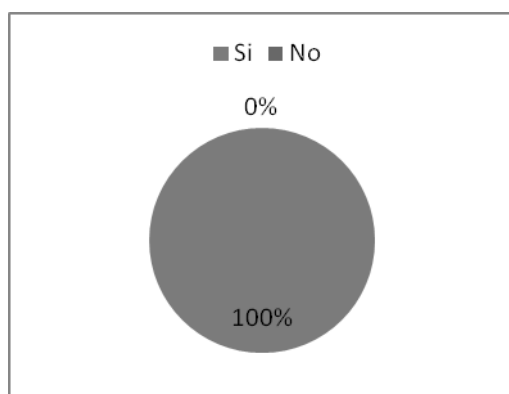
**Pregunta 5:** ¿Cree eficiente la utilización de las descripciones textuales de los casos de uso al describir los flujos de eventos de un software educativo?



**Figura 5.**

**Valoración:** Existe un pequeño grupo que considera esto parcialmente, pero la gran mayoría opina que, la utilización de las descripciones textuales de los casos de uso al describir los flujos de eventos dentro de un software educativo, es totalmente eficiente.

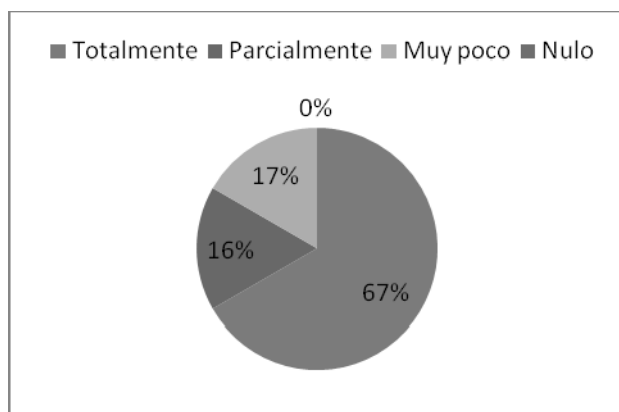
**Pregunta 7:** ¿Cree usted que debe desarrollarse un programa que permita la generación computacional de la documentación resultante de la modelación?



**Figura 6.**

**Valoración:** Según el criterio de los encuestados, consideran en su totalidad que debe desarrollarse un programa que permita la generación computacional de la documentación resultante de la modelación, para así facilitar el trabajo en aras de economizar tiempo y número de personas.

**Pregunta 8:** ¿Considera que el uso de ApEM – L sustituye la necesidad en la implementación del uso del Guión y el Mapa de Navegación?



**Figura 7.**

**Valoración:** Aunque existen opiniones que no expresan lo mismo que la mayoría, se considera que el uso de ApEM-L sustituye la necesidad del uso del guión y el mapa de navegación.

**Pregunta 10:** ¿Durante la utilización de ApEM-L se ha percatado que sería ventajoso modificar o añadir cierto(s) elemento(s)?

**Valoración:** Debido al poco tiempo de interacción con este lenguaje de modelado se han identificado muy pocos elementos que no están representados, así como otros que haya que modificar.

**Valoración general:** Es ventajoso aplicar una extensión como ApEM-L para modelación de aplicaciones educativas. Se considera que la notación es sencilla, fácil de entender y aplicar con pocos conocimientos de programación y trae consigo cierta comodidad al trabajar en sus diferentes vistas. De éstas últimas las que se consideran mejor trabajadas en ApEM-L son la Vista de Presentación y la Vista Estática con sus diagramas correspondientes. Además, son eficientes las descripciones textuales de los casos de uso al describir los flujos de eventos de un software educativo. Se está de acuerdo con la idea de desarrollar un programa que permita la generación computacional de la documentación resultante de la modelación. El uso del guión y mapa de navegación ya no es tan necesario y es sustituido por el uso de ApEM-L.

#### **2.4.1 Elementos de ApEM-L 1.0 a modificar según los proyectos productivos del MINED**

Del resultado de las encuestas se identificaron los siguientes elementos:

1. El área conceptual que aparece.
2. El diagrama de casos de uso.
3. La disminución de las responsabilidades.

### **2.4.2 Elementos no representados por ApEM-L 1.0 en los proyectos productivos del MINED**

Del resultado de las encuestas se identificaron los siguientes elementos:

1. Faltan estereotipos restrictivos a los diagramas de estructura de presentación y navegación.
2. No existen vistas de presentación, solo describen los casos de uso.
3. No existen relaciones entre las vistas.

### **2.5 Nivel de aplicación de ApEM-L 1.0 en los proyectos productivos del MINED**

Para determinar el nivel de aplicación se ha tenido en cuenta un conjunto de criterios de estrecha relación con la utilización del ApEM-L. Analizando primeramente que se han identificado muy pocos elementos a modificar y no representados, lo que da una medida de la poca interacción con la notación. Por otra parte, se tiene que no se utilizan todos los estereotipos existentes ni realizan todos los diagramas que pueden facilitar en gran medida el proceso de producción, así como la comunicación entre los diferentes roles. Además, las vistas no son abordadas en su totalidad aunque de alguna manera se trabaja en todas, pero todavía no se emplean todos los componentes de estas. También, al evaluar el uso del patrón arquitectónico Modelo – Vista – Controlador – Entidad (MVC-E) para la producción de software educativo se concluye que es poco aplicado, porque solo se han utilizado algunas de las sesiones del diagrama de clases establecido por el lenguaje. Entonces, teniendo en cuenta lo anterior y que se aplica en solo un pequeño grupo de proyectos y desde hace un pequeño período de tiempo, se define como “bajo” el nivel de aplicación de ApEM-L 1.0.

### **Conclusiones parciales**

En este capítulo se llevó a cabo un estudio de cómo realizan la modelación en la producción del software educativo en el Ministerio de Educación. Se analizaron las ventajas que tiene el ApEM-L para demostrar la importancia de la utilización del mismo. También se identificaron algunos factores que limitan la introducción en los proyectos del MINED, de dicho lenguaje de modelado. Luego se determinaron elementos a modificar y los no representados en la versión 1.0, para finalmente definir el nivel de aplicación de la notación en la empresa. Del análisis de cada uno de estos aspectos mencionados permite arribar a las siguientes conclusiones:

- En el MINED se trabaja las aplicaciones educativas a través de las tecnologías multimedia e hipermedia, utilizando un conjunto de lenguajes de alto nivel (HLL) o sistemas de autor diseñados para estos fines.

- No se emplea mucho tiempo ni esfuerzos en los proyectos productivos para el modelamiento de cualquier software que se desarrolle y donde se realiza, los métodos que emplean no son los más óptimos.
- Existe gran cantidad de factores limitando la introducción del lenguaje, lo que ha frenado este proceso, sumado a que el ApEM-L es casi desconocido y se ha aplicado en muy pocos proyectos productivos.
- Se han identificado muy pocos elementos a modificar y no representados y esto está dado por la falta de experiencia y el corto tiempo en el trabajo con ApEM-L.

### **CAPÍTULO 3: PROPUESTA DE APLICACIÓN DE APEM-L EN LOS PROYECTOS PRODUCTIVOS DE SOFTWARE EDUCATIVO DEL MINISTERIO DE EDUCACIÓN**

Para lograr una correcta inserción de ApEM-L en los proyectos productivos de software educativos del Ministerio de Educación se propone una estrategia a seguir:

1. ApEM-L tiene que ser del conocimiento de todos los integrantes de los equipos de desarrollo de software educativos, así como la importancia y las ventajas que trae para la modelación y representación de aplicaciones de este tipo.
2. Se define cuántas y quiénes son las personas indicadas para impartir la capacitación según la cantidad de individuos que serán beneficiados con el proceso.
3. Se seleccionan los trabajadores que tomarán parte en el proceso de introducción de la notación, teniendo en cuenta que posean ciertos conocimientos generales de la Ingeniería del Software y alguna experiencia en la modelación de aplicaciones educativas.
4. Se realiza un plan de capacitación y se publica para el conocimiento de todos los integrantes del curso para el aprendizaje del trabajo con la notación, el cual se controlará para que no ocurran sucesos no coordinados con anterioridad.
5. Se destina un conjunto de medios y locales para el proceso de familiarización con el lenguaje de modelado.
6. Crear lugar o repositorio donde exista la documentación y las actualizaciones del Lenguaje de Modelado para Aplicaciones Educativas.
7. Se elabora una nueva planificación de los proyectos incluidos en el proceso para que luego no existan problemas con las entregas y otros.

Para validar técnicamente la propuesta se utiliza el método de experto, que permite tomar decisiones para aceptar o no la propuesta de acuerdo con los criterios definidos.

Para llevar a cabo el desarrollo del mismo se efectuaron un conjunto de pasos:

1. Se elabora los criterios de evaluación de acuerdo a las características de la propuesta y se organizan por grupos.

Grupo No. 1: Criterios de mérito científico

- Valor científico de la estrategia
- Calidad de la investigación.
- Contribución científica.

Grupo No. 2: Criterios de implantación

- Satisfacción de las necesidades de los integrantes de los proyectos.
- Necesidad del empleo de la propuesta.
- Uso del ApEM-L.

Grupo No.3: Criterios de flexibilidad

- Adaptabilidad a proyectos productivos.
- Uso de las herramientas necesarias para la elaboración de los artefactos.

Grupo No.4: Criterios de impacto

- Repercusión en los proyectos productivos.
- Organización en el proceso de documentación del software.
- Posibilidades de aplicación.

2. Se le asigna un peso relativo a cada grupo de criterios de acuerdo al porcentaje que representa cada grupo del total y los intereses a evaluar.

Grupo No. 1..... 20

Grupo No. 2..... 35

Grupo No.3..... 15

Grupo No.4.....30

3. Se organiza un comité de expertos con una cantidad mínima de 7 teniendo en cuenta su especialidad, grado científico y currículum.

4. Se les entrega a los expertos la estrategia para que estudien el tema a evaluar y dos modelos, el **Modelo 1** para que valore el peso relativo de cada criterio y el **Modelo 2** para que exprese su valoración sobre los mismos en una escala de 1 a 5.

5. Después de recibir los valores del peso relativo de cada criterio se construye la Tabla No.1

**Tabla No.1 Resultado del trabajo de expertos**

G	C/E	E1	E2	E3	E4	E5	E6	E7	Ep
25	C1	9	8	8	9	7	8	8	8.14
	C2	8	8	7	8	8	7	7	7.57
	C3	8	9	9	7	9	9	9	8.57



30	C4	10	10	10	10	10	10	10	10
	C5	10	10	10	10	10	10	10	10
	C6	10	10	10	10	10	10	10	10
15	C7	8	8	7	8	8	8	7	7.71
	C8	7	7	8	7	7	7	8	7.29
30	C9	10	10	10	10	10	10	10	10
	C10	10	10	10	10	10	10	10	10
	C11	10	10	10	10	10	10	10	10
T		100	100	100	100	100	100	100	100

6. Se verifica la consistencia en el trabajo de los expertos, para lo que se utiliza el coeficiente de concordancia de Kendall y el estadígrafo Chi cuadrado ( $X^2$ ). Se sigue el procedimiento siguiente:

- Sea C el número de criterios que van a evaluarse y E el número de expertos que realizan la evaluación.
- Para cada criterio se determina:

$\sum E$ : Sumatoria del peso dado por cada experto

$E_p$ : Puntuación promedio del peso dado por cada experto

$M\sum E$ : media de los  $\sum E$

$\Delta C$ : Diferencia entre  $\sum E$  y  $M\sum E$

- Se determina la desviación de la media, que posteriormente se eleva al cuadrado para obtener la dispersión (S) por la expresión

$$S = \sum (\sum E - \sum \sum E / C)^2; \quad \underline{S = 665,88}$$

- Con la dispersión se puede calcular el coeficiente de concordancia de Kendall (W)

$$W = S / (E^2 (C^3 - C) / 12); \quad \underline{W = 0,1235}$$

- El coeficiente de concordancia de Kendall permite calcular el Chi cuadrado real

$$X^2_{\text{real}} = E (C-1) W; \quad \underline{X^2 = 8,65}$$

- Los valores obtenidos se muestran en la Tabla No.2.

**Tabla No.2 Tabla para el cálculo de concordancia de Kendall**

Expertos/Criterios	E <sub>1</sub>	E <sub>2</sub>	E <sub>3</sub>	E <sub>4</sub>	E <sub>5</sub>	E <sub>6</sub>	E <sub>7</sub>	ΣE	Ep	ΔC	ΔC <sup>2</sup>
C <sub>1</sub>	9	8	8	9	7	8	8	57	8.14	6.63	43.96
C <sub>2</sub>	8	8	7	8	8	7	7	53	7.57	10.63	113
C <sub>3</sub>	8	9	9	7	9	9	9	60	8.57	3.63	13.18
C <sub>4</sub>	10	10	10	10	10	10	10	70	10	6.37	40.58
C <sub>5</sub>	10	10	10	10	10	10	10	70	10	6.37	40.58
C <sub>6</sub>	10	10	10	10	10	10	10	70	10	6.37	40.58
C <sub>7</sub>	8	8	7	8	8	8	7	54	7.71	9.63	92.74
C <sub>8</sub>	7	7	8	7	7	7	8	51	7.29	12.63	159.52
C <sub>9</sub>	10	10	10	10	10	10	10	70	10	6.37	40.58
C <sub>10</sub>	10	10	10	10	10	10	10	70	10	6.37	40.58
C <sub>11</sub>	10	10	10	10	10	10	10	70	10	6.37	40.58
DC	100	100	100	100	100	100	100	700	100	81.37	665.88
M ΣE	63.63										
W	0.1235										
X <sup>2</sup> <sub>real</sub>	8.65										

- El Chi cuadrado calculado se compara con el obtenido del las tablas estadísticas para un alfa (coeficiente de confianza) que nos de cierto grado de veracidad; α = 0,5.

$$X^2_{(\alpha, c-1)} = X^2_{(0,5, 11-1)} = X^2_{(0,5, 10)} = 9,3418$$

- Si se cumple:

$$X^2_{real} < X^2_{(\alpha, c-1)} \quad \text{Existe concordancia en el trabajo de expertos}$$

Entonces:

Como 8,56 < 9,34.18            Existe concordancia en el trabajo de los expertos.

7. Después de comprobar la consistencia del trabajo de expertos se puede definir el peso relativo de cada criterio (P).
8. Conociendo el peso de cada criterio y la calificación dada por los evaluadores en una escala de 1-5 se puede construir la Tabla No.3, para obtener el valor de de  $P \times c$ ., donde (c), es el criterio promedio concebido por los expertos.

**Tabla No.3 Tabla de calificación de cada criterio**

Criterios	Calificación (c)					P	P × c
	1	2	3	4	5		
C <sub>1</sub>				X		0.0814	0.3256
C <sub>2</sub>				X		0.0757	0.3028
C <sub>3</sub>			X			0.0857	0.2571
C <sub>4</sub>			X			0.1	0.3
C <sub>5</sub>				X		0.1	0.4
C <sub>6</sub>				X		0.1	0.4
C <sub>7</sub>			X			0.0771	0.2313
C <sub>8</sub>			X			0.0729	0.21.87
C <sub>9</sub>				X		0.1	0.4
C <sub>10</sub>			X			0.1	0.3
C <sub>11</sub>				X		0.1	0.4

9. Se calcula el Índice de aceptación del proyecto (IA).

$$IA = \sum (P \times c) / 5; \quad \underline{IA = 0.7071}$$

10. Por último se determina la probabilidad de éxito de la propuesta

**Rangos predefinidos de Índice de Aceptación.**

- IA > 0,7 Existe alta probabilidad de éxito
- 0,7 > IA > 0,5 Existe probabilidad media de éxito
- 0,5 > IA > 0,3 Probabilidad de éxito baja

$0,3 > IA$  Fracaso seguro

Por lo que la probabilidad de éxito es: Alta.

Además, es establecida por los expertos una valoración final de: Aceptable.

### ***Conclusiones parciales***

En este capítulo se hace una propuesta de aplicación de una estrategia que logre un proceso satisfactorio en la introducción de ApEM-L en los proyectos productivos del Ministerio de Educación. Se concluye en que:

- La estrategia puede ser aplicada puesto que se declara aceptable según las valoraciones de los expertos, que además dieron muestras y confirmaron la probabilidad de éxito de la misma.

### CONCLUSIONES GENERALES

Para cumplir los objetivos trazados en este trabajo se realizaron varias entrevistas y encuestas. También se observó el modelamiento en el proceso de producción de software educativo del Ministerio de Educación, para corroborar la situación actual de la necesidad de introducción de un lenguaje de modelado que logre representar todos los elementos característicos y específicos de las aplicaciones educativas cubanas. Por los resultados obtenidos se arriban a las siguientes conclusiones:

- Existen un conjunto de notaciones y lenguajes para el desarrollo de aplicaciones pero que no permiten la representación eficiente de todos los elementos que desde el punto de vista pedagógico son abordados en el software educativo cubano y el Lenguaje de Modelado para Aplicaciones Educativas representa un paso de avance en el modelamiento en el proceso de producción de las aplicaciones de este tipo.
- Actualmente se trabaja en el Ministerio de Educación con aplicaciones educativas a través de la utilización de las tecnologías multimedia e hipermedia, utilizando un conjunto de lenguajes de alto nivel y no se emplea mucho tiempo ni esfuerzos en el modelamiento del software que se desarrolla y donde se realiza, los métodos que emplean no son los más óptimos.
- En el MINED existe gran cantidad de factores limitando la introducción del lenguaje lo que ha frenado este proceso, pero aplicando una estrategia para la evacuación de los mismos, el proceso de introducción puede verse favorecido y finalmente tener éxito.
- Se identificaron muy pocos elementos a modificar y no representados debido a que solo aislados proyectos emplean la notación y llevan un corto período de utilización de dicha herramienta para el modelado.
- El ApEM-L es casi desconocido en la empresa lo que contribuye a que se haya insertado en pocos proyectos productivos, de ahí se define como “bajo” el nivel de aplicación.
- La estrategia propuesta puede ser aplicada puesto que se declara aceptable según las valoraciones de los expertos, que además dieron muestras y confirmaron la probabilidad de éxito de la misma.

### RECOMENDACIONES

Tomando como base la investigación realizada y la experiencia acumulada durante la realización de este trabajo, se proponen las siguientes recomendaciones:

Aplicar la estrategia en los proyectos productivos para que se logre un buen uso de ApEM-L y obtener así los beneficios que trae consigo su aplicación.

## REFERENCIAS BIBLIOGRÁFICAS

**Aguilar, J. 1997.** *Material interactivo asistido por la computadora: análisis de la experiencia canadiense.* 1997. Volume, 205-222 DOI.

**Autores, Colectivo de. 1998.** Software Educativo. [Online] 1998.

<http://cavalletto.freeservers.com/pag%20educativa%20de%20amortiza.htm>.

**Baumeistier, Hubert, Koch, Nora y Mandel, Luis. 2001.** Towards a UML Extension for Hypermedia Design. [Online] 2001. [Cited: noviembre 12, 2007.]

<http://citeseer.ist.psu.edu/cache/papers/cs/31048/http:zSzzSzwww.fast.dezSzProjektezSzforsoftzSzuml99zSzuml99.pdf/baumeister99towards.pdf>.

**Berner, Stefan, y otros. 2001.** A visualization concept for hierarchical objects models. [Online] 2001. [Cited: noviembre 12, 2007.]

<http://citeseer.ist.psu.edu/cache/papers/cs/21217/http:zSzzSzwww.ifi.unizh.chzSzgroupszSzreqzSzstaffzSzglinzzSz.zSz.zSzftpzSzpaperszSzASE98.pdf/berner98visualization.pdf>.

**Booch, Grady, Jacobson, Ivar y Rumbaugh, James. 2000.** *El Lenguaje Unificado de Modelado. Manual de Referencia (traducción del original en inglés: The Unified Modeling Language. Reference Manual, 1999).* Madrid : Addison-Wesley, 2000. 1ra edición.

**Bou, G. 1997.** *El guión multimedia.* Madrid, Anaya : CABERO, J Y DUARTE, A, 1997.

**Cataldi, I. Z. 2000.** *Metodología de diseño, desarrollo y evaluación de software educativo. Facultad de Informática.* 2000. UNLP: 1-75..

**Ciudad Ricardo, Febe Angel. 2007.** ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI. (*Tesis para optar por la categoría científica Master en Informática Aplicada*). La Habana : UCI, 2007.

**Corrales Díaz, Carlos. 1994.** LA TECNOLOGIA MULTIMEDIA: Una Nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones. [Online] 1994. [Cited: noviembre 20, 2007.] <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm#inicio>.

**Díaz-Antón, María Gabriela. 2002.** Propuesta de una metodología de desarrollo de software. [Online] 2002. [Cited: octubre 14, 2007.] <http://www.infedu.coord.usb.ve/proyectos/proyecto3.html>.

- Fernández, A. 2000.** *El formador de Formación Profesional y Ocupacional*. Barcelona : Octaedro, 2000.
- Lee, W. W. y Owens, D. L. 2000.** *Multimedia based instruction*. Massachussets : Jossey - Bass, 2000.
- Lorente Rodríguez, Abel Ernesto. 2006.** *Plataformas para el desarrollo y gestión de Cursos Educativos Multimedia*. La Habana, Cuba : CD - ROM Memorias Uciencia, Universidad de las Ciencias Informáticas (UCI), 2006.
- Marqués, Pere. 1996.** El software educativo. [Online] 1996. [Cited: diciembre 10, 2007.] [http://www.lmi.ub.es/te/any96/marques\\_software/](http://www.lmi.ub.es/te/any96/marques_software/)
- MIC, Colectivo Autores. 2003.** *Industria Cubana del Software. Perfil y perspectivas.* . La Habana : Grupo de Promoción y Publicidad de la InCuSoft, 2003.
- Pastor, Juan Antonio. 1997.** La escritura hipermedia. [Online] 1997. [Cited: octubre 11, 2007.] <http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/saorin.htm>
- Pérez Fernández, Vicenta. 1999.** *Curso de Informática Educativa (Folleto)*. La Habana : IPSJAE, 1999.
- Pérez Huertas, F. J. 1998.** *INTRODUCCION A LA MULTIMEDIA REALIZACION PRODUCCION REALIZACION Y PRODUCCION DE PROGRAMAS*. s.l.: INSTITUTO OFICIAL DE RADIO Y TELEVISION, 1998. ISBN: 978-84-88788-34-4.
- Pressman, Roger S. 2002.** *Ingeniería de Software. Un enfoque práctico (Traducción del original en inglés: Software Engeering. A practical Aproach)*. New York : Mac Graw Hill, 2002. 5ta edición.
- Rodríguez, Raúl. 2000.** *Introducción a la Informática Educativa (conferencia)*. s.l. : ISPJAE, 2000.
- Rumbaugh, James. 1997.** *Models through the development process. Journal of Object Oriented Programming, SIGS Publications. Mayo, 1997.* 1997.
- Sauer, Stefan y Engels, Gregor. 2001.** Extending UML for Modeling of Multimedia Applications. [Online] 2001. [Cited: octubre 5, 2007.] <http://www.itec.uniklu.ac.at/~harald/proseminar02/sauer1.pdf>



**BIBLIOGRAFÍA**

**Aguilar, J. 1997.** *Material interactivo asistido por la computadora: análisis de la experiencia canadiense.* 1997. Volume, 205-222 DOI.

**Almeida Cintra, Nurileidis y Viera Cisnero, Yoannia. 2007.** *Principios para la evaluación y certificación de la calidad de los productos de software educativo en la UCI. (tesis para optar por el título de Ingeniero en Ciencias Informáticas).* La Habana : UCI, 2007.

**Autores, Colectivo de. 1998.** Software Educativo. [Online] 1998.

<http://cavalletto.freeservers.com/pag%20educativa%20de%20amortiza.htm>

**Baumeistier, Hubert, Koch, Nora y Mandel, Luis. 2001.** Towards a UML Extension for Hypermedia Design. [Online] 2001. [Cited: noviembre 12, 2007.]

<http://citeseer.ist.psu.edu/cache/papers/cs/31048/http:zSzzSzwww.fast.dezSzProjektezSzforsoftzSzuml99zSzuml99.pdf/baumeister99towards.pdf>

**Berner, Stefan, y otros. 2001.** A visualization concept for hierarchical objects models. [Online] 2001. [Cited: noviembre 12, 2007.]

<http://citeseer.ist.psu.edu/cache/papers/cs/21217/http:zSzzSzwww.ifi.unizh.chzSzgroupszSzreqzSzstaffzSzqlinzzSz..zSz..zSzftpzSzpaperszSzASE98.pdf/berner98visualization.pdf>

**Booch, Grady, Jacobson, Ivar y Rumbaugh, James. 2000.** *El Lenguaje Unificado de Modelado. Manual de Referencia (traducción del original en inglés: The Unified Modeling Language. Reference Manual, 1999).* Madrid : Addison-Wesley, 2000. 1ra edición.

**Bou, G. 1997.** *El guión multimedia.* Madrid, Anaya : CABERO, J Y DUARTE, A, 1997.

**Cataldi, I. Z. 2000.** *Metodología de diseño, desarrollo y evaluación de software educativo. Facultad de Informática.* 2000. UNLP: 1-75..

**Ciudad Ricardo, Febe Angel. 2007.** ApEM – L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UCI. *(Tesis para optar por la categoría científica Master en Informática Aplicada).* La Habana : UCI, 2007.

- Corrales Díaz, Carlos. 1994.** LA TECNOLOGIA MULTIMEDIA: Una Nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones. [Online] 1994. [Cited: noviembre 20, 2007.] <http://iteso.mx/~carlosc/pagina/documentos/multidef.htm#inicio>
- Díaz-Antón, María Gabriela. 2002.** Propuesta de una metodología de desarrollo de software. [Online] 2002. [Cited: octubre 14, 2007.] <http://www.infedu.coord.usb.ve/proyectos/proyecto3.html>
- Fernández, A. 2000.** *El formador de Formación Profesional y Ocupacional*. Barcelona : Octaedro, 2000.
- Lee, W. W. y Owens, D. L. 2000.** *Multimedia based instruction*. Massachussets : Jossey - Bass, 2000.
- Lorente Rodríguez, Abel Ernesto. 2006.** *Plataformas para el desarrollo y gestión de Cursos Educativos Multimedia*. La Habana, Cuba : CD - ROM Memorias Uciencia, Universidad de las Ciencias Informáticas (UCI), 2006.
- Marqués, Pere. 1996.** El software educativo. [Online] 1996. [Cited: diciembre 10, 2007.] [http://www.lmi.ub.es/te/any96/marques\\_software/](http://www.lmi.ub.es/te/any96/marques_software/)
- MIC, Colectivo Autores. 2003.** *Industria Cubana del Software. Perfil y perspectivas*. . La Habana : Grupo de Promoción y Publicidad de la InCuSoft, 2003.
- Pastor, Juan Antonio. 1997.** La escritura hipermedia. [Online] 1997. [Cited: octubre 11, 2007.] <http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/saorin.htm>
- Pérez Fernández, Vicenta. 1999.** *Curso de Informática Educativa (Folleto)*. La Habana : IPSJAE, 1999.
- Pérez Huertas, F. J. 1998.** *INTRODUCCION A LA MULTIMEDIA REALIZACION PRODUCCION REALIZACION Y PRODUCCION DE PROGRAMAS*. s.l. : INSTITUTO OFICIAL DE RADIO Y TELEVISION, 1998. ISBN: 978-84-88788-34-4.
- Pressman, Roger S. 2002.** *Ingeniería de Software. Un enfoque práctico (Traducción del original en inglés: Software Engeering. A practical Aproach)*. New York : Mac Graw Hill, 2002. 5ta edición.
- Rodríguez, Raúl. 2000.** *Introducción a la Informática Educativa (conferencia)*. s.l. : ISPJAE, 2000.

**Rumbaugh, James. 1997.** *Models through the development process. Journal of Object Oriented Programming, SIGS Publications. Mayo, 1997. 1997.*

**Sauer, Stefan y Engels, Gregor. 2001.** Extending UML for Modeling of Multimedia Applications. [Online] 2001. [Cited: octubre 5, 2007.] <http://www.itec.uniklu.ac.at/~harald/proseminar02/sauer1.pdf>

## ANEXOS

**Anexo 1: Guía para las entrevistas realizadas a los líderes de cada proyecto productivo de los que conforman la muestra: “A jugar”, “Multisaber” y “Español para no hispanohablantes”**

1. ¿Cuáles son las notaciones que se utilizan hoy para el modelado de las aplicaciones multimedia en el MINED?
2. ¿Qué artefactos se generan o utilizan hoy en la producción?
3. ¿Logran los artefactos utilizados representar todos los elementos del Software Educativo producido en el Ministerio de Educación?
4. ¿Cuáles son los elementos distintivos del Software Educativo que se produce el Ministerio de Educación?
5. ¿Cuáles de estos elementos de los software educativos desarrollados no son representados por los artefactos que se utilizan actualmente?
6. ¿Qué artefactos o elementos de estos deben de mantenerse en futuras soluciones?
7. ¿Qué cantidad de personas están involucradas en el proceso de producción y cuáles son los diferentes roles que existen?

**Anexo 2: Encuesta realizada a especialistas y miembros del equipo de desarrollo de los proyectos productivos: “A jugar” y “Multisaber”; los cuales forman parte de la muestra, donde se utiliza ApEM – L**

Compañero(a): Actualmente existe una investigación sobre el Lenguaje de Modelado para Aplicaciones Educativas y su repercusión. La encuesta que se le presenta a continuación es sencilla y agradeceríamos fidelidad en la información para que sirva eficientemente al cumplimiento del objetivo de esta investigación.

Muchas Gracias, Colectivo de Autores.

Nombre: \_\_\_\_\_

Rol: \_\_\_\_\_

Estudiante: \_\_\_\_ Especialista: \_\_\_\_\_

1. ¿Considera usted ventajoso la aplicación de una extensión como ApEM – L para la modelación de aplicaciones educativas?

Totalmente \_\_\_\_\_ Parcialmente \_\_\_\_\_ No \_\_\_\_\_

2. ¿Se siente usted cómodo al trabajar con la notación ApEM – L en sus distintas vistas?

Totalmente \_\_\_\_\_ Parcialmente \_\_\_\_\_ Muy poco \_\_\_\_\_ Nulo \_\_\_\_\_

3. ¿Considera usted que es sencilla, fácil de entender y aplicar la notación con unos pocos conocimientos de programación?

Si\_\_\_\_ No\_\_\_\_

4. Marque con una X las vistas y los diagramas que considera mejor trabajados en ApEM – L para la modelación de aplicaciones educativas:

Vista Estática\_\_\_\_\_

Vista de Arquitectura\_\_\_\_\_

Vista de Comportamiento \_\_\_\_\_

Vista de Presentación\_\_\_\_\_

5. ¿Cree eficiente la utilización de las descripciones textuales de los casos de uso al describir los flujos de eventos de un software educativo?

Totalmente\_\_\_\_ Parcialmente\_\_\_\_ Nulo\_\_\_\_\_

6. Marque con una X los elementos que a su consideración deban de reforzarse, mejorarse o ampliarse futuras versiones de la propuesta:

Diagrama de casos de uso\_\_\_\_\_

Diagrama de actividades\_\_\_\_\_

Diagrama de clases\_\_\_\_\_

Diagrama de componentes\_\_\_\_\_

Diagrama de despliegue\_\_\_\_\_

Diagrama de secuencia\_\_\_\_\_

Diagrama de colaboración\_\_\_\_\_

Diagrama de estados\_\_\_\_\_

Diagrama de estructura de navegación\_\_\_\_\_

Diagrama de estructura de presentación\_\_\_\_\_

7. ¿Cree usted que debe desarrollarse un programa que permita la generación computacional de la documentación resultante de la modelación?

Si\_\_\_\_ No\_\_\_\_

8. ¿Considera que el uso de ApEM – L sustituye la necesidad en la implementación del uso del Guión y el Mapa de Navegación?

Totalmente\_\_\_\_ Parcialmente\_\_\_\_ Muy poco\_\_\_\_ Nulo\_\_\_\_\_

9. ¿Existe factor alguno que le impida a UD. la utilización de este lenguaje de modelado (ApEM-L)?

No\_\_\_\_ Si\_\_\_\_ ¿Cual o cuales? \_\_\_\_\_

10. ¿Durante la utilización de ApEM-L se ha percatado que seria ventajoso modificar o añadir cierto(s) elemento(s)?

No\_\_\_\_ Si\_\_\_\_ ¿Cual o cuales? \_\_\_\_\_

**Anexo 3: Encuesta realizada a especialistas, líderes de proyectos y miembros del equipo de desarrollo de los proyectos productivos: “Español para no hispanohablantes”, “A jugar” y “Multisaber”; los cuales forman parte de la muestra, donde se utiliza ApEM – L**

Compañero(a): Actualmente existe una investigación sobre el Lenguaje de Modelado para Aplicaciones Educativas y su repercusión. La encuesta que se le presenta a continuación es sencilla y

agradeceríamos fidelidad en la información para que sirva eficientemente al cumplimiento del objetivo de esta investigación.

Muchas Gracias, Colectivo de Autores.

Nombre: \_\_\_\_\_

Rol: \_\_\_\_\_

Estudiante: \_\_\_\_ Especialista: \_\_\_\_\_

Existen un grupo de factores que limitan la introducción de ApEM-L en los proyectos productivos de software educativo del Ministerio de Educación, algunos de ellos son:

\_\_\_\_\_ ApEM-L es algo muy novedoso todavía.

\_\_\_\_\_ Falta de personal capacitado para impartir la capacitación.

\_\_\_\_\_ Descontento con la Introducción de ApEM-L.

\_\_\_\_\_ Falta de interés por la capacitación para la aplicación de ApEM-L.

\_\_\_\_\_ No disponibilidad por de los medios y locales para la capacitación.

\_\_\_\_\_ Falta de experiencia para la aplicación del mismo.

\_\_\_\_\_ La aplicación del lenguaje trae consigo algunos cambios.

\_\_\_\_\_ Falta de tiempo para la migración desde el modelamiento que se hacía a aplicar ApEM-L.

\_\_\_\_\_ La aplicación del lenguaje trae consigo muchos cambios.

\_\_\_\_\_ Cambia el plan de trabajo del proyecto, así como las metas trazadas.

\_\_\_\_\_ Otros, ¿cuales? \_\_\_\_\_

Marque con una **X** los factores que Ud. identifica como los que están o estuvieron presentes en su entorno productivo en el momento de presentación o inserción de ApEM-L.

**Anexo 4: Encuesta realizada a especialistas y miembros del equipo de desarrollo de los proyectos productivos: “A jugar” y “Multisaber”; los cuales forman parte de la muestra, donde se utiliza ApEM – L**

Compañero(a): Actualmente existe una investigación sobre el Lenguaje de Modelado para Aplicaciones Educativas y su repercusión. La encuesta que se le presenta a continuación es sencilla y agradeceríamos fidelidad en la información para que sirva eficientemente al cumplimiento del objetivo de esta investigación.

Muchas Gracias, Colectivo de Autores.

Nombre: \_\_\_\_\_

Rol: \_\_\_\_\_

Estudiante: \_\_\_\_ Especialista: \_\_\_\_\_

De los siguientes elementos que pertenecen a las Vistas y los Diagramas de ApEM-L marque con una X los que Ud. utiliza en su entorno productivo:

**Estructura lógica***Vista Estática*

- *Diagrama de clases*
  - Clase\_\_\_\_\_
  - asociación\_\_\_\_\_
  - generalización\_\_\_\_\_
  - dependencia\_\_\_\_\_
  - realización\_\_\_\_\_
  - interfaz\_\_\_\_\_
- *Diagrama de casos de uso*
  - Caso de uso\_\_\_\_\_
  - actor\_\_\_\_\_
  - asociación\_\_\_\_\_
  - extensión\_\_\_\_\_
  - inclusión\_\_\_\_\_
  - generalización\_\_\_\_\_

*Vista de arquitectura*

- *Diagrama de componentes*
  - Componente\_\_\_\_\_
  - interfaz\_\_\_\_\_
  - dependencia\_\_\_\_\_
  - realización\_\_\_\_\_
- *Diagrama de despliegue*
  - Nodo\_\_\_\_\_
  - componente\_\_\_\_\_
  - dependencia\_\_\_\_\_
  - localización\_\_\_\_\_

**Comportamiento dinámico***Vista de comportamiento*

- *Diagrama de actividad*
  - Estado\_\_\_\_\_
  - actividad\_\_\_\_\_
  - transición de terminación\_\_\_\_\_
  - división\_\_\_\_\_
  - unión\_\_\_\_\_
- *Diagrama de colaboración*
  - Colaboración\_\_\_\_\_
  - interacción\_\_\_\_\_
  - rol de colaboración\_\_\_\_\_
  - mensaje\_\_\_\_\_
- *Máquina de estados*
  - Estado\_\_\_\_\_
  - evento\_\_\_\_\_
  - transición\_\_\_\_\_
  - acción\_\_\_\_\_
- *Diagrama de secuencia*
  - Interacción\_\_\_\_\_
  - objeto\_\_\_\_\_
  - mensaje\_\_\_\_\_
  - activación\_\_\_\_\_

**Gestión del modelo***Vista de Presentación*

- *Diagrama de Estructura de navegación*
  - *Clases de navegación*\_\_\_\_\_
  - *elementos de navegación*\_\_\_\_\_
  - *composición, asociación*\_\_\_\_\_
  - *dependencia, uso*\_\_\_\_\_
- *Diagrama de Presentación*
  - *Clases de presentación*\_\_\_\_\_
  - *clases Frameset*\_\_\_\_\_
  - *asociaciones*\_\_\_\_\_
  - *componentes medias*\_\_\_\_\_
  - *paquetes*\_\_\_\_\_

**Anexo 5: Modelo 1****Guía para asignar el peso a los criterios**

Nombre y Apellidos del evaluador \_\_\_\_\_

Le otorgará un peso a cada criterio de acuerdo a su opinión y el peso total de cada grupo debe sumar:

- Grupo No.1..... 25
- Grupo No.2..... 30
- Grupo no.3..... 15
- Grupo No.4.....30

Para que el peso total asignado sea 100.

Grupo No. 1: Criterios de mérito científico

1. Valor científico de la estrategia.  
Peso.....
2. Calidad de la investigación.  
Peso.....
3. Contribución científica.  
Peso.....

Grupo No. 2: Criterios implantación

4. Satisfacción de las necesidades de los integrantes de los proyectos.  
Peso.....
5. Necesidad del empleo de la propuesta.  
Peso.....
6. Uso del ApEM-L.  
Peso.....

Grupo No.3: Criterios de flexibilidad

7. Adaptabilidad a proyectos productivos.



Peso.....

8. Uso de las herramientas necesarias para la elaboración de los artefactos.

Peso.....

Grupo No.4: Criterios de impacto

9. Repercusión en los proyectos productivos.

Peso.....

10. Organización en el proceso de documentación del software.

Peso.....

11. Posibilidades de aplicación.

Peso.....

**Anexo 6: Modelo 2**

**Guía para la evaluación**

- Criterios de medida que se evalúan en una escala de **1 a 5**:

Grupo No. 1: Criterios de mérito científico

- 1) Valor científico de la estrategia.

Peso.....

- 2) Calidad de la investigación.

Peso.....

- 3) Contribución científica.

Peso.....

Grupo No. 2: Criterios implantación

- 4) Satisfacción de las necesidades de los integrantes de los proyectos.

Peso.....

- 5) Necesidad del empleo de la propuesta.

Peso.....

- 6) Uso del ApEM-L.

Peso.....

Grupo No.3: Criterios de flexibilidad

- 7) Adaptabilidad a proyectos.

Peso.....

- 8) Uso de las herramientas necesarias para la elaboración de los artefactos.

Peso.....

Grupo No.4: Criterios de impacto

9) Repercusión en los proyectos productivos.

Peso.....

10) Organización en el proceso de documentación del software.

Peso.....

11) Posibilidades de aplicación.

Peso.....

- Categoría final de la estrategia

\_\_\_ Excelente: Alta novedad científica, con aplicabilidad y resultados relevantes.

\_\_\_ Bueno: Novedad científica, resultados destacados.

\_\_\_ Aceptable: Suficientemente bueno con reservas.

\_\_\_ Cuestionable: No tiene relevancia científica y los resultados son malos.

\_\_\_ Malo: No aplicable.

- Valoración final de la estrategia

Sugerencias del evaluador para mejorarla: \_\_\_\_\_

\_\_\_\_\_.

Elementos críticos que deben mejorarse: \_\_\_\_\_

\_\_\_\_\_.

## GLOSARIO DE TÉRMINOS

**Proyecto:** Elemento organizativo a través del cual se gestiona el desarrollo del software, tiene como resultado la versión de un producto.

**Líder de proyecto:** Rol encargado de dirigir un proyecto productivo en particular.

**Software:** Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

**Proceso de desarrollo:** Definición del conjunto completo de actividades necesarias para transformar los requisitos de un usuario en un producto. Plantilla para crear proyectos.

**Artefacto:** Pieza de información utilizada o producida por un proceso de desarrollo de software como un documento externo o el producto de un trabajo. Un artefacto puede ser un modelo, una descripción o el software.

**Ingeniería de Software:** Es una tecnología multicapa en la que, se pueden identificar: los métodos (indican cómo construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos).

**Metodologías:** Se entiende como la parte del proceso de investigación que permite sistematizar los métodos y las técnicas necesarios para llevarla a cabo. Son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

**Metodologías de Desarrollo:** Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

**Lenguaje de modelado:** Es un conjunto estandarizado de símbolos y de modos de disponerlos para modelar un diseño de software orientado a objetos o parte de él.

**Herramienta CASE:** *Computer Aided Software Engineering*, Ingeniería de Software Asistida por Ordenador. Diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el costo de las mismas en términos de tiempo y de dinero. Conjunto de

programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.