

Manuel Caro – Raúl Toscano – Filadelfia Hernández – María  
David

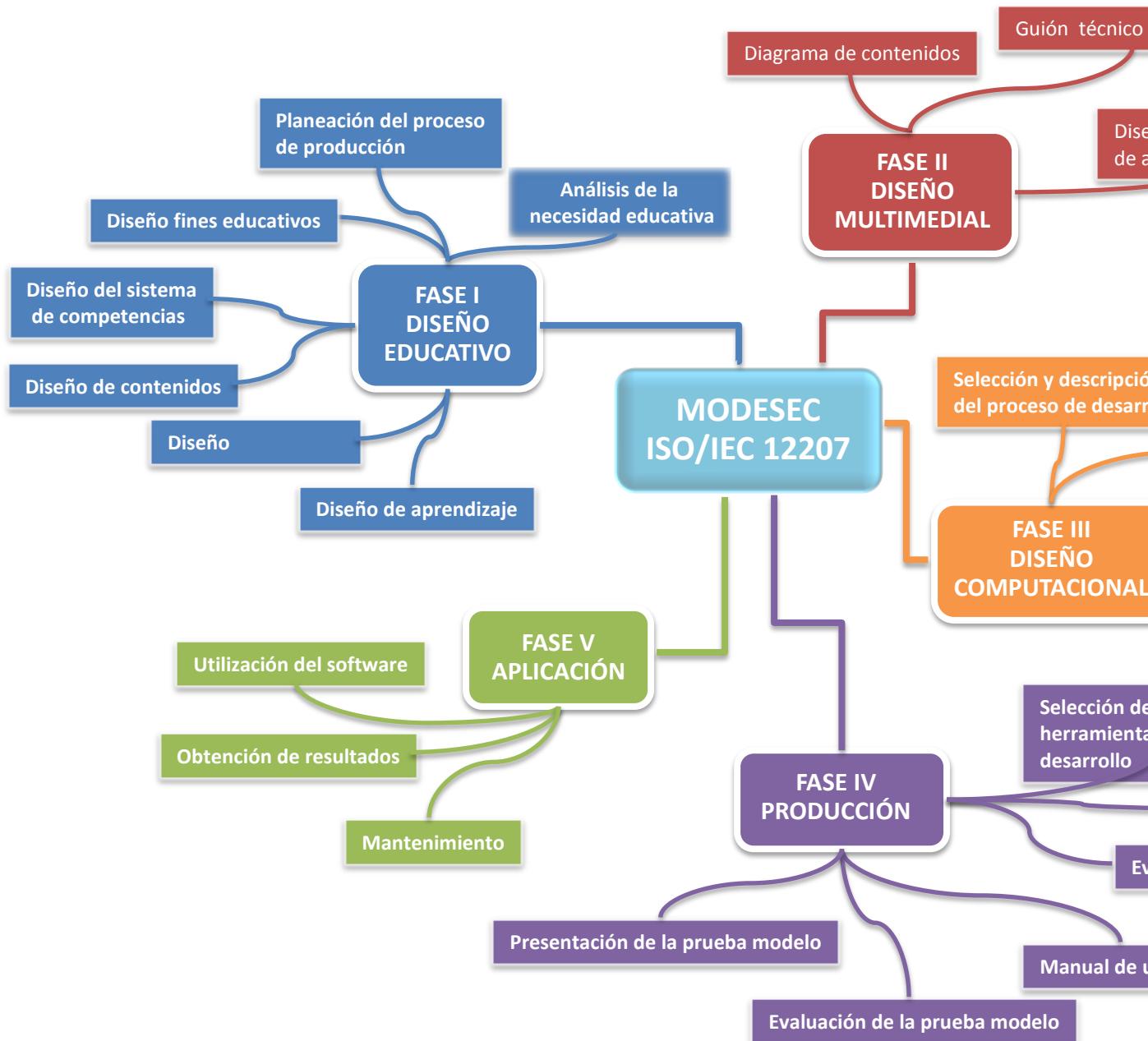
# Diseño de software educativo basado en competencias

Enfoque práctico

2012

## **Capítulo I**

### **INTRODUCCIÓN AL MODELO MODESEC-ISO/IEC 12207**



## 1.1 SOFTWARE EDUCATIVO

El software educativo se puede definir de dos formas, una de estas se concibe como la utilización intencional de todo software para mediar el proceso de enseñanza-aprendizaje; la otra concepción, es aquel software que es desarrollado con la finalidad de ser utilizado como herramienta didáctica en el proceso enseñanza-aprendizaje.

El software educativo se ha convertido en una herramienta didáctica utilizada en las aulas de clases como dinamizadores de los conocimientos y como propiciadores de nuevos entornos que parte de una necesidad manifiesta en el contexto educativo<sup>1</sup>. El software educativo como un elemento didáctico, diseña espacios y ambientes basados en los requerimientos cognitivos de los estudiantes. Lo anterior implica que en su realización debe tener en cuenta no solo aspectos técnicos sino también aspectos de aprendizaje, curriculares y de contenido específico, es decir de un maestro<sup>2</sup>.

El maestro entonces, pasa de ser un transmisor de conocimiento, un repetidor de información sumergido en una cotidianidad monótona en su práctica pedagógica, a ser un creador de materiales elaborados en plataformas informáticas, un diseñador de ambientes de aprendizaje, por lo tanto a centrar su tarea pedagógica en la caracterización de las necesidades de sus alumnos y en la implementación de soluciones apoyado en las tecnologías de la información<sup>3</sup>.

Todos los programas educativos orientan y regulan el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a facilitar el logro de unos objetivos educativos específicos<sup>4</sup>.

Al elaborar software educativo se deben tener en cuenta en la creación de programas de calidad esto implica la eficacia en los procesos, esto hace que “la complejidad en la elaboración del software educativo radica en que requiere tomar múltiples decisiones no solo de orden técnico sino también de orden pedagógico y de diseño. Sin embargo, se podría afirmar que el aspecto central es el pedagógico, pues en torno a su concepción y las decisiones que se tomen allí, se determina la forma en que se desarrollan los restantes aspectos”<sup>5</sup>.

## 1.2 DESCRIPCIÓN GENERAL MODESEC-ISO/IEC 12207

El modelo de desarrollo de software educativo que se plantea en este escrito tiene como base fundamental el sistema de competencias, e integra cinco fases: *Diseño educativo, diseño multimedial, diseño computacional, producción y aplicación*. Este modelo es sencillo de aplicar y sus fases están bien documentadas, lo que lo hace idóneo para su utilización en procesos de desarrollo de software educativo.

Cada fase del modelo está bien delimitada y se encarga de un trabajo en particular así: en la Fase I se hace el estudio de los factores educativos que sustentarán el diseño del software en cuanto a la didáctica, la pedagogía y la ética. La Fase II, trata acerca de los aspectos estéticos y del sistema de comunicación, según la población, sistema de contenidos y estrategias planeadas el la fase anterior. En la Fase III se trabajan los fundamentos de ingeniería de software que requiere el desarrollo de una aplicación robusta y amigable. En la Fase IV se

<sup>1</sup> MOLINA VASQUEZ, Ruth. Algunos aportes sobre el software educativo. Disponible en: [http://www.karisma.org.co/documentos/softwareredp/APORTES\\_AL\\_CONCEPTO\\_DE\\_SOFTWARE\\_EDUCATIVO.doc](http://www.karisma.org.co/documentos/softwareredp/APORTES_AL_CONCEPTO_DE_SOFTWARE_EDUCATIVO.doc). Acceso 02 mayo de 2009.

<sup>2</sup> Ibíd.

<sup>3</sup> Ibíd.

<sup>4</sup> MARQUÈS, Pere. El software educativo. Universidad Autónoma de Barcelona

<sup>5</sup> Opcit, MOLINA.

desarrollan y ensamblan los componentes del software, finalmente en la Fase V se utiliza el software en el aula de clases, evaluando su pertinencia e impacto.

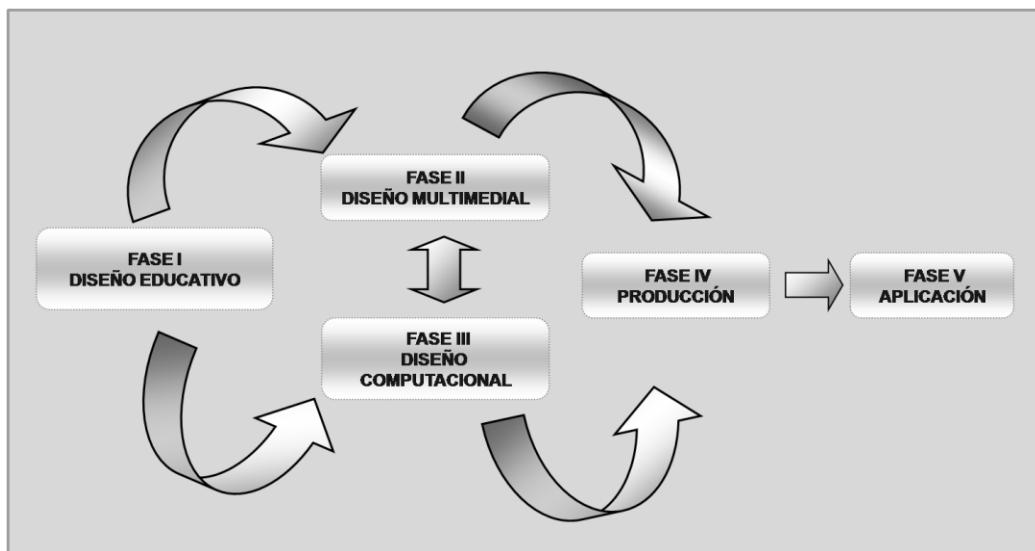


Figura 1. Modelo para el desarrollo de software educativo basado en competencias (MODESEC-ISO/IEC 12207 -2008)

El modelo sugerido se compone de cinco fases que detallan paso a paso los aspectos a tener en cuenta para la creación de software educativo, distribuyéndose en cada una de ellas los componentes de los procesos principales del ciclo de vida especificados en el estándar ISO/IEC 12207. La fase inicial constituye la descripción del *diseño educativo*, en la cual se analiza la necesidad educativa, se plantean los objetivos de aprendizaje y se describen las competencias que se pretenden desarrollar con la aplicación; del diseño de éstas resultan las siguientes subfases: *diseño de contenidos*, *diseño pedagógico* y *diseño de aprendizaje*. Las competencias son el aspecto fundamental que abarca el desarrollo de este modelo, las cuales son primordiales para la realización de las fases a seguir conformadas por el *diseño computacional* y el *diseño multimedial*, estas se encargan del análisis y modelado del software, y del sistema de comunicación hombre-máquina. En la fase de *producción* se ensamblan los componentes elaborados o recolectados, según el caso. La última fase es la de *aplicación*, donde se hacen las pruebas de rigor para evaluar el desempeño del software en los contextos para los que fue desarrollado.

### 1.3 ESTRUCTURA DEL MODELO: LAS COMPETENCIAS COMO EJE CENTRAL DEL MODESEC-ISO/IEC 12207

En este apartado se describen los aspectos que hacen especialmente particular este modelo que presenta el grupo de investigación EDUPMEDIA para el diseño de software educativo, cuyas características frente a los que en la actualidad se utilizan, brindan ventajas para su implementación en el contexto colombiano.

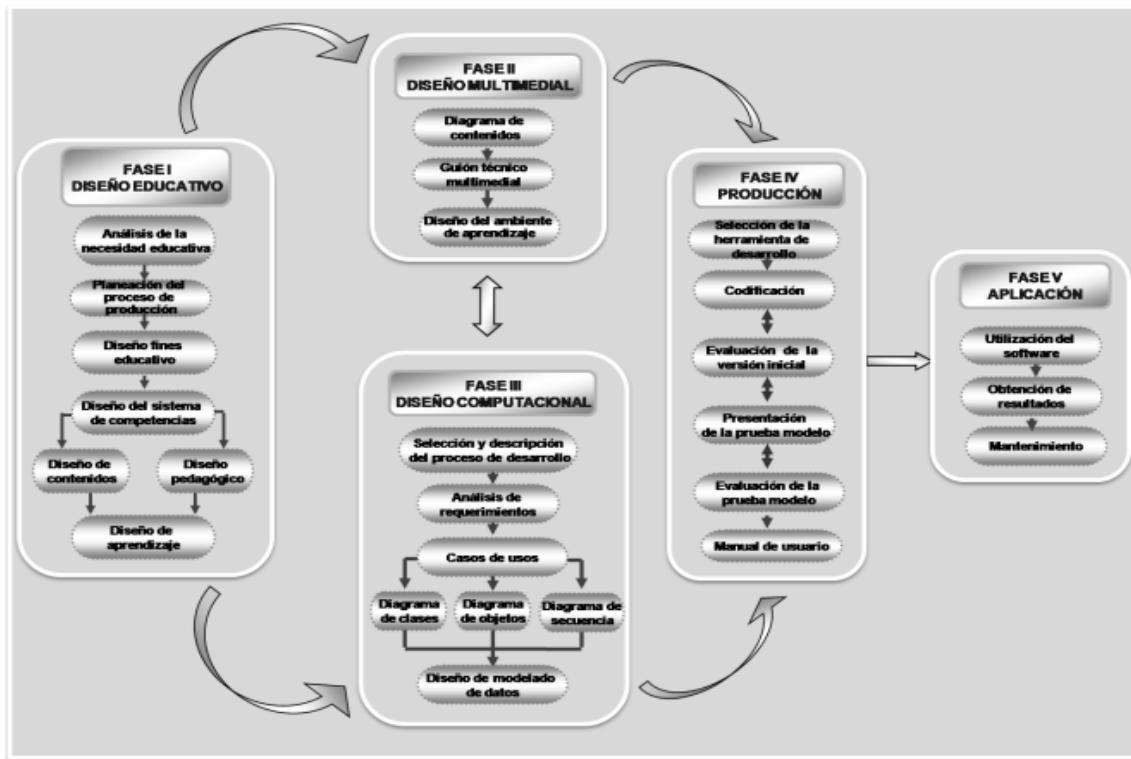


Figura 2. MODESEC-ISO/IEC 12207: FASES (MODESEC-ISO/IEC 12207 -2008)

Siendo las competencias el eje central del modelo, es necesario indicar las razones que lo justifican, con lo cual se brinda el sustento básico de su diferenciación frente a los modelos existentes así:

1. Los modelos utilizados en la actualidad no presentan una sección o apartado que brinde toda la información relacionada con las competencias requeridas según las necesidades educativas detectadas en la investigación previa, como sí lo contempla el modelo MODESEC-ISO/IEC 12207, pues en éste no sólo se enuncian las competencias, sino que también se establece un formato de presentación (por objetivos y normas), ver *formato 5*, que agrupa todos los componentes de las misma, lo que permite al diseñador recoger los aspectos necesarios para realizar una documentación completa, lo cual guiará con mayor certeza el proceso de desarrollo.
2. El diseño del sistema de competencias actúa como eje integrador del proceso de diseño y producción del software educativo, pues cada una de sus secciones brinda información de entrada para otros procesos del desarrollo, convirtiéndose así en el centro de partida y principal fuente de información para las fases siguientes. Así pues, la sección de *Recursos* dentro de la **Norma** da nociones precisas acerca de las herramientas complementarias que requiere el estudiante para realizar las actividades propuestas, las cuales debe proveer el software educativo.
  - La sección de los *Elementos* brinda las bases de las actividades que el usuario debe desarrollar con el software, por lo cual el diseñador debe tomar la decisión de hacer una ventana para cada *elemento* o agrupar varios elementos en una ventana; cabe destacar que a mayor cantidad de elementos por ventana, ésta se torna más compleja.

- El listado de conceptos que se obtienen de las competencias y que se consignan en la sección de los *Conceptos* da origen al *diseño de contenidos*, en el cual se determina el nivel de profundidad y el tipo de vocabulario que se utilizará en el software.
- La sección del *Contexto* brinda al diseñador información que le permite determinar las características del ambiente que debe rodear las actividades de aprendizaje que se van a realizar con el software. Éstas pueden ser de tipo físico (laboratorios u otros lugares), o de tiempo (actualidad, siglo XIX).
- La sección de los *Indicadores* servirá para el diseño del sistema de evaluación del software educativo, pues en ellos se evidencia las habilidades y conocimientos que debe demostrar una persona para la realización de una tarea (elemento) en particular dentro de la competencia.

Con lo dicho anteriormente se evidencia que el eje central de todo el modelo son las competencias.

Por otra parte, desde lo pedagógico se puede aducir que el modelo MODESEC-ISO/IEC 12207 no se centra con ninguna teoría en particular, hecho que lo hace flexible y le permite amoldarse a las condiciones propias de la población objeto y a las características propias del entorno de la necesidad estudiada. También es factible que el modelo se adapte a las directrices pedagógicas propia de cada institución educativa, las cuales se encuentran consignadas en su Proyecto Educativo Institucional (PEI), así no se entraría a imponer un software que se trabaje en forma diferente a la realidad escolar.

## 1.4 APLICACIÓN DE ISO/IEC 12207 EN EL MODELO MODESEC-ISO/IEC 12207

El modelo propuesto es una extensión del estándar ISO/IEC 12207 (*ver figura 3*) para la unificación del ciclo de vida del software. Los procesos principales del ciclo de vida consignados en el estándar se encuentran distribuidos al interior del modelo MODESEC-ISO/IEC 12207, así:

- La Fase I (Diseño educativo) contiene la tarea de *descripción de la necesidad* contenida en la actividad de *Iniciación del Proceso de Adquisición*. *Ver figura 4*
- La Fase II (Diseño multimedial), *Ver figura 9*, contiene las siguientes tareas:
  - *Producir una arquitectura del sistema e identificar elementos hardware y software* de la actividad de *Diseño de la arquitectura del sistema del Proceso de Desarrollo*.
  - *Identificar los elementos de los componentes del software* de la actividad *Diseño de la Arquitectura del software del Proceso de Desarrollo*.
- La Fase III (Diseño computacional) contiene las siguientes tareas:
  - *Definición de los requerimientos del software* contenida en la actividad de *Iniciación del Proceso de Adquisición*. *Ver figura 4*
  - *Seleccionar un modelo de ciclo de vida* contenida en la actividad de *Planificación del Proceso de Suministro*. *Ver figura 5*
  - *Definir o seleccionar un modelo de desarrollo* contenido en la actividad de *Implementación del Proceso de Desarrollo*. *Ver figura 6*
  - *Requisitos del software y Diseño detallado del software* contenida en las la actividad “*Actividades del software*” del *Proceso de Desarrollo*. *Ver figura 8*
  - La actividad *Diseño de la arquitectura del software del Proceso de desarrollo*. *Ver figura 9*

- La Fase IV (Producción) contiene las siguientes tareas y actividades:
  - *Definición del sistema de requerimientos*, que se encuentra consignada en la actividad de **Iniciación del Proceso de Adquisición**. Ver figura 4
  - La tarea *Seleccionar, adaptar, y usar normas, métodos, herramientas y lenguajes de programación*. La tarea *Desarrollar y preparar planes para preparar las actividades del proceso de desarrollo*. Que se encuentra consignada en la actividad de **Implementación del Proceso de desarrollo**. Ver figura 6
  - *Requisitos del sistema, Integración del sistema y Pruebas de calificación del sistema*. Que se encuentran consignadas a las *Actividades del sistema* del **Proceso de Desarrollo**. Ver figura 7
  - *Diseño detallado del software y Pruebas de calificación del software* se encuentran consignadas en las *Actividades del Software* en el **Proceso de Desarrollo**. Ver figura 8
  - Las actividades *Diseño de la arquitectura del sistema, Diseño de la arquitectura del Software, Codificación y pruebas del software, y la Integración del Software* pertenecen al **Proceso de Desarrollo**. Ver figura 9 y 10
  - *Establecer procedimientos para la resolución de problemas y para probar el producto* consignados en la actividad de **Implementación del Proceso de Operación**. Ver figura 11
- La Fase V (Aplicación) contiene las siguientes tareas y actividades:
  - *Asegurar la operación y mantenimiento del software entregable* que está contenida en la actividad de **Implementación del Proceso de Desarrollo**. Ver figura 6
  - *Requisitos del sistema* que se encuentra en *Las actividades del sistema* del **Proceso de desarrollo**. Ver figura 7
  - La actividad *Diseño de la arquitectura del sistema* del **Proceso de Desarrollo**. Ver figura 9
  - *Establecer procedimientos para la resolución de problemas* que está contenida en la actividad de **Implementación del Proceso de operación**. Ver figura 11
  - Las actividades de *Implementación del proceso* (Ver figura 12), *análisis de problemas y modificaciones* (Ver figura 13), *implementación de las modificaciones* (Ver figura 14), *migración* (Ver figura 15) y *retirada del software* (Ver figura 16) del **Proceso de Mantenimiento**.

Los procesos de apoyo al ciclo de vida se desarrollan en cada fase del modelo, así pues, la *documentación*, el *aseguramiento de la calidad* y la *solución de problemas* son actividades de obligatoria realización en cada una de las actividades del modelo. Ver ejemplo de la aplicación de la *norma ISO/IEC 12207*.

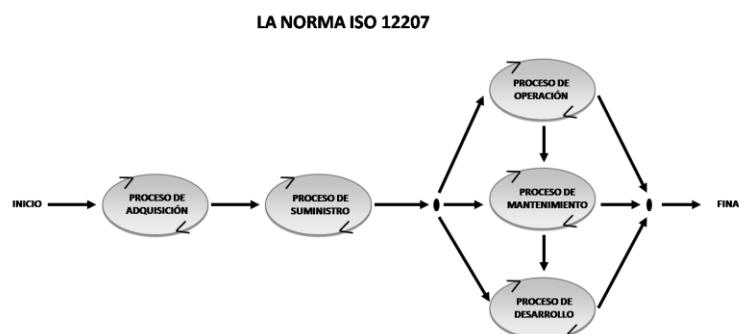


Figura 3. Norma ISO/IEC 12207.

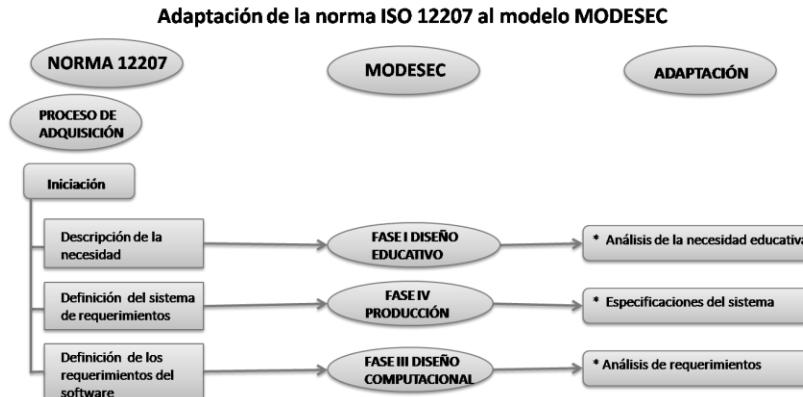


Figura 4. Adaptación de la norma ISO 12207, proceso de adquisición, actividad de Iniciación (MODESEC-ISO/IEC 12207 -2008)

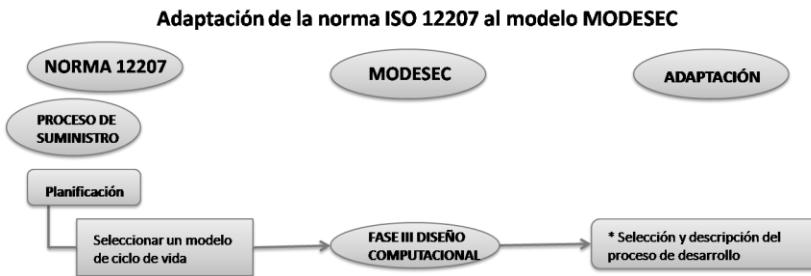


Figura 5. Adaptación de la norma ISO 12207, proceso de suministro, actividad de planificación (MODESEC-ISO/IEC 12207 -2008)

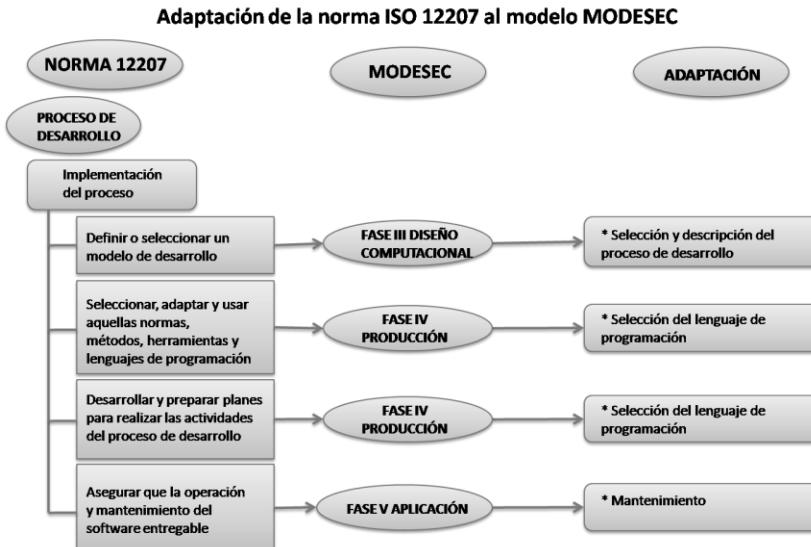


Figura 6. Adaptación de la norma ISO 12207, proceso de desarrollo, actividad de implementación del proceso (MODESEC-ISO/IEC 12207 -2008)

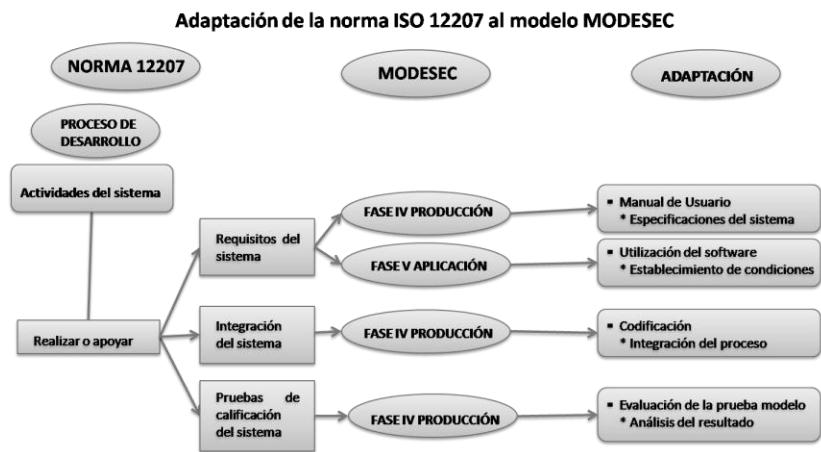


Figura 7. Adaptación de la norma ISO 12207, proceso de desarrollo, actividad del sistema (MODESEC-ISO/IEC 12207 -2008)

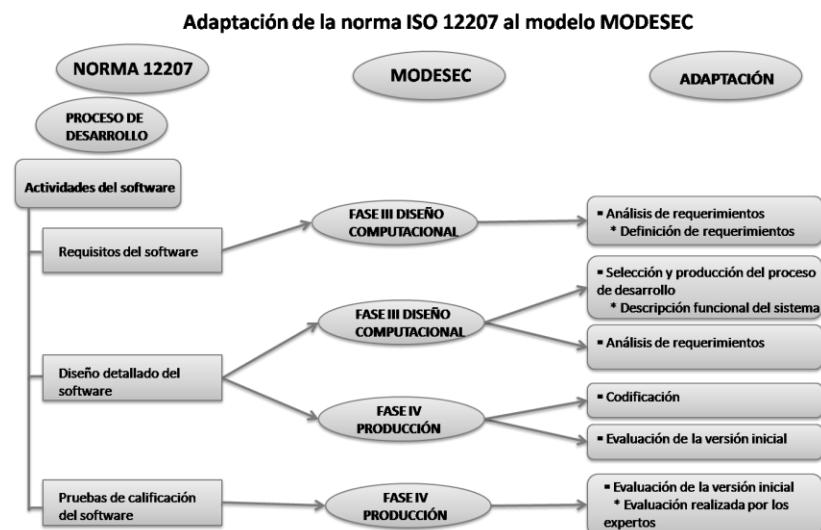


Figura 8. Adaptación de la norma ISO 12207, proceso de desarrollo, actividad del software (MODESEC-ISO/IEC 12207 -2008)

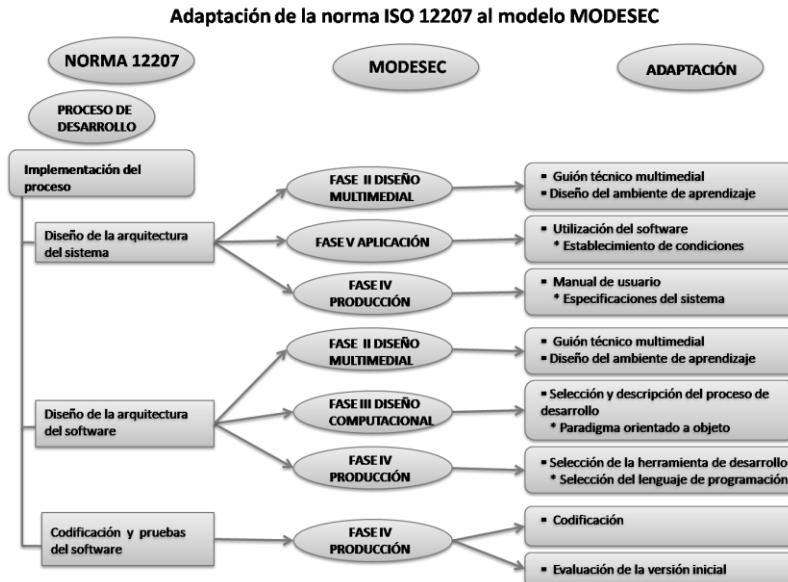


Figura 9. Adaptación de la norma ISO 12207, proceso de desarrollo, actividad del software (MODESEC-ISO/IEC 12207 -2008)

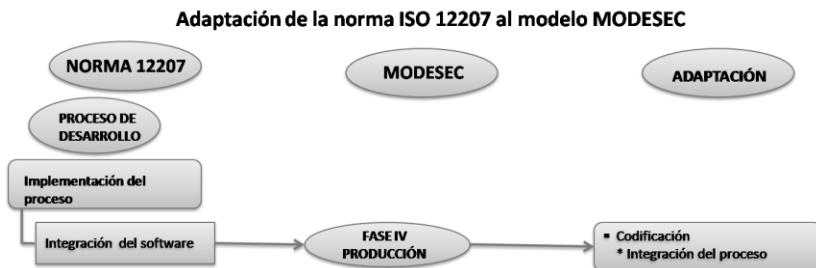


Figura 10. Adaptación de la norma ISO 12207, proceso de desarrollo, implementación del proceso (MODESEC-ISO/IEC 12207 -2008)

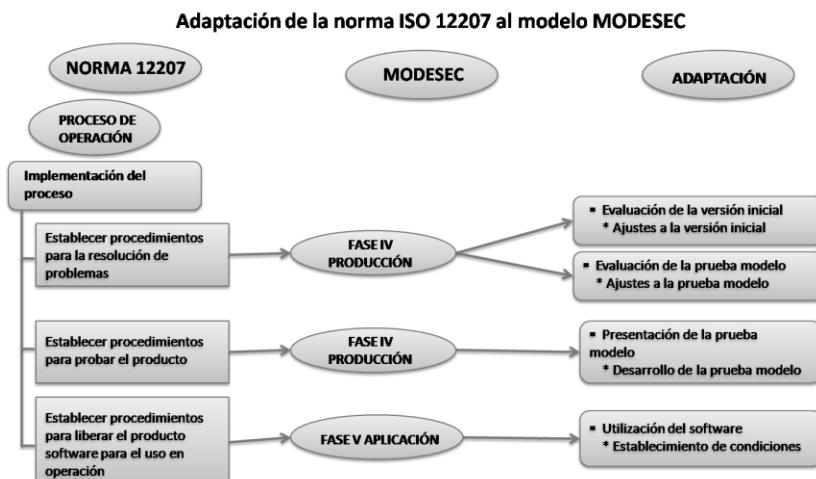


Figura 11. Adaptación de la norma ISO 12207, proceso de operación, implementación del proceso (MODESEC-ISO/IEC 12207 -2008)

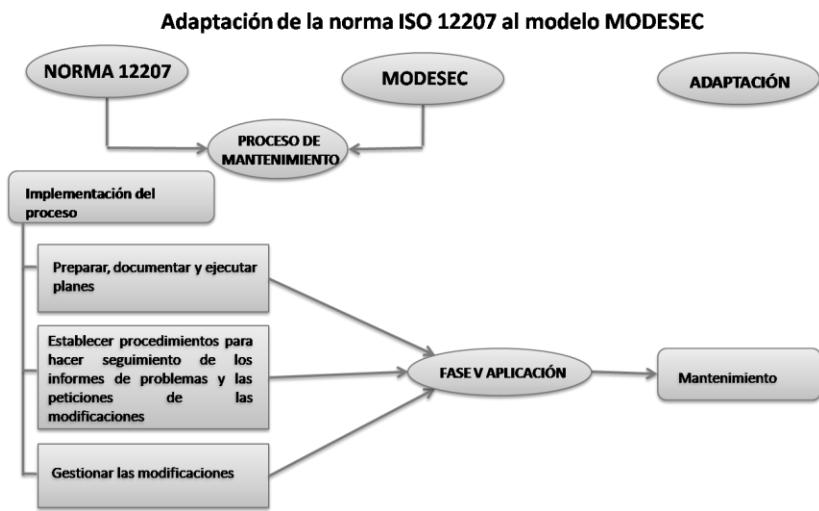


Figura 12. Adaptación de la norma ISO 12207, proceso de mantenimiento, implementación del proceso (MODESEC-ISO/IEC 12207 - 2008)

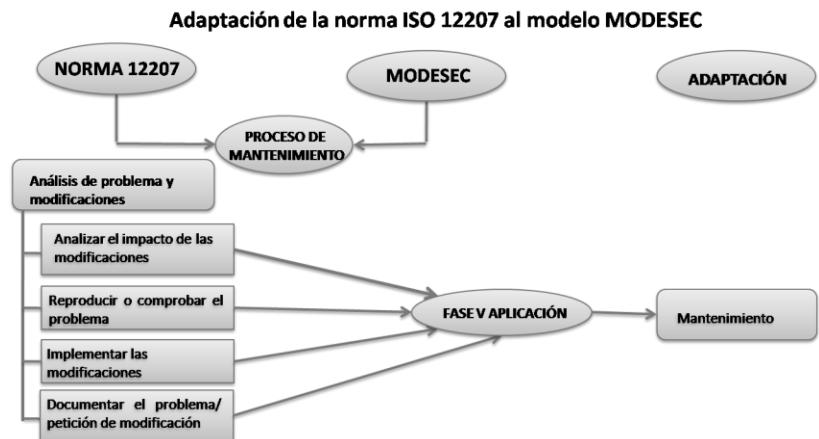


Figura 13. Adaptación de la norma ISO 12207, proceso de mantenimiento, análisis de problemas y modificaciones (MODESEC-ISO/IEC 12207 -2008)



Figura 14. Adaptación de la norma ISO 12207, proceso de mantenimiento, análisis de problemas y modificaciones (MODESEC-ISO/IEC 12207 -2008)

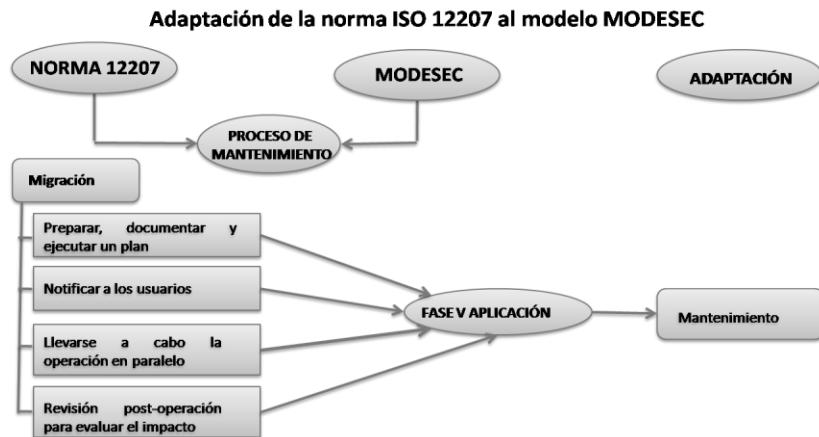


Figura 15. Adaptación de la norma ISO 12207, proceso de mantenimiento, análisis de problemas y modificaciones (MODESEC-ISO/IEC 12207 -2008)

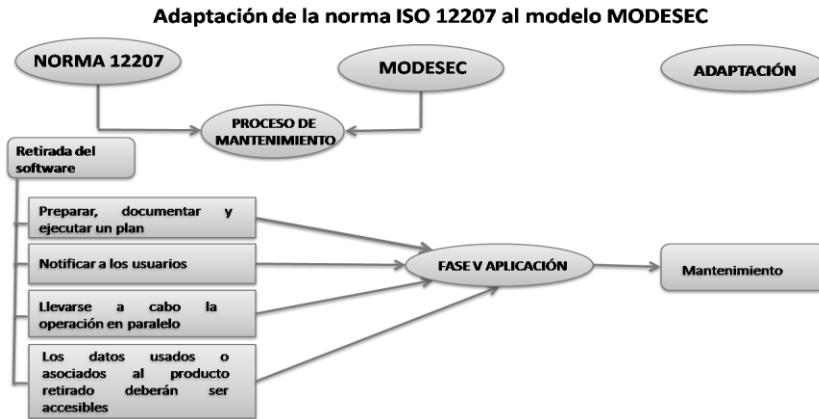


Figura 16. Adaptación de la norma ISO 12207, proceso de mantenimiento, análisis de problemas y modificaciones (MODESEC-ISO/IEC 12207 -2008)

## **PUNTOS CLAVES**

- MODESEC ISO/IEC 12207 se componen de cinco fases que detallan paso a paso los aspectos a tener en cuenta para crear software educativo, distribuyéndose en cada una de ellas los componentes de los procesos principales del ciclo de vida especificados en el estándar ISO/IEC 12207.
- El modelo que se plantea en este libro tiene como base fundamental el sistema de competencias, debido a que estas actúan como eje integrador del proceso de diseño y producción del software educativo, pues cada una de sus secciones brinda información de entrada para otros procesos del desarrollo, convirtiéndose así en el centro de partida y principal información de las demás fases.

## **REFERENTES**

- ISO 12207. Disponible en: <http://www.abelia.com/docs/12207cpt.pdf>. Consultado el 23 de febrero de 2009
- MARQUÈS, Pere. El software educativo. Universidad Autónoma de Barcelona
- MOLINA VASQUEZ, Ruth. Algunos aportes sobre el software educativo. Disponible en: [http://www.karisma.org.co/documentos/softwareredp/APORTES\\_AL\\_CONCEPTO\\_DE\\_SOFTWARE\\_EDUCATIVO.doc](http://www.karisma.org.co/documentos/softwareredp/APORTES_AL_CONCEPTO_DE_SOFTWARE_EDUCATIVO.doc). Acceso 02 mayo de 2009.

**Capítulo II**

**DISEÑO EDUCATIVO**

## 2. FASE I: DISEÑO EDUCATIVO

El diseño educativo es el punto de partida del modelo MODESEC-ISO/IEC 12207 en donde se identifican y analizan los factores educativos que intervienen en el diseño del software. La fase I de este modelo es fundamental para el desarrollo de las demás fases por tal motivo es necesario realizar un estudio exhaustivo sobre esos factores que afectan el aprendizaje que conlleve al establecimiento de un plan de trabajo en donde se especifique los aspectos necesarios como son la parte pedagógica, didáctica y la ética que configuran las distintas etapas que hacen parte de la fase del diseño educativo. A continuación se realiza una explicación detallada de las etapas que constituyen esta fase:



Figura 17. Fase I. Diseño educativo. (Modesec-ISO/IEC 12207-2008)

### 2.1 Análisis de la Necesidad Educativa

En este modelo se adopta la definición dada por Kaufman que concibe una necesidad educativa como la discrepancia entre estado educativo ideal (deber ser) y otro existente (realidad). Por consiguiente, su determinación debe llevar a resolver tres interrogantes que a continuación se mencionan, el tercero de los cuales refleja la necesidad: ¿qué es lo ideal? (Meta o aprendizaje esperado)?, ¿qué de esto se puede satisfacer con lo que existe?, (lo que se puede lograr o aprender con los medios y actividades aplicables) y ¿qué falta por alcanzar? (La necesidad). De este modo, la determinación de necesidades educativas en el entorno de enseñanza-aprendizaje es equivalente al establecimiento de lo que hay que aprender con apoyo de las nuevas tecnologías informáticas y de comunicación<sup>6</sup>.

<sup>6</sup> KAUFMAN, R.A. *Needs Assesment: What It Is How to Do It*. University Consortium on Instructional Development and Technology: San Diego, Cal, 1976.

Al iniciar un proceso de diseño y desarrollo de software educativo es importante tener presente los problemas que se evidencian en una población determinada para saber exactamente el tipo de necesidad educativa en la que se va a hacer énfasis, identificando las posibles causas que permitan plantear las diversas alternativas de solución a la problemática encontrada. Ver formato 1.

A nivel nacional e internacional se han realizado diversos estudios acerca de la identificación y clasificación de necesidades educativas, dentro de los que se destacan los realizados por Zabalza<sup>7</sup> (1986) donde realiza una clasificación de las necesidades, dividiendo éstas en cinco tipos: normativa, sentida, expresadas, comparativas y prospectivas.

- Normativa: son necesidades que se establecen por medio de una ley o norma social.
- Sentida: son las necesidades subjetivas o percepciones de cada situación.
- Expresada o demandada: son las necesidades que exigen una pronta solución, es decir de exigencia generalizada y mayoritaria.
- Comparativa: son las necesidades que son derivadas de la diferencia entre dos colectivos de igualdad de condiciones en la que uno de ellos posee algo que otros no tienen.
- Prospectiva: son las necesidades que pueden derivarse o surgir en un futuro.

Entre las posibles fuentes de información para el análisis de la necesidad educativa podríamos contar los documentos de las instituciones educativas, folletos, boletines informativos, planes de estudio, estadísticas sobre pruebas de estado, libros de apuntes, entrevistas y cuestionarios.

En este capítulo se tratan las características que comprenden el Análisis de la Necesidad Educativa (*véase norma ISO/IEC 12207 Proceso de adquisición, tarea descripción de la necesidad*), las cuales se describen a continuación:

---

<sup>7</sup> Miguel Ángel Zabalza, catedrático de la Universidad de Santiago de Compostela. Doctor en psicología (1979).

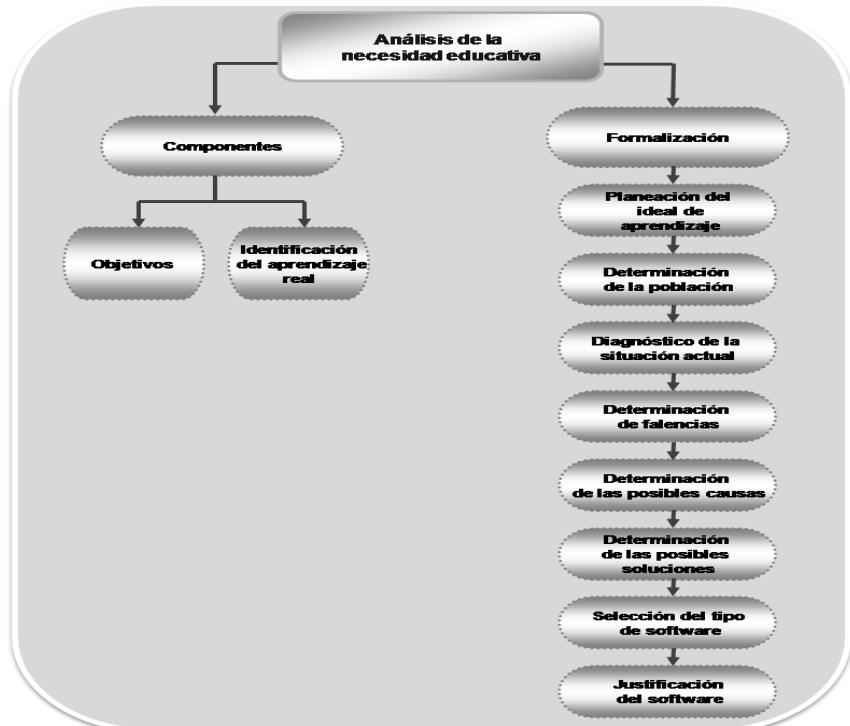


Figura 18. Análisis de la necesidad de educativa. (Modesec-ISO/IEC 12207-2008)

### **2.1.1 Componentes**

Esta dimensión abarca los objetivos y la identificación del aprendizaje real.

**2.1.1.1 Objetivos:** En este planteamiento se establecen los objetivos determinados en el plan de área, específicamente del tema seleccionado para el desarrollo de la investigación. Ya que los objetivos son catalogados como guías del estudio deben tenerse presentes para analizarlos y reformularlo si así lo requiere el estudio, esto permitirá alcanzar los resultados que se esperan al concluir la temática.

**2.1.1.2 Identificación del Aprendizaje Real:** Hace referencia al conocimiento adquirido por el estudiante en todo el proceso de aprendizaje, es decir la forma como el aprendiz asume en realidad lo aprendido en el aula de clases. Este aprendizaje se detecta a través de un diagnóstico que se realiza por medio de evaluaciones cuantitativas y cualitativas, a demás de esto es recomendable efectuar un estudio detallado de los registros académicos.

### **2.1.2 Formalización**

Esta dimensión abarca varios componentes que permiten describir la necesidad educativa, tales como: la planeación del ideal de aprendizaje, la determinación de la población, el diagnóstico de la situación actual, la determinación de falencias, de las posibles causas y las posibles soluciones, la selección del tipo de software y por último la justificación del software.

**2.1.2.1 Planeación del Ideal de aprendizaje:** Al aprendizaje se le considera como un proceso de adquisición de nuevos conocimientos, que incluyen habilidades y capacidades, las cuales se deben manifestar en un

tiempo futuro, este debe permitir la solución de problemas, incluso diferentes en su esencia a las que motivaron inicialmente el desarrollo del conocimiento, habilidad o capacidad<sup>8</sup>.

En la búsqueda del conocimiento es necesario planear alternativas de enseñanza que faciliten el aprendizaje en la población estudiantil, es por esto que en la educación actual han incursionado nuevos modelos de enseñanza que han permitido cambios en la metodología utilizada por los docentes en el aula de clases, estos cambios ayudan a que el aprendizaje en los educandos sea significativo, de tal forma que estos aprendan lo planeado por el docente.

**2.1.2.2 Determinación de la población:** La población son el conjunto de actores que se ven afectados de forma directa e indirecta por un problema de tipo educativo. Para determinar la población es necesario especificar las características a tener en cuenta para la realización del estudio, estas se identifican de la siguiente forma: la ciudad es el lugar en donde se ubica la institución seleccionada; la ubicación o contexto, se refiere a la institución en donde se presenta la problemática; el área del conocimiento es la disciplina escogida para llevar a cabo el estudio; la unidad temática se le denomina a los capítulos que identifica los contenidos que hacen parte de un área del conocimiento determinada; el grado es el nivel académico que se encuentra cursando el estudiante; el grupo es donde se presenta la delimitación del grado seleccionado y por último se debe tener en cuenta el rango de edades en donde se define la edad con que cuenta la población objeto de estudio.

**2.1.2.3 Diagnóstico de la situación actual:** En el transcurso de la actividad educativa se presentan falencias en el proceso de enseñanza-aprendizaje, que conllevan a la realización de estudios minuciosos que logren resolver a través de ayudas computacionales dichos problemas educativos, para esto es necesario realizar un diagnóstico en donde se apliquen técnicas de recolección de información que incluyan encuestas, pruebas orales o escritas, test y entrevistas, el resultado que arrojen las pruebas ejecutadas permitirán evidenciar el problema planteado.

**2.1.2.4 Determinación de falencias:** En esta sección se realiza un análisis de los resultados obtenidos del diagnóstico efectuado a la población objeto de estudio; este análisis consta del estudio específico de cada temática que incluyen la necesidad educativa que se va a trabajar. Por medio de este estudio se determinará cuales son las falencias que presentan la población con respecto a cada contenido esbozado.

**2.1.2.5 Determinación de las posibles causas:** Esta variable presenta un listado que demarca la procedencia de las dificultades encontradas en la población de estudio que demuestran las razones por las cuales existen fallas en el proceso educativo, esto conlleva a enfrentar el problema de origen que permitirá proporcionar posibles soluciones.

**2.1.2.6 Determinación de las posibles soluciones:** En la realización de un estudio educativo que se evidencie un problema, es necesario plantear alternativas de solución que ayudaran a resolver las falencias y las causas diagnosticadas en el análisis de la necesidad educativa.

**2.1.2.7 Selección del tipo de software:** Después de haber analizado todas las posibles soluciones a la necesidad educativa y de estas seleccionar la opción computacional se procede a escoger el tipo de software el cual fundamenta la forma como se desarrollará el software educativo.

**2.1.2.8 Justificación del software:** La justificación indica el por qué del tema y del problema elegido para la investigación teniendo en cuenta la importancia y utilidad que puede brindar a la comunidad la solución que

---

<sup>8</sup> ALFONSO S, Ileana; GONZÁLEZ P, Troadio. Proceso de enseñanza-aprendizaje: Algunas características y particularidades. [En línea] Disponible en: <http://www.monografias.com/trabajos7/proe/proe.shtml?relacionados>. Acceso 02 marzo de 2009

se aspira a hallar con la investigación planeada. A demás, se debe incluir en la justificación el aporte teórico y metodológico que podría ser alcanzado a favor de la disciplina de que se trata mediante la solución del problema que se pesquisa<sup>9</sup>.

En la justificación se hace referencia a las razones que explican las necesidades de crear un software educativo como alternativa de solución a las causas relevantes que se demuestran con la investigación planeada, a demás se debe enunciar los beneficios que aporta a la población la aplicación de un producto computacional como apoyo al proceso de enseñanza y aprendizaje.

A Continuación se presenta un ejemplo del formato de cómo se realiza el análisis de la necesidad educativa:

<b>ANÁLISIS DE LA NECESIDAD EDUCATIVA</b>	
<b>Tipo: sentida</b>	La necesidad a tratar es de tipología sentida, debido a que es un problema que se viene presentando en las Instituciones Educativas de Córdoba
<b>Identificación del aprendizaje ideal</b>	
Los estudiantes deben conocer e identificar: <ul style="list-style-type: none"> <li>✓ Ubicación geográfica de la cultura Zenú</li> <li>✓ Datos históricos sobre la cultura Zenú</li> <li>✓ Aspectos sociales, económicos y culturales de los Zenúes</li> </ul>	
<b>Población</b>	<b>Rango de edad:</b> 12 años en adelante
	<b>Escolaridad:</b> Grado 6º en adelante
	<b>Conocimiento que posee:</b> Deben tener conocimientos básicos de lectura y nociones básicas de historia Colombiana y de manejo del PC
	<b>Intereses y expectativas: (Población)</b> Aprender a manejar el software y que ayude a mejorar la forma de comprensión del tema.
<b>Área de formación</b>	<b>Intereses y expectativas: (Creadores)</b> Desarrollar motivación en los estudiantes hacia el software y que este de una manera más dinámica les facilite conocer parte de las culturas Precolombinas de Colombia, específicamente la Zenú.
	<b>Área del saber:</b> Ciencias sociales e Historia <b>Área de contenido:</b> Culturas Precolombinas de Colombia
<b>Estado actual</b>	Diagnóstico: Examinando algunos de los planes de clases de los docentes del área de Ciencias Sociales e Historia de las Instituciones Educativas de Montería en los grados 4 y 5, haciendo una comparación con los estándares básicos de competencias de Ciencias Sociales en grado 4 y 5 que expide el Ministerio de Educación Nacional MEN donde se afirma que el estudiante debe ” Identificar, describir y comparar algunas características sociales, políticas, económicas y culturales de las comunidades precolombinas de Colombia”; ”Reconocer la diversidad étnica y cultural de su comunidad, ciudad, país...”, pero al hacer preguntas de tipo directa y un cuestionario de diagnóstico a los estudiantes donde se le pregunta: ¿conoces algunas culturas de Colombia o de Córdoba? solo el 60 % mencionó algunas; pero ninguno señaló la cultura Zenú aún siendo de su departamento; Igualmente se le preguntó ¿Qué conoces de los Zenú? Y la mayoría los relacionó con el río Sinú de la ciudad de Montería, entre otras preguntas; Teniendo en cuenta lo anterior cabe anotar que estas temáticas se están dando de una manera muy básica y hace falta profundizar en la cultura de nuestro departamento específicamente la cultura Zenú.
<b>Necesidad</b>	La necesidad educativa está reflejada en la falta de conocimientos en lo concerniente a las culturas precolombinas de Colombia e ideas muy pobres sobre la cultura Zenú, donde los estudiantes desconocen varios aspectos de esta cultura por ejemplo la ubicación, aspectos sociales y culturales entre otros.

<sup>9</sup> Díaz, Fernando. Módulo pedagógico Invitación a la investigación científica. Editorial Louis Argel asociados. Montería. 1999.

<b>Causas</b>	<ul style="list-style-type: none"> <li>➤ Falta de profundización de los temas por parte del docente puesto que estos temas son muy extensos y tediosos para el estudiante.</li> <li>➤ Falta de herramientas didácticas donde se haga un desglose del extenso tema, para motivar al estudiante y obtener la atención adecuada, para que este tenga una mejor adquisición de conocimientos.</li> <li>➤ Falta de interés por parte de los estudiantes por apropiarse de sus propias culturas.</li> </ul>
<b>Soluciones</b>	<ul style="list-style-type: none"> <li>➤ Capacitar al docente para que use una metodología más didáctica para poder desarrollar clases de temas extensos.</li> <li>➤ Compra de materiales y herramientas pedagógicas (carteleras, videos, fotografías del tema) para que los estudiantes tengan conocimiento directo con la cultura.</li> <li>➤ Crear una metodología donde el docente utilice más herramientas didácticas que despierten la motivación y la atención del estudiante al referirse a este tema.</li> </ul>
<b>Conocimientos y habilidades que debe tener el estudiante</b>	<p><b>Preconceptos:</b> que tengan nociones básicas de historia Colombiana y conocimientos de la existencia de las culturas precolombinas.</p> <p><b>Precondiciones:</b> manejo básico del computador, habilidades de lectura y comprensión, visuales y motrices.</p>
<b>Justificación</b>	
<p>Teniendo en cuenta que la necesidad encontrada requiere de nuevas alternativas de solución, y que dentro de estas posibles soluciones esta capacitar a los docentes para el uso de nuevas metodologías y la compra de materiales y herramientas pedagógicas, para ello las instituciones gastarían tiempo y dinero con los cuales no cuenta actualmente, planteamos como la solución más viable la creación de un software educativo sobre la cultura Zenú en el cual los conocimientos faltantes en los estudiantes acerca del tema sean proporcionados de una manera más comprensible; a la vez se aprovecha los beneficios de las nuevas tecnologías; gracias a la interacción del estudiante con el software educativo se buscará lograr una mayor motivación y atención en ellos, pues podrán ir eligiendo el temática que más capte su atención para esto se utilizaran imágenes e interfaces más llamativas, saliéndose del ambiente tradicional del salón de clases.</p>	

*Formato 1. Ejemplo de Formato de Análisis de la necesidad de educativa. (Software los Zenúes - Edupmedia-2009).*

## 2.2 Planeación del proceso de producción

En todo proceso de desarrollo de software educativo se necesita planificar como quedará constituido u organizado el desarrollo de las actividades, esto ayuda a que durante el proceso se tenga un control, que permita cumplir a cabalidad lo establecido en la planificación de la producción. El conjunto de procesos integra las metas u objetivos que se pretenden alcanzar, la distribución de las funciones que cada persona tendrá que asumir para lograr los objetivos que han sido planteados y por último un cronograma de actividades. Ver formato 2 y 3

La subfase, Planeación del proceso de producción contiene aspectos importantes como por ejemplo, la organización del proceso de desarrollo de actividades para la creación del software.

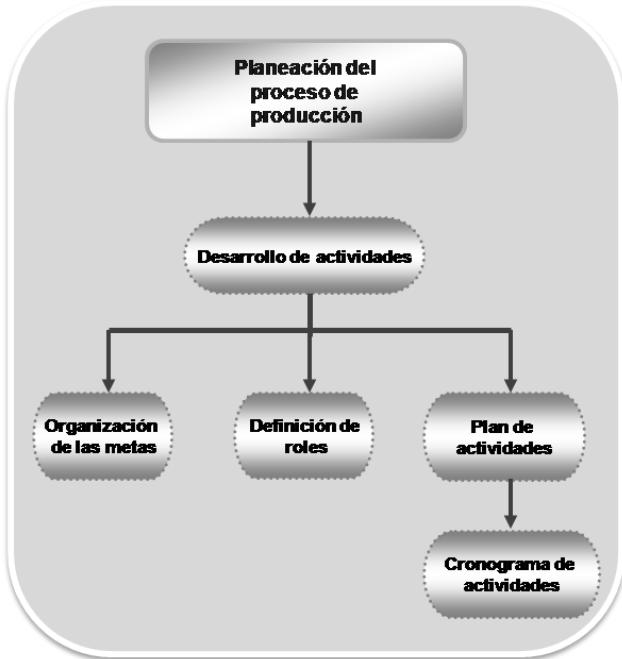


Figura 19. Planeación del proceso de producción

### 2.2.1 Desarrollo de actividades

Esta planificación bien estructurada debe tener en cuenta los siguientes aspectos:

**2.2.1.1 Organización de las Metas:** Las metas constituyen las responsabilidades o compromisos personales que deben asumir cada integrante del grupo de trabajo, con respecto a las tareas o roles asignados para la creación y desarrollo del software educativo. La organización de estas metas se describe a través de un formato de registros de tiempo, el cual establece el tiempo que invierte el grupo de trabajo para desarrollar su tarea. Este formato muestra una serie de características que se describirán a continuación: *Ver formato 2*

- **Fecha:** En este componente se detalla la fecha en que se realiza una actividad.
- **Inicio:** En este campo se anota la hora de comienzo de una actividad.
- **Fin:** En esta unidad se identificará la fecha de terminación de una actividad.
- **Tiempo de interrupción:** Se detalla la pérdida de tiempo o minutos debido a una interrupción.
- **Tiempo:** Esta sección hace referencia al tiempo dedicado a una actividad en minutos.
- **Descripción de la actividad:** Este elemento explica el nombre descriptivo de cada actividad.
- **Comentarios:** Los comentarios definen la descripción más completa de la actividad y la descripción de las interrupciones.
- **Completado:** En este campo se indica si la tarea fue terminada en esta oportunidad.
- **Unidades:** En este componente se detalla el número de unidades de una tarea finalizada.

<b>FORMULARIO DE REGISTROS DE TIEMPO</b>
--

Fecha	Hora		Tiempo de interrupción	Tiempo	Descripción de la actividad	Comentarios	Completado	Unidades
	Inicio	Fin						

Formato 2. Formulario de registros de tiempo Basado en el libro *Introducción al Proceso Software Personal* (Watts S. Humphrey)

**2.1.2 Definición de roles:** Esta fase es donde se selecciona el grupo de trabajo, esta selección depende del perfil cognoscitivo de cada persona que hará parte de esta labor, se debe tener en cuenta cual es el conocimiento en el que se profundiza el sujeto para la asignación de roles o funciones que harán parte del desarrollo de las actividades.

#### 2.2.1.3 Plan de Actividades

- **Cronograma de actividades:** El cronograma o plan de actividades cumple la función de establecer el tiempo que se tardará en desarrollar cada actividad que hacen parte del diseño y desarrollo del software. Estas actividades están dispuestas por fechas las cuales se presentan en forma cronológica, esto permite tener una organización y un control efectivo de todo el proceso. Ver formato 3, este formato varía en cuanto al tiempo porque esta variable depende del tamaño del proyecto software, es decir que cada proyecto tiene un tiempo determinado.

DESARROLLO DE ACTIVIDADES												
Actores	Actividades	MES 1		MES 2		MES 3		MES 4		MES 5		MES 6
		1	2	3	4	1	2	3	4	1	2	3

Formato 3. Formulario de desarrollo de actividades

#### 2.3 Diseño de Fines Educativos

En el proceso de enseñanza y aprendizaje intervienen diversos factores que propenden la calidad educativa, dentro de estos factores sobresalen los fines educativos que integran los objetivos de aprendizaje, los valores y las dimensiones humanas, estos constituyen los propósitos fundamentales de la enseñanza que serán el punto de partida del desarrollo de las actividades escolares. Ver formato 2.

El diseño de fines educativos comprende el *nivel cognitivo*, el cual hace referencia al tipo de conocimiento que serán trabajados por los estudiantes y se redacta normalmente en forma de objetivos, y el *nivel valorativo*, el cual hace referencia a los valores, normas y conductas que serán privilegiadas<sup>10</sup>.

Para describir los fines educativos se necesitan desarrollar categorías que fundamenten el proceso de enseñanza y aprendizaje, las cuales se detallan de la siguiente forma:

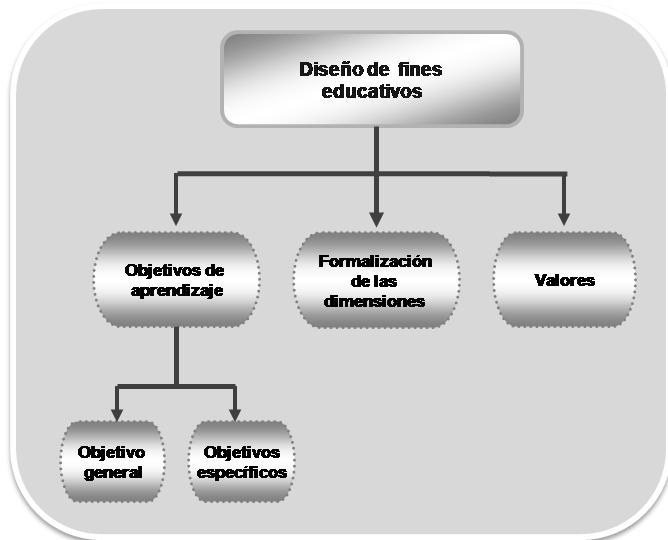


Figura 20. Diseño de fines educativos

### 2.3.1 Objetivos de aprendizaje

En esta sección se establecerán el objetivo tanto general como los objetivos específicos que se trabajarán en el transcurso de la temática seleccionada.

**2.3.1.1 Objetivo general:** Los pasos a seguir para el planteamiento del objetivo general se detallan en función del problema a estudiar, donde se debe tener en cuenta la temática propuesta para alcanzar los logros trazados y sirven como punto de referencia para determinar si lo definido se logró efectuar.

Al redactar un objetivo general es necesario tener presente que el enunciado debe ser claro y conciso, se inicia con un verbo que debe estar en infinitivo y deben revisarse durante todo el proceso de estudio.

**2.3.1.2 Objetivos específicos:** La formulación del objetivo general implica la elaboración de objetivos específicos que detallaran concretamente a través de incisos las acciones que se deben realizar a corto plazo; estos objetivos siempre deben estar relacionados con la finalidad que se pretende alcanzar.

### 2.3.2 Formalización de las dimensiones

El ser humano es un ser integral que se desenvuelve dentro de un ambiente; en él influye un sinnúmero de características biológicas, psicológicas, sociales y espirituales<sup>11</sup>.

<sup>10</sup> DE ZUBIRÁA, J. De la escuela nueva al constructivismo. Un análisis crítico. Cooperativa Editorial. Magisterio. Bogotá. 2001.

La educación es un proceso de desarrollo integral, por ende ésta debe permitir el desarrollo de las dimensiones de los estudiantes para lograr la educación integral; por tal razón es necesario incluir en la Fase I de *Diseño Educativo* la formalización de las dimensiones, donde se deben especificar el tipo de dimensiones del ser humano que serán abordadas o potencializadas por medio del software, es decir las que se evidenciarán en la población después de interactuar con el software.

### 2.3.3 Valores

Los valores humanos son aquellos bienes universales que pertenecen a nuestra naturaleza como personas y que, en cierto sentido, nos humanizan, porque mejoran nuestra condición de personas y perfeccionan nuestra naturaleza humana<sup>12</sup>.

Educar en valores es participar en un auténtico proceso de desarrollo y construcción personal. Una participación que en lenguaje educativo consiste en crear condiciones pedagógicas y sociales para que dicha construcción se lleve a cabo de una forma óptima. También es formar ciudadanas y ciudadanos auténticos que sepan asumir conscientemente los retos de la globalización y puedan comprometerse en la construcción de un mundo más justo, más inclusivo, equitativo e intercultural.<sup>13</sup>

Fomentar la educación en valores permite lograr en los estudiantes la configuración de su propia identidad, logrando así personas capaces de actuar conforme a sus convicciones, creencias, sus propias ideas y pensamientos. En esta fase se especificarán los valores que se trabajarán o se fomentarán al interactuar los estudiantes con el software.

A Continuación se presenta un ejemplo del formato de cómo se plantean los fines educativos:

DISEÑO DE FINES EDUCATIVOS	
Objetivos de aprendizaje	Objetivo general
	Diseñar e implementar un software educativo como ayuda didáctica en el área de ciencias sociales e historia para incrementar el interés y motivación de los alumnos de noveno grado en la temática del Frente Nacional.
	Objetivos específicos

<sup>11</sup> ALONSO PALACIO Luz Marina & ESCORCIA DE VÁSQUEZ Isabel. El ser humano como una totalidad. [En línea]. Disponible en: [http://www.escenf.unam.edu.ar/docs/programas/salud/Articulo\\_1SC.pdf](http://www.escenf.unam.edu.ar/docs/programas/salud/Articulo_1SC.pdf). Acceso 22 marzo de 2009.

<sup>12</sup> Valores humanos, ¿qué son?. [En línea]. Disponible en: [http://www.masalto.com/oque/template\\_oquedestacado.phtml?consecutivo=5537&subsecc=3&cat=11&subcat=31&subj=47](http://www.masalto.com/oque/template_oquedestacado.phtml?consecutivo=5537&subsecc=3&cat=11&subcat=31&subj=47) Acceso 22 marzo de 2009.

<sup>13</sup> HOYOS, Guillermo. MARTÍNEZ, Miquel & et al. ¿Qué significa educar en valores hoy?. Edición Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura (OEI) – Octaedro. 2004. [En línea]. Disponible en: <http://www.oei.es/publicaciones/educaval1.htm>. Acceso 22 marzo de 2009.

	<ul style="list-style-type: none"> <li>◆ Identificar las causas que genera el desinterés de los alumnos para el estudio del frente nacional.</li> <li>◆ Seleccionar una herramienta didáctica multimedia e interactiva para implementar la temática.</li> <li>◆ Implementar un modelo de diseño para construir la herramienta didáctica multimedia interactiva escogida</li> <li>◆ Crear la herramienta didáctica multimedia interactiva.</li> <li>◆ Implementar la herramienta didáctica interactiva en el aula de clases.</li> </ul>
<b>Dimensiones</b>	<ul style="list-style-type: none"> <li>• Capaz de identificar y comprender los antecedentes del frente nacional para que explicar las causas que dieron lugar al frente nacional</li> <li>• Capaz de explicar el impacto que tuvo el frente nacional y sus principales consecuencias.</li> </ul>
<b>Valores</b>	<p>El presente proyecto puede fortalecer:</p> <ol style="list-style-type: none"> <li>1. El trabajo grupal en algunos estudiantes.</li> <li>2. Capacidad reflexiva sobre algunas políticas colombianas</li> <li>3. Creatividad para identificar información</li> <li>4. Razonamiento lógico sobre efectos negativos de algunas políticas</li> <li>5. Contextualización y toma de decisiones antes las problemáticas sociales y políticas.</li> </ol>

Formato 4. Ejemplo de Formato Diseño de fines educativos. (Software Frente Nacional - Edupmedia-2009)

## 2.4 Diseño del Sistema de Competencias

En el contexto educativo el término competencias se extiende al conjunto de actividades que se deben realizar para enfatizar el desarrollo de potencialidades del sujeto a partir de lo que aprende en la escuela<sup>14</sup>.

El sistema de competencias tiene varias clasificaciones según diversos autores, los cuales se destacan el proyecto Eurydice<sup>15</sup> donde se formuló 8 competencias las cuáles son comunicación en la lengua materna, comunicación en la lengua extranjera, técnicas de la información y comunicación (TIC's), competencias básicas en matemáticas, ciencias y tecnología, espíritu emprendedor, competencias cívicas, capacidad de aprender, cultura general.

El proyecto DeSeCo<sup>16</sup> busca definir no todas las múltiples competencias que son necesarias para actuar en la sociedad sino aquellas que pueden considerarse básicas o esenciales, que llama **key competences** o competencias claves.

DeSeCo ha creado un marco de análisis que identifica tres categorías de competencias claves: Competencias que permiten dominar los instrumentos socioculturales necesarios para interactuar con el conocimiento, competencias que permiten interactuar en grupos heterogéneos, competencias que permiten actuar autónomamente<sup>17</sup>.

<sup>14</sup> Losada, A., Moreno, H. Competencias básicas aplicadas al aula. Ediciones SEM (Servicio educativo del magisterio). Bogotá D.C. 2003.

<sup>15</sup> Eurydice es la red europea de información y es una de las redes más desarrolladas sobre educación. Se creó en 1980 con el objetivo de favorecer el intercambio de información y de experiencias de la educación en Europa.

<sup>16</sup> DeSeCo es el nombre del Proyecto de la Organización para la Cooperación y el Desarrollo Económicos (OCDE) , OECD según su sigla ingles, encargado de definir y seleccionar las competencias consideradas esenciales para la vida de las personas y el buen funcionamiento de la sociedad.

<sup>17</sup> BRUNNER, José Joaquín. Competencias para la vida: Proyecto DeSeCo. Publicado en Diciembre 11, 2005. [En línea]. Disponible en: [http://mt.educarchile.cl/mt/jbrunner/archives/2005/12/\\_desecco\\_es\\_el\\_n.html](http://mt.educarchile.cl/mt/jbrunner/archives/2005/12/_desecco_es_el_n.html). Consultado 22 marzo de 2009.

El proyecto Alfa-Tuning<sup>18</sup> se propone alcanzar un amplio consenso a escala regional sobre la forma de entender los títulos desde el punto de vista de las actividades que los poseedores de dichos títulos estarían en capacidad de desempeñar<sup>19</sup>.

El proyecto Alfa-Tuning elaboró un total de 27 competencias genéricas acordadas para América Latina entre las que se destacan: capacidad de abstracción, análisis y síntesis; capacidad de aplicar los conocimientos en la práctica; capacidad para organizar y planificar el tiempo; conocimientos sobre el área de estudios y la profesión; responsabilidad social y compromiso ciudadano; habilidades en el uso de las tecnologías de la información y comunicación; capacidad de investigación; Capacidad para identificar, plantear y resolver problemas; entre otras<sup>20</sup>.

Ahora bien, la educación en Colombia se rige por un sistema de competencias en donde se educa al estudiante para que éste sea activo en la sociedad, es decir que sea capaz de enfrentar las distintas situaciones que se le presenten de una forma crítica y responsable, permitiendo que se desenvuelva en un espacio social, orientado por un pensamiento de carácter sistémico y tecnológico. A partir de esto es necesario incluir en la creación de software educativo especificaciones claras acerca de cómo organizar las competencias que se pretenden desarrollar con el software. En este caso en la primera fase del modelo, se introduce una sección para gestionar las competencias que se han identificado, luego de realizar el análisis de la necesidad educativa (*ver sección 2.4.1*). Estas se estructuran en un formato (*ver formato 5*) que comprende dos grandes ámbitos que son los objetivos y las normas, los cuales se describen a continuación:

---

<sup>18</sup> Proyecto alfa tuning – América Latina: carreras basadas en competencias, el objetivo general del proyecto es contribuir al desarrollo de titulaciones fácilmente comparables y comprensibles en una forma articulada en diferentes sistemas.

<sup>19</sup> Proyecto alfa tuning – América Latina: carreras basadas en competencias. [En línea]. Disponible en: [http://www.udc.es/grupos/cndeuto/docs/Sintesis\\_del\\_proyecto.pdf](http://www.udc.es/grupos/cndeuto/docs/Sintesis_del_proyecto.pdf). Consultado 22 marzo de 2009.

<sup>20</sup> BENEITONE, Pablo; et al. Reflexiones y perspectivas en la Educación Superior en América Latina. Informe final-Proyecto Tuning- América Latina 2004-2007. [En línea]. Disponible en: <http://tuning.unideusto.org/tuningal/index.php>. Consultado 22 marzo de 2009.

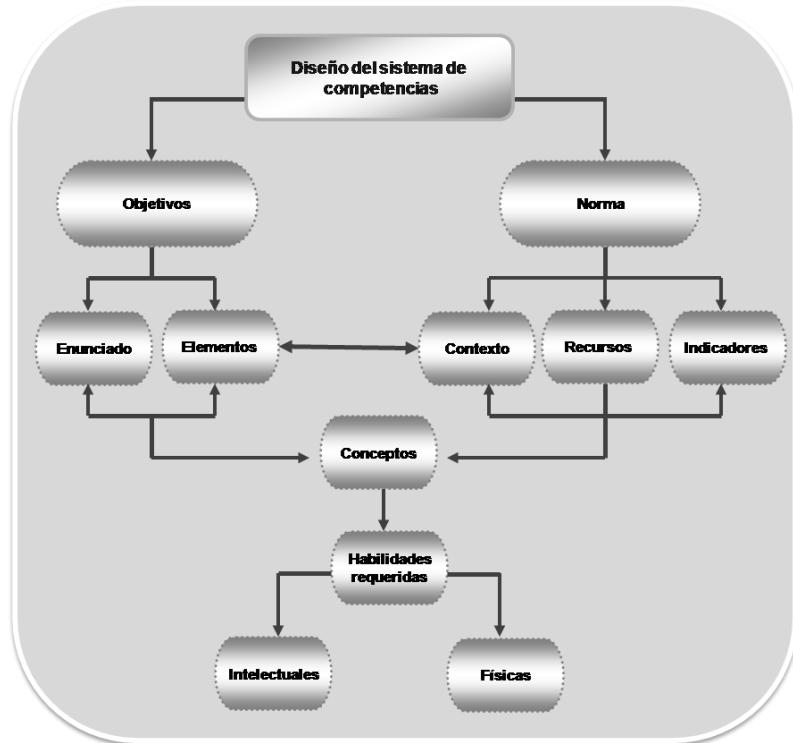


Figura 21. Diseño del sistema de competencias

### 2.4.1 *Objetivos*

Esta dimensión abarca el enunciado y los elementos de la competencia.

**2.4.1.1 Enunciado:** Los objetivos son elementos fundamentales en todo proceso debido a que estos constituyen el derrotero a seguir y admite a su vez la evaluación del estudio abordado. En esta sección se establecerá los objetivos de las competencias para ampliar las nociones sobre la forma de definir un ambiente de aprendizaje. Aquí se indica las finalidades que se pretenden lograr o adquirir por parte de los sujetos involucrados en el proceso.

**2.4.1.2 Elementos:** Para el tratamiento de las competencias se requieren una serie de actividades que dependen de cada objetivo, las cuales son enunciadas específicamente determinando lo que el estudiante debe ser capaz de realizar para cumplir el objetivo general.

### 2.4.2 *Norma*

Es la dimensión de la competencia que trata acerca del contexto, los recursos y las evidencias de conocimientos.

**2.4.2.1 Contexto:** Aquí se especifica el contexto donde se hace evidente la competencia, este puede ser un lugar físico o una situación determinada por un problema o reto que se debe solucionar.

**2.4.2.2 Recursos:** Son los recursos tangibles e intangibles que se requieren para realizar la competencia en el contexto determinado. Estos recursos se deben proveer en lo posible con el software. Así por ejemplo si se requiere un diccionario o una libreta de apuntes, el software debe proveer herramientas de diccionario y de libreta o agenda según el caso. Por tanto esta sección es el punto de partida para los elementos que debe llevar la barra de herramientas.

**2.4.2.3 Indicadores:** También llamados evidencias, son los conocimientos o actitudes que se deben ver o evaluar para saber si el estudiante o persona es competente. Por cada *elemento* de la competencia se pueden tener uno o más indicadores, por tanto esta sección será la base para la elaboración del sistema de evaluación del aprendizaje del software educativo.

### 2.4.3 Conceptos

Esta sección contiene una lista detallada de los conceptos que aparecen en los apartados anteriores del formato de competencias y que se relacionan directa o indirectamente con el ámbito de la temática del software educativo. Aquí no es necesario definir o explicar cada concepto, pues más adelante con esta lista se desarrolla el *diseño de contenidos* donde se tratan con mayor profundidad sus significados y relaciones.

### 2.4.4 Habilidades Requeridas

La realización de cualquier actividad implica ciertos conocimientos, destrezas que son necesarias para lograr su cumplimiento, esto conlleva a que los estudiantes deban poseer ciertas aptitudes que se requieren para desarrollar cada una de las competencias así:

**2.4.4.1 Intelectuales:** Las habilidades intelectuales son aquellas que se necesitan en la realización de actividades mentales. Los test de Coeficiente intelectual (CI), los test de admisión, los test de admisión para el postgrado en negocios, están diseñados para asegurarse de la habilidad intelectual de los individuos<sup>21</sup>.

Las habilidades intelectuales como la destreza numérica, la comprensión verbal, la velocidad perceptual, el razonamiento inductivo, el razonamiento deductivo, la visualización espacial y la memoria, son las aptitudes que deben poseer los estudiantes como requisito para desarrollar las actividades propuestas en cada competencia.

**2.4.4.2 Físicas:** Las habilidades físicas son aquellas que se necesitan para realizar tareas que demandan vigor, destreza manual, fortaleza y características parecidas<sup>22</sup>.

Las habilidades físicas comprenden las destrezas motoras que son fundamentales para el manejo de herramientas computacionales, que permiten a los estudiantes realizar con menor dificultad distintos tipos de actividades, en este caso los que intervienen en las competencias planteadas.

A Continuación se presenta un ejemplo del formato de cómo se diseña una competencia:

Competencia 1	Tipo Cognitiva
Objetivos	Norma
1:Enunciado	1: Contexto
Reconocer aspectos importantes de tipo geográfico, cultural, económico y político de la cultura Zenú.	Cuando se le solicite datos históricos y aspectos socio-culturales de la cultura Zenú.
	2: Recursos
	• Mapa de Colombia

<sup>21</sup> AMORÓS, Eduardo. Comportamiento organizacional: En busca del desarrollo de ventajas competitivas.

<sup>22</sup> Ibíd.

	<ul style="list-style-type: none"> <li>• Imágenes de las artesanías Zenú</li> <li>• Evolución histórica de los Zenues</li> <li>• Software educativo de la cultura Zenú</li> </ul>
<b>2: Elementos</b>	<b>3:Evidencias</b>
1- Reconoce la ubicación geográfica y datos históricos sobre la cultura Zenú	<p><b>1.1</b> Identifica geográficamente la ubicación de los Zenues.</p> <p><b>1.2</b> Identifica la evolución histórica de los Zenues</p>
2-Identificar aspectos políticos y económicos de los Zenues	<p><b>2.1</b> Reconoce la cultura Zenú y la constitución de un gobierno centralizado, se organizaron en tres grandes caciques, El cacique de Panzenú, el de Zenufana y la gran Cacica de Finzenú.</p> <p><b>2.2</b> Conocer los conocimientos de ingeniería hidráulica de los Zenues para construir y manejar las terrazas en el San Jorge.</p>
3- Identificar aspectos socio-culturales y algunos aportes de los Zenues	<p><b>3.1</b> Reconoce la vida religiosa de la cultura Zenú.</p> <p><b>3.2</b> Identificar los ritos mortuorios Zenues eran bastante simples pero esto no les restaba belleza</p> <p><b>3.3</b> Conocer la forma de obtener tan abundantes provisiones de oro.</p> <p><b>3.4</b> Conocer las técnicas utilizadas por los orfebres para la elaboración de artesanías.</p> <ul style="list-style-type: none"> <li>• De raspado, pulimento, clasificación, deshidratación, cocción y coloración de la fibra de caña flecha</li> <li>• Tallado en piedra, hueso, totumo, madera o coco</li> </ul> <p>La talabartería</p>
<b>Conceptos</b>	
<ul style="list-style-type: none"> <li>• <b>Historia:</b> Cultura, Zenues,</li> <li>• <b>Política y economía:</b> gobierno centralizado, caciques, cacique Panzenú, Zenufana y la gran Cacica de Finzenú, ingeniería hidráulica</li> <li>• <b>Socio – cultura:</b> religión, ritos, mortuorios, orfebrería, artesanías, técnica raspado, técnica deshidratación, técnica cocción, técnica coloración, caña flecha, tallado, talabartería, sombrero Vueltiao</li> </ul>	
<b>Habilidades y destrezas</b>	
<ul style="list-style-type: none"> <li>• <b>Intelectual:</b> tener manejo de diferentes conceptos relacionados con la temática, conocer historia y diferentes aspectos notables de esta cultura, manejo básico y utilización del computador.</li> <li>• <b>Social:</b> capacidad de estar en grupos y de explorar e interactuar escuchando opiniones y sugerencias.</li> <li>• <b>Físicas:</b> tener unas condiciones físicas y mentales normales.</li> </ul>	

Formato 5. Ejemplo de Formato de competencias. (Software Los Zenúes - Edupmedia-2009)

## 2.5 Diseño de Contenidos

Para la realización de un software educativo es necesario describir cada contenido que comprenda el problema estudiado, indicando la definición de cada categoría conceptual utilizada en dicho material educativo, haciendo énfasis en las características principales del conjunto de bases conceptuales.

El contenido se obtiene del listado de la sección de *conceptos* del formato de las competencias (*ver formato 3*). Si existe más de una competencia, se hace un único listado con la sumatoria de todos los conceptos y se procede a elaborar sus definiciones según las características de la población y el nivel de profundidad deseado. *Ver formato 4*.

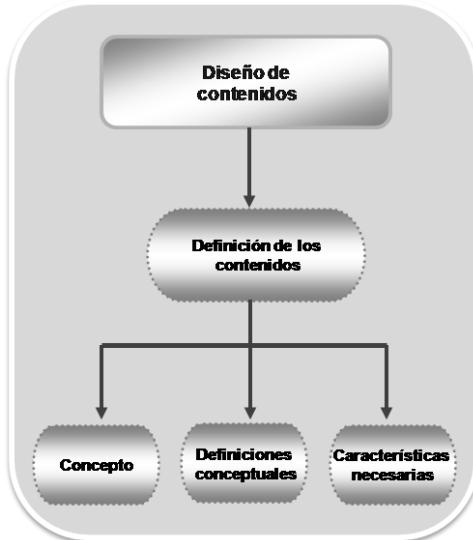


Figura 22. Diseño de contenidos

### 2.5.1 Definición de los Contenidos

**2.5.1.1 Concepto:** El concepto se define como una representación teórica que se expresa de una idea en general. En esta sección se expone específicamente los calificativos que vislumbran los contenidos que se trabajan en todo el desarrollo del software.

**2.5.1.2 Definiciones conceptuales:** La definición es la expresión de los rasgos fundamentales que constituyen el contenido del concepto<sup>23</sup>. En esta fase se presenta la declaración teórica de cada concepto.

**2.5.1.3 Características necesarias:** Esta etapa especifica lo que debe caracterizar a cada definición conceptual, es decir puntualizar los atributos representativos de cada palabra extraída de las bases conceptuales.

CONCEPTOS DE LAS COMPETENCIAS		
Conceptos	Características	Definición
Artesanías	<ul style="list-style-type: none"> <li>• Trabajos en forma manual</li> <li>• Objetos decorativos o de uso común</li> <li>• Cada pieza es única</li> <li>• Arte u obra de artesanos</li> </ul>	La artesanía comprende, básicamente, obras y trabajos realizados manualmente por una persona y con poca o nula intervención de maquinaria, habitualmente son objetos decorativos o de uso común, cada pieza es distinta de las demás. Al que se dedica a esta actividad se le denomina artesano.
Cultura	<ul style="list-style-type: none"> <li>• Conjunto de valores, creencias, ideologías, hábitos, costumbres y normas</li> <li>• Es el modo de vida de un pueblo</li> <li>• Conocimiento que conduce a un enriquecimiento personal</li> <li>• Expresiones de una sociedad</li> </ul>	Es el conjunto de todas las formas, los modelos o los patrones, explícitos o implícitos, a través de los cuales una sociedad regula el comportamiento de las personas que la conforman. Como tal incluye costumbres, prácticas, códigos, normas y reglas de la manera de ser, vestimenta, religión, rituales, normas de comportamiento y sistemas de creencias. Es toda la información y habilidades que posee el ser humano.

<sup>23</sup> BLAUBERG, I; KOPNIN, P & PANTIN, I. Diccionario filosófico marxista segunda edición. Armadillo ediciones. 1975

<b>Orfebrería</b>	<ul style="list-style-type: none"> <li>• Trabajo artístico realizado sobre utensilios o adornos de metales preciosos</li> <li>• Utilizaban principalmente la plata y el oro o una mezcla de ambos</li> <li>• Fabrica o comercia objetos de metales preciosos.</li> </ul>	Trabajo artístico realizado sobre utensilios o adornos de metales preciosos. Los metales que constituyen los objetos de orfebrería propiamente dichos son la plata y el oro o una mezcla de ambos. Con dichos metales se fabrican utensilios muy variados como vasijas, piezas de adorno, joyas, monedas, estatuas entre otras cosas.
<b>Religión</b>	<ul style="list-style-type: none"> <li>• Conjunto de creencias y prácticas relativas a cosas sagradas</li> <li>• Sistema de la actividad humana</li> <li>• Formas específicas de manifestación del fenómeno religioso</li> </ul>	Es un sistema de la actividad humana compuesto por creencias y prácticas acerca de lo considerado como divino o sagrado, personal como colectivas, de tipo existencial, moral y espiritual. Manifestación del fenómeno religioso, compartidas por los diferentes grupos humanos.

Formato 6. Vista parcial de la matriz del diseño de contenidos. (Software Los Zenúes- Edupmedia-2009)

## 2.6 Diseño Pedagógico

La pedagogía se refiere al saber o discurso sobre la educación como proceso de socialización, de adaptación. En sentido estricto, por pedagogía entendemos el saber riguroso sobre la enseñanza, que se ha venido validando y sistematizando en el siglo XX como una disciplina científica en construcción, con su campo intelectual de objetos y metodologías de investigación propios, según cada paradigma pedagógico<sup>24</sup>.

En esta sección se especifica en primera instancia el modelo pedagógico que se ha seleccionado, recordando que el modelo MODESEC no trabaja con un modelo estándar, sino a partir de la elección hecha por los expertos teniendo en cuenta diversos factores como las competencias a desarrollar y las necesidades educativas específicas y el contexto donde se aplicará. Ver formato 5.

El modelo pedagógico seleccionado se tomará como base para el diseño de las actividades de aprendizaje que tendrá el software educativo, esto se hace en forma detallada para tener una visión amplia del modelo en cuestión. Una vez hecho eso, se procede a especificar las características o aspectos de dicho modelo que serán materializadas en el software, pues generalmente debido a la gran variedad de aspectos que abarca un modelo en particular, no todos son viables de aplicar a determinadas necesidades educativas.

<sup>24</sup> FLÓREZ, Rafael. Pedagogía del Conocimiento, segunda edición. Editorial Nomos. Bogotá D.C. 2005

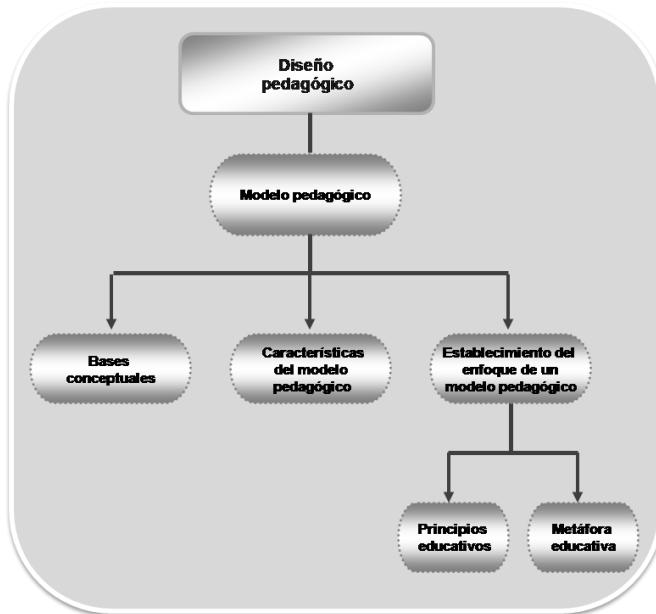


Figura 23. Diseño pedagógico

### 2.6.1 *Modelo pedagógico*

El modelo pedagógico se define como la construcción teórico formal que fundamentada científica e ideológicamente interpreta, diseña y ajusta la realidad pedagógica que responde a una necesidad histórica concreta<sup>25</sup>.

**2.6.1.1 Bases conceptuales:** Cada modelo explica desde distintas perspectivas los procesos de aprendizaje, para identificar los distintos postulados es indispensable conocer de forma detallada la definición conceptual que contienen cada modelo pedagógico

**2.6.1.2 Características del modelo pedagógico:** En todo modelo pedagógico puede encontrarse las características que lo identifican, como los fundamentos filosóficos, psicológicos, teóricos y pedagógicos en los cuales se sustentan, esto ayuda a la clarificación de la necesidad de escoger un modelo determinado. Por este motivo en este espacio se describen las particularidades del modelo seleccionado para el diseño del software.

**2.6.1.3 Establecimiento de los enfoques de un modelo pedagógico:** Las posturas o enfoques son las teorías o concepciones que toman como base distintos autores para respaldar y justificar un modelo pedagógico.

- **Principios educativos:** Las concepciones educativas o principios son los fundamentos que explican en qué consisten cada enfoque pedagógico.
- **Metáfora educativa:** El planteamiento de la metáfora educativa depende de la selección del enfoque pedagógico y de los principios educativos que este promueva, teniendo claridad sobre esos aspectos se

---

<sup>25</sup> ORTIZ OCAÑA, Alexander Luís. Modelos pedagógicos: Hacia una escuela del desarrollo integral.

procede a expresar el papel que deberá cumplir cada uno de los actores del proceso de enseñanza-aprendizaje.

<b>FORMATO 7. DISEÑO PEDAGÓGICO</b>		<b>MODELO PEDAGÓGICO CONSTRUCTIVISTA</b>
	<b>Bases conceptuales</b>	<b>Características</b>
	Es el modelo que mantiene que una persona, tanto en los aspectos cognitivos, sociales y afectivos del comportamiento, no es un mero producto del ambiente ni un simple resultado de sus disposiciones internas, sino una construcción propia que se va produciendo día a día como resultado de la interacción de estos dos factores.	<ul style="list-style-type: none"> <li>- El conocimiento no es una copia de la realidad, sino una construcción del ser humano.</li> <li>- La construcción del conocimiento se realiza con los esquemas que la persona ya posee (conocimientos previos), o sea con lo que ya construyó en su relación con el medio que lo rodea.</li> </ul>
<b>Enfoques:</b> Social		
Es un modelo de enseñanza el cual nos dice que el individuo aprende mientras convive o interactúa con el medio en donde habita y los sujetos que lo rodean los cuales ayudan a que este construya un nuevo conocimiento.		El constructivismo se plantea el desarrollo personal haciendo énfasis en la actividad mental constructiva, actividad auto constructiva del sujeto para lo cual insiste en lograr un aprendizaje significativo mediante la necesaria creación de situaciones de aprendizaje por el maestro que le permiten a los alumnos una actividad mental y también social y afectiva que favorece su desarrollo.
	<b>Principios educativos</b>	<b>Metáfora educativa</b>
	<ul style="list-style-type: none"> <li>- Se debe generar una situación problemática que genere un choque cognitivo</li> </ul> <p>Al iniciar cada tema del software el usuario se encontrara con una situación problemática que será la base y motivación para que siga explorando los temas propuestos en el software.</p> <ul style="list-style-type: none"> <li>- El docente se convierte en un guía para el estudiante en dos momentos en la hora de elaborar el producto y al momento de orientar y explicar al estudiante su utilización.</li> </ul>	<p>Estudiante</p> <ul style="list-style-type: none"> <li>- Es responsable de su aprendizaje</li> </ul> <p>El individuo interactuara de manera individual con el software y adquirirá su conocimiento de acuerdo a las actividad que va desarrollando.</p> <ul style="list-style-type: none"> <li>- Aprende cuando interactúa con otros</li> </ul> <p>Los individuos se reúnen en grupos con otros individuos a explorar y interactuar con el software educativo del frente nacional , en donde los individuos se pueden ayudar mutuamente en caso de que si uno de los Individuos no entiende o presenta problemas al momento de interactuar con el Software los demás individuos del grupo le pueden explicar o resolver la inquietud que tenga acerca de la temática o el uso de este mediante un chat incorporado en el software y así adquirir nuevos conocimientos.</p>

Formato 7. Ejemplo del diseño de contenidos. (Software Frente Nacional - Edupmedia-2009)

## 2.7 Diseño de Aprendizaje

Una vez realizado el *diseño pedagógico*, se inicia el diseño de una *estrategia de aprendizaje*, cuyo propósito es otorgar a los estudiantes estrategias efectivas para el mejor desempeño en áreas y contenidos específicos. La utilización de distintas formas de aprendizaje dependerá de las actividades efectuadas y de las características cognitivas de los estudiantes.

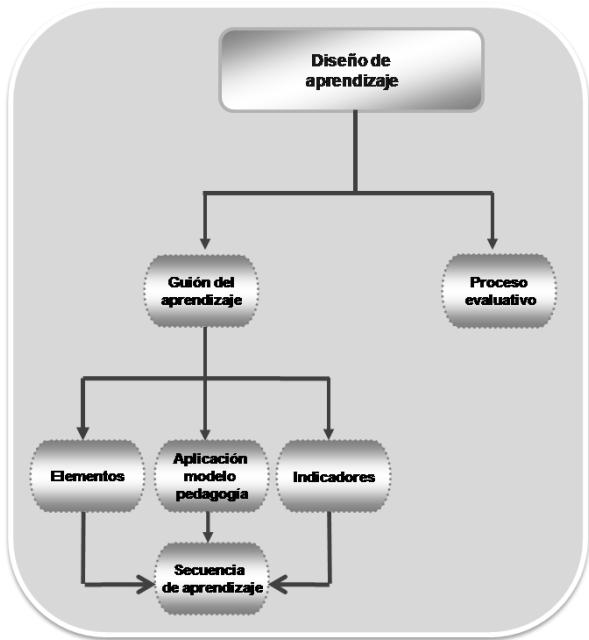


Figura 24. Diseño de aprendizaje

### 2.7.1 Guión del aprendizaje

La finalidad del guión de aprendizaje es lograr que los estudiantes realicen las actividades descritas en los elementos bajo las directrices especificadas en las características del modelo pedagógico. Ver formato 8

Para realizar el guión del aprendizaje se lleva a cabo el siguiente proceso:

**2.7.1.1 Elementos:** En este componente se hace la descripción de cada elemento de las competencias seleccionados (trabajado en la sección *diseño de competencias*, ver formato 5).

**2.7.1.2 Aplicación modelo pedagogía:** Como complemento al modelo pedagógico (trabajado en la sección de *diseño pedagógico*, ver formato 7), en esta fase se describe las características esenciales del enfoque del modelo seleccionado en el diseño pedagógico. Estas se presentan como fundamento del diseño del software educativo, debido a que a partir de las características surge el desarrollo de los elementos de las competencias. Las estrategias por modelo se identifican con un nombre que debe relacionarse con los principios dispuestos en el enfoque preseleccionado.

**2.7.1.3 Indicadores:** Para la descripción de un elemento es necesario identificar los conocimientos que ostentan cada estudiante, para ello se toman los indicadores o evidencias (*trabajadas* en la sección *diseño de competencias*, ver formato 5), esto permitirá la comparación del indicador de cada competencia con los logros adquiridos por el educando.

**2.7.1.4 Secuencia de aprendizaje:** Para el desarrollo de esta sección se necesita de la realización de las tres secciones que hacen parte del guión del aprendizaje que fueron descritas anteriormente, después de tener esos tres datos se diseña una secuencia de aprendizaje cuya finalidad es que los estudiantes realicen las actividades descritas en los elementos bajo las directrices especificadas en las características del modelo pedagógico. La secuencia debe evidenciar los aspectos relacionados en los indicadores, si dicha secuencia cumple con los aspectos mencionados, puede ser considerada como adecuada para continuar el proceso de diseño.

COMPETENCIA#1
---------------

Reconocer algunas de las políticas que precedieron al frente nacional que tuvo lugar en Colombia durante el siglo XX.		
Elementos	Aplicación modelo pedagógico	Indicadores
1. Identificar algunos acuerdos políticos bipartidistas anteriores al Frente Nacional	<p>Se debe generar una situación problemática que conlleve a un choque cognitivo.</p> <p>El aprendizaje es responsabilidad del estudiante.</p>	<p>1.1 Reconoce el acuerdo Unión nacional</p> <p>1.2 Reconoce el acuerdo Golpe de Opinión</p> <p>1.3 Reconoce el acuerdo Frente Civil</p>
<b>Secuencia de aprendizaje</b>		
<b>Objetivo:</b> Identificar coalición de partidos políticos que se dieron en Colombia antes del Frente Nacional		<b>BARRA DE RECURSOS</b>
<b>Problema:</b> Michel Brando es un profesor de Sociales e Historia de Estados Unidos.  Michel ha decidido realizar una pasantía en Colombia, aquí la embajada de nuestro país lo ubicó en una Institución Educativa del departamento de Córdoba y se le asignó el grado noveno de dicha Institución.  El rector de plantel le ha dicho a Michel que la temática que le corresponde enseñar es la que registra sobre la historia política de Colombia donde se presentaron alianzas entre partidos políticos para apoyar o derrocar gobiernos existentes desde 1954.  Michel no tiene muchos conocimientos sobre la historia política de nuestro país por pertenecer a otro, por lo que decide buscarse a ti para que lo asesores y le indique cuales son los temas más importantes que debe enseñar.		<b>Navegación: Frente Nacional</b> <ul style="list-style-type: none"> <li>- Acuerdos bipartidistas</li> <li>- Partidos políticos</li> <li>- Pactos y acuerdos</li> <li>- Grupos guerrilleros</li> <li>- Gobernantes</li> </ul>
<b>Estrategias:</b> Para poder asesorar a Michel debes tener en cuenta los principales acuerdos y gobiernos de coalición.		<b>Documentación:</b> <ul style="list-style-type: none"> <li>- Pdf</li> <li>- Fotos</li> <li>- Videos</li> </ul>
		<b>Comunicación:</b> <ul style="list-style-type: none"> <li>- Correo</li> <li>- Chat</li> </ul>

Formato 8. Ejemplo resumido del diseño de la secuencia de aprendizaje basado en competencias. (Software Frente Nacional - Edupmedia-2009)

### 2.7.2 Proceso evaluativo

La evaluación es el proceso que permite establecer el estado en el que se han alcanzado los objetivos, el objetivo de una evaluación es formar y emitir un juicio de valor con respecto a una situación, dependiendo de la evaluación se toman las decisiones para corregir y mejorar el medio evaluado. El proceso evaluativo debe estar acorde con el elemento de aprendizaje y el contexto del mismo.

En este proceso se debe evaluar los indicadores de las competencias (*ver formato 5*), utilizando un criterio de evaluación en donde se especifique el método que se utiliza para medirla, además se muestran los ejercicios

que se predeterminaron para que resuelva el estudiante. Estas actividades deben ser descritas detalladamente como aparecen en cada escenario. Ver formato 9

PROCESO EVALUATIVO			
Competencia n° 2			
Elementos n° 1	Indicadores n° 1.1	Criterio	Actividad n° 1.1 1.10
Identifica y escribe los números en idioma Mandarín.	1.1 Escribe 一 y pronuncia yī (1). <b>Indicadores n° 1.2</b> 1.2 Escribe 二 y pronuncia èr (2). <b>Indicadores n° 1.3</b> 1.3 Escribe 三 y pronuncia sān (3). <b>Indicadores n° 1.4</b> 1.4 Escribe 四 y pronuncia sì (4). <b>Indicadores n° 1.5</b> 1.5 Escribe 五 y pronuncia wǔ (5). <b>Indicadores n° 1.6</b> 1.6 Escribe 六 y pronuncia shí (10). <b>Indicadores n° 1.7</b> 1.7 Escribe 二十 y pronuncia èr shí (20). <b>Indicadores n° 1.8</b> 1.8 Escribe 三十 y pronuncia sān shí (30). <b>Indicadores n° 1.9</b> 1.9 Escribe 四十 y pronuncia sì shí (40). <b>Indicadores n° 1.10</b> 1.10 Escribe 九十 y pronuncia jiǔ shí (90).	El programa contará todos los intentos que realice el usuario tanto los acertados como los fallidos.  Cuando el usuario seleccione el número, inmediatamente el programa lanzará un mensaje diciendo: si lo ha hecho bien o si tiene que volver a intentarlo.	<b>La Identificación de los números:</b> consta de un espacio para el pictograma, un lugar para mostrar los resultados del proceso de evaluación (aciertos y fallas) y un menú Infinito con los números del 1 al 9.  El objetivo principalmente es escoger del Menú infinito el número correspondiente al Pictograma que se muestra.

Formato 9. Ejemplo del diseño proceso evaluativo. (Software Learning Mandarín - Edupmedia-2008)

## **PUNTOS CLAVE**

- Como necesidad educativa se entiende a la discrepancia entre estado educativo ideal (deber ser) y otro existente (realidad) y pueden ser: normativas, sentidas, expresada, comparativas y prospectivas.
- El aprendizaje real hace referencia al conocimiento adquirido por el estudiante en todo el proceso de aprendizaje, es decir la forma como el aprendiz asume en realidad lo aprendido en el aula de clases.
- La población es el conjunto de actores que se ven afectados de forma directa e indirecta por un problema de tipo educativo.
- El cronograma o plan de actividades cumple la función de establecer el tiempo que se tardará en desarrollar cada actividad que hacen parte del diseño y desarrollo del software
- El diseño de los fines educativos comprende el nivel cognitivo, el cual hace referencia al tipo de conocimiento que serán trabajados por los estudiantes y se redacta normalmente en forma de objetivos, y el nivel valorativo, el cual hace referencia a los valores, normas y conductas que serán privilegiadas.
- En el contexto educativo el término competencias se extiende al conjunto de actividades que se deben realizar para enfatizar el desarrollo de potencialidades del sujeto a partir de lo que aprende en la escuela.
- Se selecciona un modelo pedagógico teniendo en cuenta diversos factores como las competencias a desarrollar y las necesidades educativas específicas y el contexto donde se aplicará.
- El modelo pedagógico seleccionado se tomará como base para el diseño de las actividades de aprendizaje. Una vez hecho eso, se procede a especificar las características o aspectos de dicho modelo que serán materializadas en el software.
- Una vez realizado el diseño pedagógico, se inicia el diseño de una estrategia de aprendizaje, cuyo propósito es otorgar a los estudiantes estrategias efectivas para el mejor desempeño en áreas y contenidos específicos.

## **REFERENCIAS**

- ALFONSO S, Ileana; GONZÁLEZ P, Troadio. Proceso de enseñanza-aprendizaje: Algunas características y particularidades. [En línea]. Disponible en: <http://www.monografias.com/trabajos7/proe/proe.shtml?relacionados>. Acceso 02 marzo de 2009.
- ALONSO PALACIO, Luz Marina & ESCORCIA DE VÁSQUEZ, Isabel. El ser humano como una totalidad. [En línea]. Disponible en: [http://www.escenf.unam.edu.ar/docs/programas/salud/Articulo\\_1SC.pdf](http://www.escenf.unam.edu.ar/docs/programas/salud/Articulo_1SC.pdf). Acceso 22 marzo de 2009.
- AMORÓS, Eduardo. Comportamiento organizacional: En busca del desarrollo de ventajas competitivas.
- BENEITONE, Pablo; et al. Reflexiones y perspectivas en la Educación Superior en América Latina. Informe final-Proyecto Tuning- América Latina 2004-2007. [En línea]. Disponible en: <http://tuning.unideusto.org/tuningal/index.php>. Consultado 22 marzo de 2009.
- BLAUBERG, I; KOPNIN, P & PANTIN, I. Diccionario filosófico marxista segunda edición. Armadillo ediciones. 1975
- BRUNNER, José Joaquín. Competencias para la vida: Proyecto DeSeCo. Publicado en Diciembre 11, 2005. [En línea]. Disponible en: [http://mt.educarchile.cl/mt/jjbrunner/archives/2005/12/\\_desecho\\_el\\_n.html](http://mt.educarchile.cl/mt/jjbrunner/archives/2005/12/_desecho_el_n.html). Consultado 22 marzo de 2009.
- DE ZUBIRÍA, J. De la escuela nueva al constructivismo. Un análisis crítico. Cooperativa Editorial. Magisterio. Bogotá. 2001.
- DÍAZ, Fernando. Módulo pedagógico Invitación a la investigación científica. Editorial Louis Argel asociados. Montería. 1999.
- FLÓREZ, Rafael. Pedagogía del Conocimiento, segunda edición. Editorial Nomos. Bogotá D.C. 2005
- HOYOS, Guillermo. MARTÍNEZ, Miquel & et al. ¿Qué significa educar en valores hoy?. Edición Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura (OEI) - Octaedro. 2004. [En línea]. Disponible en: <http://www.oei.es/publicaciones/educaval1.htm>. Acceso 22 marzo de 2009.

- KAUFMAN, R.A. *Needs Assessment: What It Is How to Do It*. University Consortium on Instructional Development and Technology: San Diego, Cal, 1976.
- LOZADA A., Moreno H. Competencias básicas aplicadas al aula. Ediciones SEM (Servicio educativo del magisterio). Bogotá D.C. 2003.
- ORTIZ OCAÑA, Alexander Luís. Modelos pedagógicos: Hacia una escuela del desarrollo integral.
- Proyecto alfa tuning - América Latina: carreras basadas en competencias. [En línea]. Disponible en: [http://www.udc.es/grupos/cndeuto/docs/Sintesis\\_del\\_proyecto.pdf](http://www.udc.es/grupos/cndeuto/docs/Sintesis_del_proyecto.pdf). Consultado 22 marzo de 2009.
- Valores humanos, ¿qué son?. [En línea]. Disponible en: [http://www.masalto.com/oque/template\\_oquedestacado.phtml?consecutivo=5537&subsecc=3&cat=11&subcat=31&subj=47](http://www.masalto.com/oque/template_oquedestacado.phtml?consecutivo=5537&subsecc=3&cat=11&subcat=31&subj=47) Acceso 22 marzo de 2009.

## **CAPÍTULO III**

## **DISEÑO MULTIMEDIAL**

### **3. FASE II: DISEÑO MULTIMEDIAL**

A partir de los resultados obtenidos de la fase del diseño educativo se establece la fase número II denominada diseño multimedial. Esta fase se encarga de la descripción de las etapas que permitirán conocer en conjunto la estructura que se va a utilizar en el software, utilizando representaciones gráficas y recursos multimedia que mostrarán el diseño de cada interfaz.



Figura 26. Fase II. Diseño multimedial

#### **3.1 Diagrama de contenidos**

La funcionalidad de los diagramas es representar gráficamente la información contenida en un estudio temático de tal forma que se hace necesario complementar el diseño de un software con un esquema que represente los contenidos que se encuentran en la fase del diseño de contenidos. Esta representación se puede esquematizar de distintas formas ya sea por medio de mentefactos, mapas conceptuales y mapas mentales.

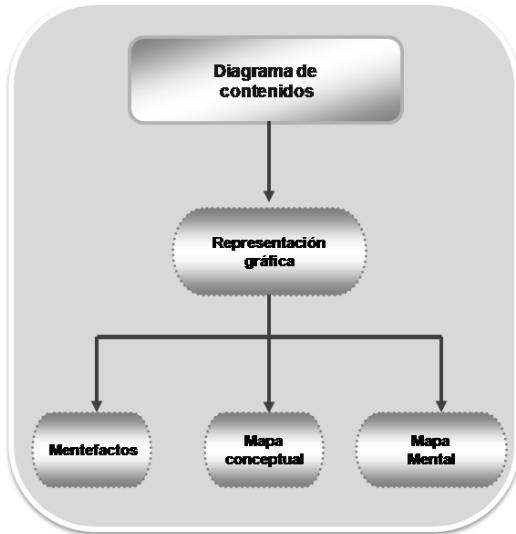


Figura 27. Diagrama de contenidos

##### **3.1.1 Representación Gráfica**

El diagrama de contenidos proporciona una forma diferente de trabajar los contenidos, estas opciones de representación pueden ser de tres tipos:

**3.1.1.1 Mentefactos:** mente= cerebro factos= hechos por tanto los mentefactos, son formas gráficas para representar las diferentes modalidades de pensamientos y valores humanos. Los mentefactos definen cómo existen y se representan los instrumentos de conocimiento y sus operaciones intelectuales mediante conceptos como la supraordinada que es la generalidad, las isoordinadas que serían las características, las infraordinadas que son las distintas formas de presentación, sin ser excluidas<sup>26</sup>.

**3.1.1.2 Mapa conceptual:** Un mapa conceptual es una herramienta para la organización y representación del conocimiento. Su objetivo es representar relaciones entre conceptos en forma de proposiciones. Los conceptos están incluidos en cajas o círculos, mientras que las relaciones entre ellos se explicitan mediante líneas que unen sus cajas respectivas. Las líneas, a su vez, tienen palabras asociadas que describen cuál es la naturaleza de la relación que liga los conceptos. Sirve para dar en forma resumida y más atractiva un texto<sup>27</sup>.

**3.1.1.3 Mapa mental:** Un mapa mental es un diagrama usado para representar las palabras, ideas, tareas, u otros conceptos ligados y dispuestos radialmente alrededor de una palabra clave o de una idea central. Se utiliza para la generación, visualización, estructura, y clasificación taxonómica de las ideas, y como ayuda interna para el estudio, organización, solución de problemas, toma de decisiones y escritura. Es un diagrama de representación semántica de las conexiones entre las porciones de información<sup>28</sup>. (Ver gráfico 1)

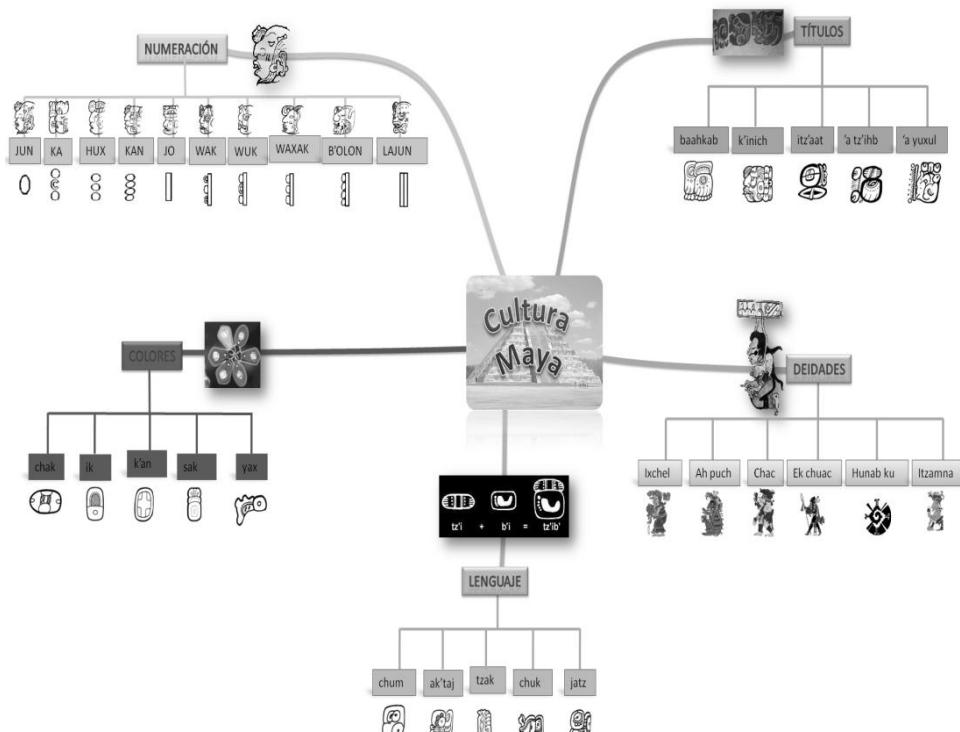


Gráfico 1. Ejemplo de un mapa utilizado como diagrama de contenidos. (Software Lenguaje Maya- Edupmedia 2008)

## 3.2 Guión técnico multimedial

<sup>26</sup> <http://www.psicopedagogia.com/definicion/mentefacto>.

<sup>27</sup> [http://es.wikipedia.org/wiki/Mapa\\_conceptual](http://es.wikipedia.org/wiki/Mapa_conceptual).

<sup>28</sup> [http://es.wikipedia.org/wiki/Mapa\\_mental](http://es.wikipedia.org/wiki/Mapa_mental).

El guión técnico multimedial es un escrito que contiene detalladamente que es lo que queremos mostrar en la pantalla de nuestro computador. En el guión se escribe como va a ser cada “pantallazo”, eso es lo que vamos a ver cada vez que interactuemos con el entorno. Debemos escribir todo, como van a ser los fondos, botones, sonidos, fotografías, colores, tipo y color de letra y donde va a estar situado todo<sup>29</sup>.

En un guión deben quedar definidos los siguientes aspectos: La interfaz de pantalla y el sistema de navegación. Todos los elementos que se van a incluir en la aplicación, su punto de aparición, su forma de entrada, su papel y acción en la aplicación, su interactividad con el usuario y su relación con otros elementos<sup>30</sup>.

La fase del guión técnico multimedial es muy importante, debido a que permite visualizar de una forma detallada la conformación de los diferentes componentes de las interfaces gráficas de usuario. En esta fase es donde se recolecta, describe, produce todos los recursos multimedia, los textos que se van utilizar en el diseño de cada una de las ventanas.

El guión técnico multimedial (*véase norma ISO/IEC 12207 proceso de desarrollo, actividad diseño de la arquitectura del sistema y del software*) es una subfase donde se especifican cada uno de los detalles requeridos en el diseño de las ventanas, se debe incluir las siguientes categorías:

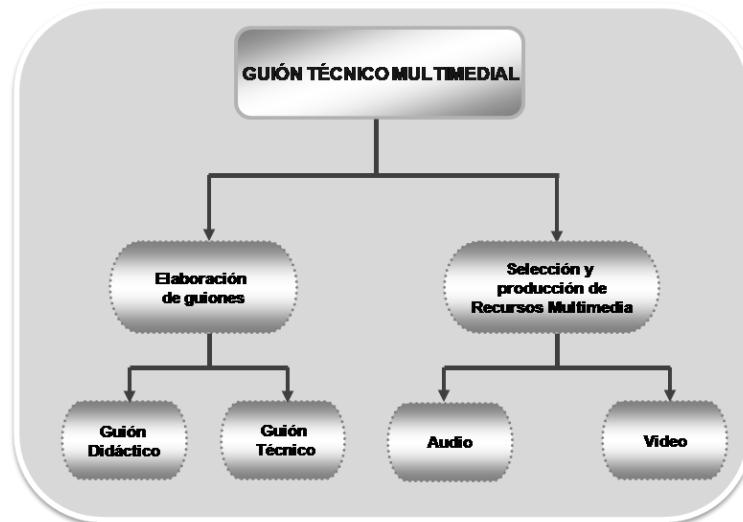


Figura 28. Guión técnico multimedial

### 3.2.2 Elaboración de Guiones

**3.2.2.1 Guión didáctico:** Este guión tiene como objetivo trasmisitir detalladamente la información o en este caso la precisión del tipo de interacción que se pretende potenciar con los alumnos a partir de la construcción de esta didáctica propiamente tal<sup>31</sup>. (Ver formato 10)

Para realizar un guión didáctico debemos tener varias cosas claras:

- ¿Para qué se va a realizar?

<sup>29</sup> <http://nodocreativo.blogspot.com/2008/04/cmo-realizar-un-guin-multimedia-parte-i.html>.

<sup>30</sup> Especificación del proyecto multimedia: Guión multimedia. Disponible en:

[www.dsic.upv.es/~fgarcia/imm/Traspas/Especificaci%20n%20del%20proyecto%20multimedia.ppt](http://www.dsic.upv.es/~fgarcia/imm/Traspas/Especificaci%20n%20del%20proyecto%20multimedia.ppt) -

<sup>31</sup> PEREZ, Felipe. Guión didáctico. <http://laxantesaboramora.oblog.com/guion-didactico-108808>

- ¿Cuál va ser su función?
- ¿A quién va dirigido?
- ¿Cómo se puede hacer más interactivo?
- Tener un tema claro.
- Una sinopsis de lo que se quiere hacer<sup>32</sup>.

En un guión didáctico se presentan las siguientes características:

- **Título:** En el título se menciona el nombre del tema que se selecciono para resolver el problema educativo.
- **Sinopsis de la temática:** En este planteamiento se describe de forma superficial las características de la temática que se trabaja en todo el proceso de desarrollo del software educativo.
- **Finalidad educativa:** En esta sección se muestra las dimensiones, competencias, habilidades y valores que se desarrollarán en los estudiantes por medio del estudio realizado a la temática seleccionada.
- **Objetivos didácticos:** Al escribir los objetivos se debe tener en cuenta lo que se quiere lograr con el desarrollo de la temática, además es importante tener claridad al formular los objetivos ya que de esto depende el cumplimiento de las metas trazadas en el diseño del software.
- **Contextos valorativos:** Los contextos valorativos describen los valores que representan al contenido abordado en el estudio de la temática que se está implementando. La finalidad de esta etapa es que los estudiantes asimilen y adquieran los valores seleccionados para tal fin.
- **Características de la población objetivo:** Para la realización de un guión didáctico se deben tener en cuenta el rango de edades, las características psicológicas, el nivel académico de la población seleccionada para el estudio y la unidad temática.

GUIÓN DIDÁCTICO	
Título	
Sinopsis de la temática	
Finalidad educativa	
Objetivos didácticos	
Contextos valorativos	
<b>Características de la población objetivo</b>	
Rango de edades	
Características psicológicas	
Nivel académico	
Unidad temática	

Formato 10. Guión didáctico

**3.2.2.2 Guión técnico:** El guión técnico 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.

---

<sup>32</sup> Ibíd., Pérez.

Los guiones técnicos la mayoría de las veces se hacen en cuadros o tablas para mejor comprensión y organización<sup>33</sup>. (*Ver formato 11*)

El guión técnico o storyboard es la sucesión de tarjetas con el diseño de cada una de las pantallas en las que vamos a desarrollar el guión literario. El storyboard es la visualización de los contenidos y su organización conceptual<sup>34</sup>.

- **Titulo de la ventana:** Se necesita que las interfaces sean identificadas con un nombre para distinguirse de las demás. Cada ventana debe ser nombrada con un título distinto para evitar posibles confusiones al momento de agregarlas en el guión técnico.
- **Descripción:** En esta sección se hace una explicación detallada sobre la forma cómo van a ser, ya sea el texto, imágenes o sonidos de la ventana.

Los textos son elementos esenciales en el desarrollo de un software debido a que estos pueden captar la atención de los usuarios si se utilizan de una forma llamativa. En este elemento se describen la fuente, tipo, color y tamaño de los textos. Las imágenes pueden ser el fondo de la interfaz, las fotografías, los gráficos, los botones que se utilizan en cada pantalla. El audio está compuesto por los sonidos que se utilizaran en cada interfaz gráfica. Si la ventana tiene música, los botones tienen sonido, si se requiere de la voz en off.

- **Formato/fuente:** El formato de un archivo es la extensión del archivo, es decir el tipo de archivo que se utilizará en la ventana por ejemplo: los textos pueden ser .txt, .doc, entre otros. La fuente hace referencia de dónde fue obtenido el archivo, ya sea el texto, las imágenes y el sonido.
- **Acción:** Aquí se escriben las acciones que tendrá cada pantallazo, por ejemplo, si vamos a guardar cada usuario que ingresó, la interactividad de los botones, si apagamos o ponemos la música o si nos va a llevar a otro pantallazo, si hay una navegación con botones o áreas sensibles<sup>35</sup>.

Se detallan cada una de las acciones que tendrán los textos, las imágenes y los sonidos de cada interfaz de usuario.

- **Evento:** los eventos están relacionados con la acción, es decir para lograr que se den las acciones se debe realizar un evento por ejemplo, si la acción es que el texto cambie de color, el evento relacionado podría ser que al pasar el mouse por el texto se ejecute la acción.

GUIÓN TÉCNICO				
Titulo de la Ventana	Descripción	Formato / fuente	Acción	Evento
Texto				
Imagen				
Sonido				

*Formato 11. Guión técnico*<sup>36</sup>.

<sup>33</sup> <http://www.monografias.com/trabajos908/guion-didactico-multimedia/guion-didactico-multimedia2.shtml>

<sup>34</sup> Multimedia 2. <http://www.slideshare.net/raymarq/multimedia-2-presentation>

<sup>35</sup> <http://www.monografias.com/trabajos908/guion-didactico-multimedia/guion-didactico-multimedia2.shtml>.

### **3.2.3 Selección y producción de recursos multimedia**

Los materiales multimedia deben utilizarse cuando hagan alguna aportación relevante a los procesos de enseñanza y aprendizaje. Su uso eficiente siempre estará supeditado a la existencia de una necesidad educativa que razonablemente pueda satisfacer<sup>37</sup>.

En la realización de un software educativo es vital seleccionar o producir recursos multimedia como el audio y video para lograr la interactividad del material educativo y captar la atención de los usuarios, permitiendo que estos obtengan mayor comprensión de los contenidos utilizados en el software. En esta sección se recolectara todo el audio y los videos necesarios para el desarrollo de las actividades, esto se hará como creación original o tomada de fuentes de información.

**3.2.3.1 Audio:** El audio actúa como complemento al diseño de las interfaces ya que a través de este se puede evocar imágenes, contextualizar los contenidos, además el usuario comprende mejor los mensajes emitidos en el software. Los distintos tipos de audio que se pueden utilizar en el diseño de un material computacional se identifican por la utilización de palabras sobre las imágenes, música, efectos sonoros, narración y el silencio.

**3.2.3.2 Video:** El video tiene como función principal respaldar los contenidos, las actividades que se presentan en cada interfaz. El video se caracteriza por las imágenes en movimiento y los efectos visuales.

## **3.3 Diseño del ambiente de aprendizaje**

Esta etapa comprende el conjunto de directrices que permitirán una comunicación directa entre el usuario y la computadora, ya que se presentará visiblemente el diseño de cada una de las interfaces gráficas de usuario. Estas directrices se identifican secuencialmente iniciando con el diseño de la ventana estándar para seguir con cada ventana o interfaz y para finalizar con el diseño del mapa de navegación.

El diseño de ambiente de aprendizaje (*véase norma ISO/IEC 12207 proceso de desarrollo, actividad diseño de la arquitectura del sistema y del software*) se dividen en dos componentes principales como son: el diseño de interfaces y el diseño de mapa de navegación.

---

<sup>36</sup> Diseñado por Julio Rangel Vellojín. Especialista en informática y multimedios. Docente del departamento de Informática de la Facultad de Educación y Ciencias Humanas.

<sup>37</sup> <http://dewey.uab.es/pmarques/interven.htm>

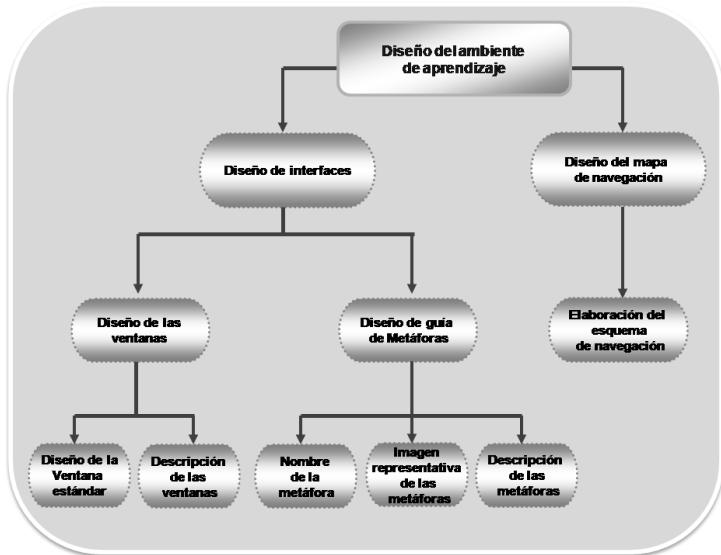


Figura 29. Diseño del ambiente aprendizaje

### 3.3.1 Diseño de interfaces

**3.3.1.1 Diseño de las ventanas:** Las ventanas o interfaces son la representación gráfica que concederá al usuario la visualización de las herramientas multimediales, los contenidos, las actividades y las evaluaciones que pueda tener el software educativo. Para diseñar las ventanas es necesario tener en cuenta el tipo de usuario a quien va dirigido el material computacional, a demás para su diseño y creación es importante tomar como base el tipo de software seleccionado en la fase del diseño educativo. (Ver formato 12)

- **Diseño de la ventana estándar:** Es la ventana principal que servirá de modelo para el diseño de las demás interfaces. La ventana estándar se diseña por secciones que deben ser explicadas detalladamente para precisar la división de la ventana.

DESCRIPCIÓN DE LA VENTANA ESTÁNDAR
<i>Título de la ventana: Maya's 2</i>

<b>Descripción</b>
Es la primera ventana de interacción que tiene el usuario con el sitio y es aquí donde puede visualizar los temas que ofrece la aplicación maya e ir a la ayuda para un mejor desenvolvimiento del este con la misma.

*Formato 12. Ejemplo del diseño de interfaces: Descripción de la ventana estándar. (Software Lenguaje Maya - Edupmedia-2008)*

- **Descripción de las ventanas:** Todas las ventanas se deben describir teniendo en cuenta el diseño y la organización dispuesta para su presentación, identificándolas en distintas categorías tales como: la imagen, el nombre de cada ventana, el texto, el audio, videos, animación y por último las acciones. Estas categorías darán una mejor explicación del contenido que posee cada una de las interfaces.

DESCRIPCIÓN DE LAS VENTANAS						
<b>Ventana: 01 principal</b>						
<i>Texto</i>	<i>Imagen</i>	<i>Audio</i>	<i>Videos</i>	<i>Animación</i>	<i>Acciones</i>	
						

-Mayensesoft		Ambiente	No	Movimientos dinámicos, realizados con herramienta flash 8 de formato swf.	Animación en línea de tiempo
- Menú		Ambiente	No	Movimientos dinámicos, realizados con herramienta flash 8 de formato swf	Al hacer Clic en cualquiera de los iconos del menú se efectúa una acción del vínculo al contenido de la aplicación.

Formato 13. Ejemplo del diseño de interfaces: Descripción de las ventanas. (Software Lenguaje Maya - Edupmedia-2008)

### 3.3.2 Diseño de Guía de Metáforas

Una metáfora es una figura retórica consistente en trasladar el significado de una palabra en otro figurado como en virtud de una comparación, normalmente se utilizan metáforas para la presentación de las actividades, una metáfora debe ir asociada al contexto de la población y el ambiente del aprendizaje. (Ver formato 14)

**3.3.2.1 Nombre de la metáfora:** En esta sección se especifica el nombre o calificativo que se designa a cada ícono o imagen representada en el diseño de las interfaces.

**3.3.2.2 Imagen representativa de las metáforas:** Se le denomina imagen a la representación gráfica con que se simboliza a los íconos que se presentan en el diseño de las ventanas del software.

**3.3.2.3 Descripción de las metáforas:** El significado o descripción son las funciones que representan o indican cada una de las imágenes descritas en el diseño de guías de metáforas.

DISEÑO DE GUÍA DE METÁFORAS		
Nombre	Imagen	Descripción
Ayuda/Help		Botón que indica el archivo donde se visualiza la ayuda.
Menú		Botón que permite regresar al menú principal y escoger a que aplicación se desea acceder.
Cerrar		Botón que cierra la aplicación que se esté observando.
Números/Number		Botón que permite acceder a la aplicación de los números de 1 a 10 en español y maya y su simbología.
Religión / Religión		Botón que permite conocer las diferentes deidades mayas.

Lenguaje/ Language		Botón que muestra algunos verbos del lenguaje maya.
Títulos / Holding		Botón que muestra los títulos principales de la cultura maya.
Vocales / Vowels		Botón que permite conocer las vocales en maya y su representación.
Colores/ Colors		Botón que muestra los colores básicos en maya.
Diccionario/ Dictionary		Botón que muestra algunas palabras en maya.

Formato 14. Ejemplo del diseño de guía de metáforas. (Software Lenguaje Maya - Edupmedia-2008)

### 3.3.3 Diseño del Mapa de navegación

Al culminar el diseño de todas las ventanas se hace indispensable la creación de una guía o mapa de búsqueda que permita ubicar de forma ordenada cada una de las interfaces, que a su vez lograra que los usuarios tengan mejor conocimiento sobre las particularidades que identifican a cada ventana. (Ver gráfico 2).

**3.3.3.1 Elaboración del esquema de navegación:** El mapa de navegación de un software es una guía gráfica que se elabora para brindarle al usuario un fácil manejo del material computacional. Este esquema muestra la forma como están organizadas las interfaces gráficas de usuario, dentro de este esquema se incluye la imagen de la ventana principal y el despliegue de las demás cuando hay un hipervínculo que la interconecta.

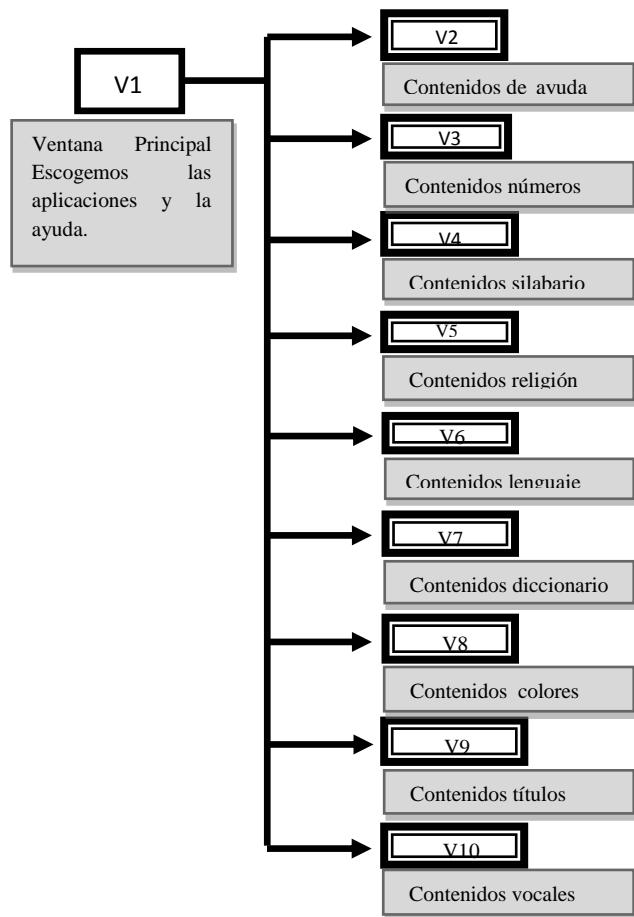


Gráfico 2. Ejemplo del diseño del mapa de navegación. (Software Lenguaje Maya - Edupmedia-2008)

## PUNTOS CLAVE

- La fase de diseño multimedial se encarga de la descripción de las etapas que permitirán conocer en conjunto la estructura que se va a utilizar en el software.
- Se representa gráficamente los contenidos que serán trabajados en el software, y que están consignados en fase del diseño de contenidos, ya sea un mentefacto, mapa conceptual o mapa mental.
- La fase del guión técnico multimedial permite visualizar de una forma detallada la conformación de los diferentes componentes de las interfaces gráficas de usuario que tendrá el software.
- En la sección selección y producción de recursos multimedia se recolectara todo el audio y los videos necesarios para el desarrollo de las actividades, esto se hará como creación original o tomada de fuentes de información.
- La etapa diseño del ambiente de aprendizaje comprende el conjunto de directrices que presenta el diseño de cada una de las interfaces gráficas de usuario. Estas directrices se identifican secuencialmente iniciando con el diseño de la ventana estándar para seguir con cada ventana o interfaz y para finalizar con el diseño del mapa de navegación.
- En el diseño de guía de metáforas se presentan las metáforas que serán utilizadas en las actividades, estas deben ir asociadas al contexto de la población y el ambiente de aprendizaje. El formato de esta sección contiene el nombre, la imagen y la descripción de la metáfora

## REFERENCIAS

- Especificación del proyecto multimedia: Guión multimedia. Disponible en:  
[www.dsic.upv.es/~fgarcia/imm/Traspas/Especificación%20del%20proyecto%20multimedia.ppt](http://www.dsic.upv.es/~fgarcia/imm/Traspas/Especificación%20del%20proyecto%20multimedia.ppt) –
- <http://dewey.uab.es/pmarques/interven.htm>
- [http://es.wikipedia.org/wiki/Mapa\\_conceptual](http://es.wikipedia.org/wiki/Mapa_conceptual).
- [http://es.wikipedia.org/wiki/Mapa\\_mental](http://es.wikipedia.org/wiki/Mapa_mental).
- <http://www.psicopedagogia.com/definicion/mentefacto>.
- <http://nodoativo.blogspot.com/2008/04/cmo-realizar-un-guin-multimedia-parte-i.html>.
- MARQUÉS, Pere. Selección y Uso de Recursos Multimedia. Diseño de Actividades. Usos en el Aula Informática. 1999. Disponible en: <http://dewey.uab.es/pmarques/interven.htm>. Consultado el 03 de abril de 2009
- PEREZ, Felipe. Guión didáctico. <http://laxantesaboramora.obolog.com/guion-didactico-108808>
- <http://www.monografias.com/trabajos908/guion-didactico-multimedia/guion-didactico-multimedia2.shtml>
- Multimedia 2. <http://www.slideshare.net/raymarq/multimedia-2-presentation>

## **Capítulo IV**

### **DISEÑO COMPUTACIONAL**

## 4. Fase III: Diseño computacional

La fase del diseño computacional comprende la descripción de los elementos que permitirán que el sistema funcione a cabalidad con los procesos que fueron definidos en el transcurso del diseño educativo y multimedial. Asimismo se hace una representación de la interacción que tiene el usuario con el sistema en desarrollo, permitiendo que los actores detallen si el software complementa y ayuda al proceso de enseñanza y aprendizaje.



Figura 32. Fase III. Diseño computacional.

### 4.1 Selección y descripción del proceso de desarrollo

En el proceso de desarrollo se ponen de manifiesto las etapas que harán parte del diseño del software, tales como la selección del paradigma de desarrollo, la cual se incluirá en el material computacional, así como las descripciones funcionales del sistema.

El paradigma de desarrollo permite organizar los procesos de tal forma que las actividades puedan ser definidas de una manera más ordenada y con calidad. Se selecciona un paradigma de desarrollo o modelo de proceso según la naturaleza del proyecto y la aplicación, los métodos y las herramientas a utilizarse, y los controles y entregas que se requieren<sup>38</sup>

La subfase de selección y descripción del proceso de desarrollo (*véase norma ISO/IEC 12207 proceso de suministro, tarea seleccionar un modelo de ciclo de vida; también vea en el proceso de desarrollo, tarea definir o seleccionar un modelo de desarrollo*), incluye las siguientes secciones:

<sup>38</sup> PRESSMAN, R. Ingeniería del Software. Editorial Mc Graw Hill. 2002.

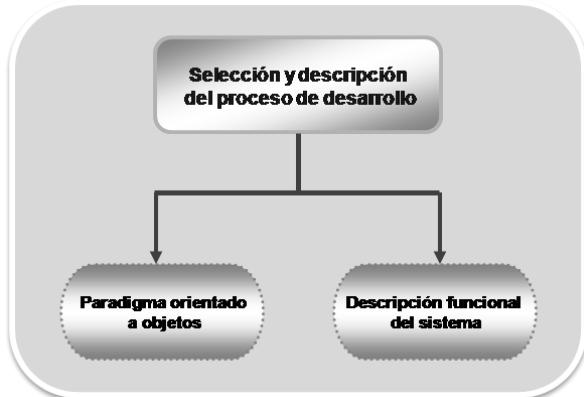


Figura 33. Selección y descripción del proceso de desarrollo.

#### 4.1.1 Paradigma Orientado A Objetos

Un sistema orientado a objetos está compuesto de objetos que interactúan, los cuales mantienen ellos mismos su estado local y proveen operaciones sobre su estado. La representación del estado es privada y no se puede acceder a ella directamente desde fuera del objeto. El proceso de diseño orientado a objetos comprende el diseño de clases de objetos y las relaciones entre estas clases de objetos y las relaciones entre estas clases. Las clases definen los objetos del sistema y sus interacciones<sup>39</sup>. (Véase norma ISO/IEC 12207 proceso de desarrollo, actividad diseño de la arquitectura del software)

El diseño orientado a objetos es parte del desarrollo a objetos en el que se utiliza una estrategia orientada a objetos en el proceso de desarrollo:

- El análisis orientado a objetos: comprende el desarrollo de un modelo orientado a objetos del dominio de aplicación. Los objetos identificados reflejan las entidades y operaciones que se asocian con el problema a resolver<sup>40</sup>.
- El diseño orientado a objetos: comprende el desarrollo de un modelo orientado a objetos de un sistema para implementar los requerimientos identificados. Los objetos de un diseño orientado a objetos están relacionados con la solución del problema a resolver. Pueden existir relaciones estrechas entre algunos objetos del problema y algunos objetos de la solución, pero inevitablemente el diseñador tiene que agregar nuevos objetos para transformar los objetos del problema e implementar la solución<sup>41</sup>.
- La programación orientada a objetos: se refiere a implementar el diseño de software utilizando un lenguaje de programación orientado a objetos, como Java. Un lenguaje orientado a objetos provee los recursos para definir las clases y un sistema para crear los objetos correspondientes a las clases<sup>42</sup>. (Véase también en la fase V producción capítulo IV sección selección de la herramienta de desarrollo).

Los sistemas orientados a objetos son más fáciles de mantener que los sistemas desarrollados con otras aproximaciones, debido a que los objetos son independientes. Tales sistemas pueden ser entendidos y modificados como entidades independientes. Cambiar la implementación de un objeto o agregarle servicios no debe afectar a los otros objetos del sistema. Puesto que los objetos están asociados a cosas, a menudo

<sup>39</sup> SOMMERVILLE, Ian. Ingeniería del Software 7<sup>a</sup> edición. Editorial Addison Wesley. Madrid. 2005. Ed. 7

<sup>40</sup> Ibid. Pág. 286

<sup>41</sup> Ibid.

<sup>42</sup> Ibid.

existe una correspondencia clara entre las entidades del mundo real (como los componentes de hardware) y los objetos de control del sistema. Esto mejora la comprensión y, por lo tanto, la mantenibilidad del diseño<sup>43</sup>.

#### 4.1.2 Descripción funcional del sistema

En esta etapa se describen las características sobresalientes que conciernen a la funcionalidad del sistema o software diseñado, se establece la población a quien va dirigido, la intencionalidad, identificación y justificación del tipo de software seleccionado, especificación sobre los beneficios que fortalecen el proceso de enseñanza y aprendizaje y como ultima particularidad se mencionan las distintas secciones en las cuales se encuentra dividido el software.

<b>DESCRIPCIÓN DEL SISTEMA</b>
<p>Mayensesoft es software educativo que busca comprender la simbología maya y la composición de palabras en este lenguaje. Permitiéndole al usuario conocer de manera detallada y básica nuevas formas de escritura y lectura de símbolo de una nueva cultura como verbos y sustantivos. Además podrán desarrollar la comprensión de títulos y colores de la lengua maya de manera didáctica y dinámica. Conociendo la importancia de la escritura que tienen las deidades en las lenguas antiguas.</p> <p>Cuenta con una serie de actividades que se describirán a continuación:</p> <ol style="list-style-type: none"><li>a. Se ilustran los verbos más importantes del lenguaje maya según la representación y pronunciación en maya y la representación de estos en español. El usuario escogerá libremente que verbo desea visualizar. Para determinar el indicador 1.1 se mostrarán botones que representan los verbos, los cuales el usuario al darles clic, se le mostrará la representación y pronunciación por medio de imágenes.</li><li>b. Se enseñaran los verbos más importantes del lenguaje maya según la representación y pronunciación en maya y el significado de estos en español. El usuario escogerá libremente que verbo desea visualizar. Para determinar el indicador 1.2 se mostrarán botones que representan los verbos, los cuales el usuario al darles clic, se le mostrará la representación y pronunciación por medio de imágenes.</li><li>c. Se ilustran los números del 1 al 10 según la representación, escritura y pronunciación en maya y la representación de estos en español. El usuario escogerá libremente que numero desea visualizar. Para determinar el indicador 1.3 se mostrarán botones que representan los números del 1 al 10, los cuales el usuario al darles clic, se le mostrará la escritura, representación y pronunciación por medio de audio y escritura de los números mayas.</li><li>d. Se ilustran los títulos más representativos del lenguaje maya según la representación y pronunciación en maya y la representación de estos en español. El usuario escogerá libremente el título que desea visualizar. Para determinar el indicador 2.1 se mostrarán botones que representan los algunos de los títulos los cuales el usuario al darles clic, se le mostrará la representación y pronunciación representándolos por medio imágenes.</li><li>e. Se muestran los colores básicos de la cultura maya, en la que se presentará un cuadro relleno con el color seleccionado, la pronunciación y la escritura del mismo. El usuario escogerá de forma libre que color desea conocer. Para determinar el indicador 2.2 se pondrán una serie de botones que representarán el nombre los colores básicos, a través de los cuales el usuario hará clic, observará la pronunciación y la escritura en maya de los colores.</li><li>f. Se muestran los títulos primordiales de la cultura maya, en la que se presentará un cuadro en el que muestra la representación del título en maya y la pronunciación del mismo. El usuario escogerá de forma libre que título desea conocer. Para determinar el indicador 2.3 se pondrán una serie de botones que representarán el nombre del título, a través de los cuales el usuario hará clic, observará la pronunciación y representación en maya de cada uno de éstos.</li><li>g. Se muestra un cuadro lleno con color y la pronunciación de los colores básicos de la cultura maya. El usuario arrastrará de manera aleatoria un cuadro de color con el nombre del mismo en maya, los cuales ubicara en unos recuadros donde se encuentra el nombre del color en español.</li><li>h. Se muestra las diferentes deidades que conforman la religión politeísta maya, mostrando una imagen del respectivo Dios escogido y haciendo una breve descripción del mismo en español o en inglés según la necesidad del usuario. El usuario escogerá autónomamente que deidad desea conocer. Para determinar el indicador 3.1 se pondrán una serie de botones que representaran el nombre de cada deidad, a través de los cuales el usuario al darles clic, tendrá la oportunidad de observar la imagen del Dios y una breve descripción de éste.</li></ol>

Formato 15. Ejemplo de la descripción del sistema (Software Lenguaje Maya - Edupmedia-2008)

<sup>43</sup> Ibid.

## 4.2 Análisis de requerimientos

Al planificar el diseño de un software educativo se debe tener presente la planta física y la población a quien va dirigida refiriendo a docentes, estudiantes y usuarios; es importante establecer los requerimientos a partir de la necesidad que se presentó en el proceso de enseñanza y aprendizaje, de esta forma se conocerá si el software cumplió a cabalidad con los objetivos dispuestos durante el proceso de diseño. Para desarrollar los requerimientos se deben dividir éstos en secciones denominadas subsistemas y requerimientos funcionales.

El análisis de los requerimientos (*véase norma ISO/IEC 12207 proceso de adquisición, tarea definición de los requerimientos del software*) permite conocer los requisitos que se necesitan para desarrollar el software, las etapas de este proceso son:

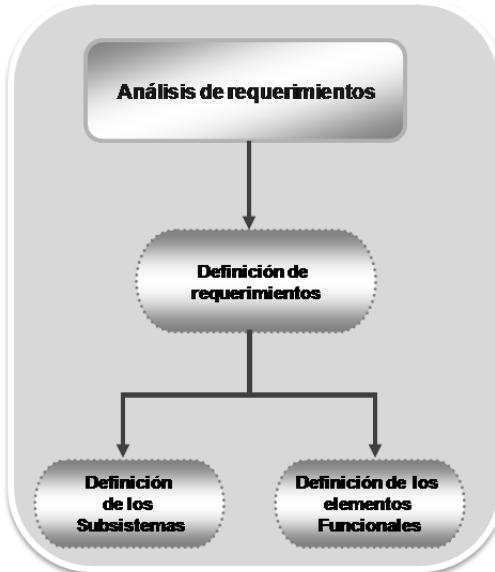


Figura 34. Análisis de requerimientos

### 4.2.1 Definición de requerimientos

(*Véase norma ISO/IEC 12207 proceso de desarrollo, tarea requisitos del software y diseño detallado del software*)

**4.2.1.1 Definición de los subsistemas:** En esta sección se enunciaran los calificativos con que se identifican los módulos que hacen parte del sistema. Los subsistemas agrupan una o varias interfaces gráficas de usuario.

**Colocar ejemplo**

**4.2.1.2 Definición de los elementos funcionales:** Los requerimientos funcionales exponen los requisitos que deben contener cada subsistema, estos son enumerados ascendente y son agrupados dependiendo del calificativo del módulo. Cabe resaltar que la numeración debe ser consecutiva sin importar la división de los módulos. Después de realizar el procedimiento anterior los requerimientos deben ser descritos gráficamente por medio de cuadros, estos serán consecutivos a la numeración dispuesta en los módulos antes mencionados. Los cuadros mostraran las especificaciones de cada uno de los requerimientos.

<b>REQUERIMIENTO Nro. 00046</b>		
El sistema debe contar con un diccionario de Glifos Mayas		
<i>Usuario</i>	X	El usuario podrá escribir la palabra en español o inglés, y el diccionario le mostrará el glifo correspondiente.
<i>Sistema</i>		
<i>Funcional</i>	X	
<i>No funcional</i>		
<b>Pre-condición</b>		<b>Post-condición</b>
La palabra introducida debe existir en el diccionario		Se muestra el glifo, la pronunciación y escritura del mismo.
<i>Fecha creación</i>	20-10-2008	<b>Responsable</b>
<i>Fecha de procesamiento</i>	01-12-2008	Raúl Toscano-Manuel Caro
<i>Modificación</i>		<b>Causa/Responsable</b>

Formato 16. Ejemplo de un requerimiento formalizado. (Software Lenguaje Maya - Edupmedia-2008)

### 4.3 Casos de uso

Un caso de uso especifica el comportamiento de un sistema o de una parte del mismo, y es una descripción de un conjunto de secuencias de acciones, incluyendo variantes, que ejecuta un sistema para producir un resultado observable de valor para un actor.

Los casos de uso se utilizan para describir la forma como el usuario interactúa con el sistema y las acciones que presenta el sistema como respuesta a dicha interacción. Estos se dividen en elementos, actores, diagramas de casos de uso y su objetivo es permitir la captura de requisitos potenciales para el nuevo sistema.

Los casos de uso deben cumplir dos objetivos: Definir los requisitos funcionales del sistema y proporcionar una base para la validación de las pruebas<sup>44</sup>.

---

<sup>44</sup> BOOCHE, G. Lenguaje de Modelado Unificado. Editorial Adisso Wesley. Madrid. 2001.

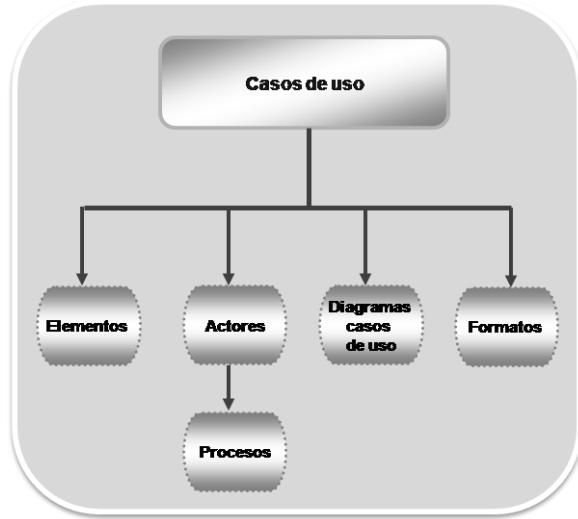


Figura 35. Casos de uso

**4.3.1 Elementos:** Los elementos son los gráficos que se utilizarán en el diagrama de los casos de uso y estos representan los actores, los casos de uso y las relaciones que se muestran entre cada una de las partes que conforman los elementos.

**4.3.2 Actores:** Un Actor representa un conjunto coherente de roles que juegan los usuarios de los casos de uso al interactuar con estos. Los actores pueden ser personas o pueden ser sistemas mecánicos, estos se representan como monigotes. Por ejemplo, en el modelado de un banco, el procesamiento de un préstamo implica, entre otras cosas, la interacción entre un cliente y un responsable de préstamos.<sup>45</sup> En esta sección se enunciaran los distintos actores que harán parte del software.

**4.3.2.1 Procesos:** Los procesos son las funciones que puede ejercer el actor con respecto al software, estas acciones dependen de los contenidos que presentan cada interfaz.

**4.3.3 Diagramas casos de uso:** Los diagramas de casos de uso son importantes para modelar el comportamiento de un sistema, un subsistema o una clase. Cada uno muestra un conjunto de casos de uso, actores y sus relaciones. Los diagramas de casos de uso se emplean para modelar la vista de casos de uso de un sistema. La mayoría de las veces, esto implica modelar el contexto del sistema, subsistema o clase, o el modelado de los requisitos de comportamiento de esos elementos<sup>46</sup>.

Gráficamente, un caso de uso se representa como una elipse de borde continuo, incluyendo normalmente solo su nombre<sup>47</sup>.

En los casos de uso existen cuatro tipos de relaciones las cuales se detallan a continuación:

**4.3.3.1 Dependencia** Es una relación semántica entre dos elementos, en la cual un cambio a un elemento (el elemento independiente) puede afectar a la semántica del otro elemento (el elemento dependiente).

<sup>45</sup> Ibíd., BOOCH. Pág. 192, 194

<sup>46</sup> Ibíd., BOOCH. Pág. 203

<sup>47</sup> Ibíd., BOOCH. Pág. 16

Gráficamente, una dependencia se representa con una línea discontinua, posiblemente dirigida, que incluye a veces una etiqueta<sup>48</sup>.

**4.3.3.2 Asociación:** Es una relación estructural que describe un conjunto de enlaces, los cuales son conexiones entre objetos. La agregación es un tipo especial de asociación, que representa una relación estructural entre un todo y sus partes. Gráficamente, una asociación se representa como una línea continua, posiblemente dirigida, que a veces incluye una etiqueta, y a menudo incluye otros adornos, como la multiplicidad y los nombres del rol<sup>49</sup>.

**4.3.3.3 Generalización:** Es una relación de especialización/generalización en la cual los objetos de los elementos especializado (el hijo) pueden sustituir a los objetos del elemento general (el padre). De esta forma, el hijo comparte la estructura y el comportamiento del padre. Gráficamente, una relación de generalización se representa con una línea continua con una punta de flecha vacía apuntando al padre<sup>50</sup>.

**4.3.3.4 Realización:** Es una relación semántica entre clasificadores, en donde un clasificador especifica un contrato que otro clasificador garantiza que cumplirá. Se pueden encontrar relaciones de realización en dos sitios: entre interfaces y las clases y componentes que las realizan, y entre los casos de usos y las colaboraciones que los realizan. Gráficamente, una relación de realización se representa como una mezcla entre una generalización y una relación de dependencia<sup>51</sup>.

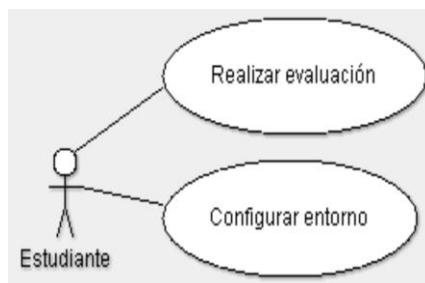


Gráfico 2. Ejemplo de un diagrama de casos de uso. (Software Lenguaje Maya - Edupmedia-2008)

#### 4.3.4 Formatos

FORMATO DE CASOS DE USO	
Nº cu – 01	<b>Nombre del proceso:</b> ver diccionario
<b>Descripción:</b> El sistema cuenta con un diccionario para la identificación de los glifos mayas y su traducción al español	
<b>Actor 1:</b> Usuario	<b>Actor 2:</b> Sistema
1. Abre la ventana <b>ver diccionario</b>	
	2. Muestra la ventana de <b>ver diccionario</b>

<sup>48</sup> Ibid. BOOC. Pág. 20

<sup>49</sup> Ibid. BOOC

<sup>50</sup> Ibid. BOOC

<sup>51</sup> Ibid. BOOC

3. Escoge la palabra que desea visualizar en maya y español		
	4. Muestra la palabra escogida por el usuario.	
<b>Caminos de excepción</b>		
<b>Actor 1</b>	<b>Actor 2</b>	
	4. En caso de no existir la palabra, vuelve al punto 3	
<b>Puntos de extensión</b>		
<b>Autor</b>	<b>Requerimiento</b>	<b>Modificación</b>
Elcy Galarcio  Sindy Ruiz C.		Septiembre 4/2008

*Formato 17. Ejemplo de un formato de casos de uso. (Software Lenguaje Maya - Edupmedia-2008)*

#### **4.4 Diagrama de clases**

Un diagrama de clase muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar la vista de diseño estático de un sistema. Principalmente, esto incluye modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas<sup>52</sup>. Al diagrama de clases también se le conoce como modelo de conceptos porque inicialmente ayuda al descubrimiento de conceptos importantes y persistentes en el desarrollo del software.

El diagrama de clases está compuesto por elementos identificados como clase y relaciones, que al unirlos conforman los esquemas o diagramas que representarán un modelo de clases. Estos diagramas resultan de la composición de los casos de uso.

El diagrama de clases debe expresar detalles de los objetos que existen en el área de enfoque, en él se incluyen atributos y métodos de cada una de las clases.

---

<sup>52</sup> Ibid. BOOCH. Pág. 93

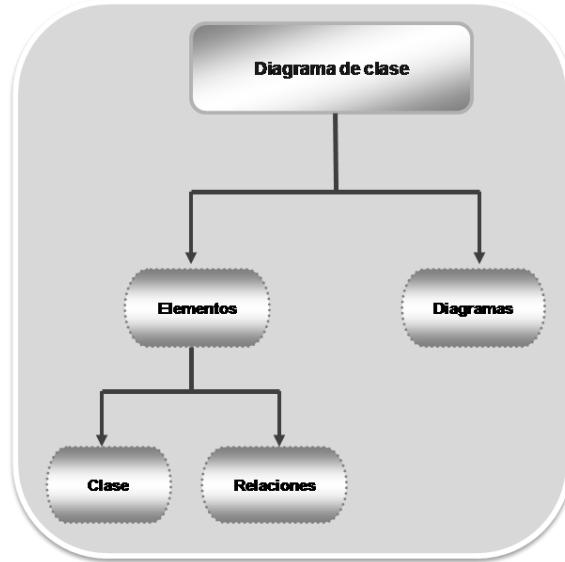


Figura 36. Diagrama de clases

En el diagrama de clase se han utilizado dos categorías principales para pormenorizar las unidades constituyentes de cada diagrama. Éstas son:

### 3.4.1 Elementos

**3.4.1.1 Clase:** Una clase es una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica. Una clase implementa una o más interfaces. Gráficamente, una clase se representa como un rectángulo, que normalmente incluye su nombre, atributos y operaciones<sup>53</sup>.

- **Nombres:** El nombre de una clase puede ser texto formado por cualquier número de letras, números y ciertos signos de puntuación (excepto signos como los dos puntos, que se utilizan para separar el nombre de una clase y el del paquete que la contiene) y puede extenderse a lo largo de varias líneas. En la práctica los nombres de clases son nombres cortos o expresiones nominales extraídos del vocabulario del sistema que se está modelando. Normalmente, en el nombre de la clase se pone en mayúscula la primera letra de cada palabra<sup>54</sup>.
- **Atributos:** Un atributo es una propiedad de una clase identificada con un nombre, que describe un rango de valores que pueden tomar las instancias de la propiedad. Una clase puede tener cualquier número de atributos o no tener ninguno. Un atributo representa alguna propiedad del elemento que se está modelando que es compartida por todos los objetos de esa clase. Gráficamente los atributos se listan en un compartimento justo debajo del nombre de la clase<sup>55</sup>.
- **Operaciones:** Es la implementación de un servicio que puede ser requerida a cualquier objeto de la clase para que muestre un comportamiento. En otras palabras, una operación es una abstracción de algo que se puede hacer a un objeto y que es compartido por todos los objetos de la clase. Una clase puede tener cualquier número de operaciones o ninguna. Gráficamente, las operaciones se listan en un compartimento justo debajo de los atributos de la clase<sup>56</sup>.

<sup>53</sup> Ibid. BOOC. Pág. 15

<sup>54</sup> Ibid. BOOC. Pág. 43

<sup>55</sup> Ibid. BOOC. Pág. 43

<sup>56</sup> Ibid. BOOC. Pág. 44

➤ **Visibilidad:** Uno de los detalles más importantes que se pueden especificar para los atributos y operaciones de un clasificador es su visibilidad. La visibilidad de una característica especifica si puede ser utilizada por otros clasificadores. En UML se puede especificar una característica con cualquiera de los tres niveles de visibilidad disponibles.

- **Public:** Cualquier clasificador externo con visibilidad hacia el clasificador dado puede utilizar la característica; se especifica precediéndola del símbolo +.
- **Protected:** Cualquier descendiente del clasificador puede utilizar la característica; se especifica precediéndola del símbolo #.
- **Private:** Sólo el propio clasificador puede utilizar la característica; se especifica precediéndola del símbolo -<sup>57</sup>.

#### **4.4.2 Diagramas**

A Continuación se muestra un ejemplo del diagrama de clases.

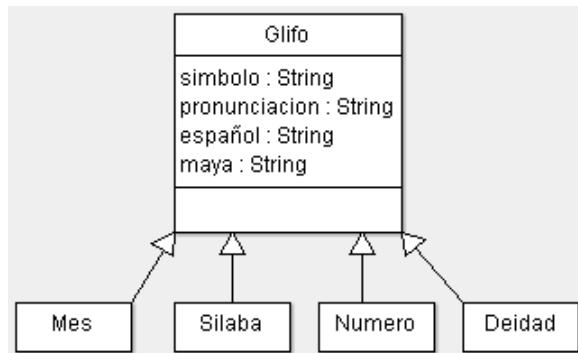


Gráfico 3. Ejemplo de un diagrama de clases. (Software Lenguaje Maya - Edupmedia-2008)

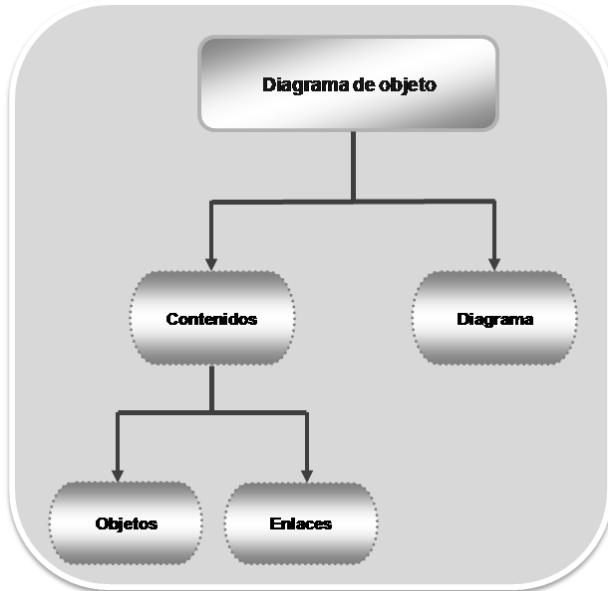
#### **4.4 Diagrama de objetos**

Un diagrama de objetos es un diagrama que representa un conjunto de objetos y sus relaciones en un momento concreto. Gráficamente, un diagrama de objetos es una colección de nodos y arcos. Un diagrama de objetos es un tipo especial de diagramas y comparte las propiedades comunes al resto de los diagramas (un nombre y un contenido gráfico que es una proyección de un modelo)<sup>58</sup>.

---

<sup>57</sup> Ibid. BOOCH. Pág. 108

<sup>58</sup> Ibid. BOOCH. Pág. 170-171



*Figura 37. Diagrama de objeto*

La elaboración del diagrama de objeto requiere de varios elementos que se describen a continuación:

#### **4.5.1 Contenidos:**

**4.5.1.1 Objetos:** Los objetos son instancias de clases, así que es perfectamente apropiado decir que todos los objetos son instancias, aunque algunas instancias no son objetos (por ejemplo, una instancia de una asociación no es un objeto, es sólo una instancia, también llamada enlace)<sup>59</sup>.

**4.5.1.2 Enlaces:** Un enlace es una conexión semántica entre objetos. En general, un enlace es una instancia de una asociación. Siempre que una clase tenga una asociación con otra clase, podría existir un enlace entre las instancias de las dos clases; siempre que haya un enlace entre dos objetos, un objeto puede enviar un mensaje al otro<sup>60</sup>.

#### **4.5.2 Diagramas**

A Continuación se muestra un ejemplo del diagrama de objetos.

---

<sup>59</sup> Ibid. BOOCH. Pág. 160.

<sup>60</sup> Ibid. BOOCH. Pág. 182.

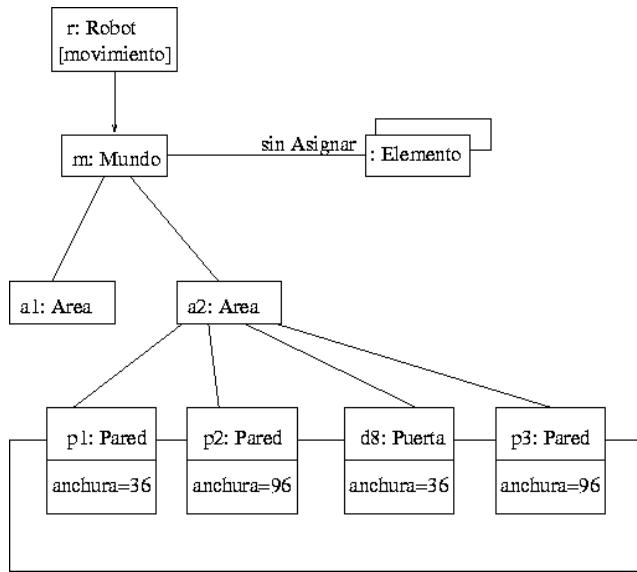


Gráfico 5. Modelado de estructuras de objetos

#### 4.6 Diagrama de secuencia

Los *Diagramas de Secuencia* representan una interacción entre objetos de manera secuencial en el tiempo. Muestra la participación de objetos en la interacción entre sus “líneas de vida” (desde que se instancia el objeto) y los mensajes que ellos organizadamente intercambian en el tiempo. El responsable o actor es quien inicia el ciclo interactuando inicialmente con la interfaz de usuario (IGU); en seguida se inician todos los objetos que intervienen en el funcionamiento del aplicativo. En este diagrama se comienza a observar el comportamiento del sistema a partir de los eventos generados por los actores<sup>61</sup>.

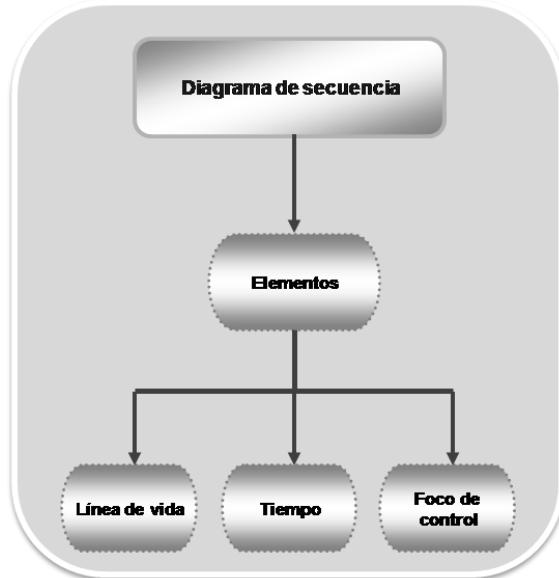


Figura 38. Diagrama de secuencia

##### 4.6.1 Elementos

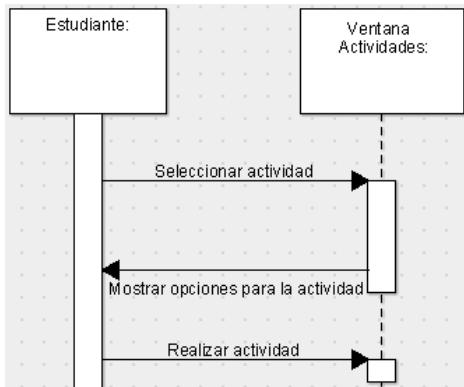
<sup>61</sup> Ibid. BOOCH.

En la subfase del diagrama de secuencia se especifica un componente fundamental que se descompone en subcategorías:

**4.6.1.1 Línea de vida:** Es la línea discontinua vertical que representa la existencia de un objeto a lo largo de un período de tiempo. La mayoría de los objetos que aparecen en un diagrama de interacción existirán mientras dure la interacción, así que los objetos se colocan en la parte superior del diagrama, con sus líneas de vida dibujadas desde arriba hasta abajo. Los objetos pueden destruirse durante la interacción.<sup>62</sup>

**4.6.1.2 Tiempo:** Los sistemas de tiempo real son, como su propio nombre lo indica, sistemas en los que el tiempo juega un papel crítico. Los eventos pueden ocurrir a intervalos de tiempos regulares o irregulares; la respuesta a un evento debe ocurrir en un lapso de tiempo predecible absoluto. El paso de mensajes representa el aspecto dinámico de cualquier sistema, así que cuando se modelan con UML los aspectos de un sistema en los que el tiempo de respuesta es un factor crítico, se puede dar un nombre a cada mensaje de una interacción para poder utilizarlo como una marca de tiempo<sup>63</sup>.

**4.6.1.3 Foco de control:** Es un rectángulo delgado y estrecho que representa el período de tiempo durante el cual un objeto ejecuta una acción, bien sea directamente o a través de un procedimiento subordinado. La parte superior del rectángulo se alinea con el comienzo de la acción; la inferior se alinea con su terminación (y puede marcarse con un mensaje de retorno)<sup>64</sup>. Veamos un ejemplo del *diagrama de secuencias*:



Gráfica 6. Ejemplo de un diagrama de secuencia. (Software Lenguaje Maya - Edupmedia-2008)

## 4.7 Diseño Modelado del sistema de conocimiento/bases de datos

En esta etapa se recomienda la utilización de bases de datos relacionales o archivos de texto con estructura XML. Si la opción es la primera, se deberá crear un MER (Modelo Entidad Relación), un MR (Modelo relacional) basado en el MER y un diccionario de datos para los metadatos (como tipos, tamaños, y otras cualidades de los campos de datos). Estos modelos son de uso generalizado y cuentan con abundante bibliografía. Si la opción es el archivo de XML, bastará entonces con el diccionario de datos.

<sup>62</sup> Ibid. BOOCH. Pág. 214.

<sup>63</sup> Ibid. BOOCH. Pág. 285.

<sup>64</sup> Ibid. BOOCH. Pág. 214.

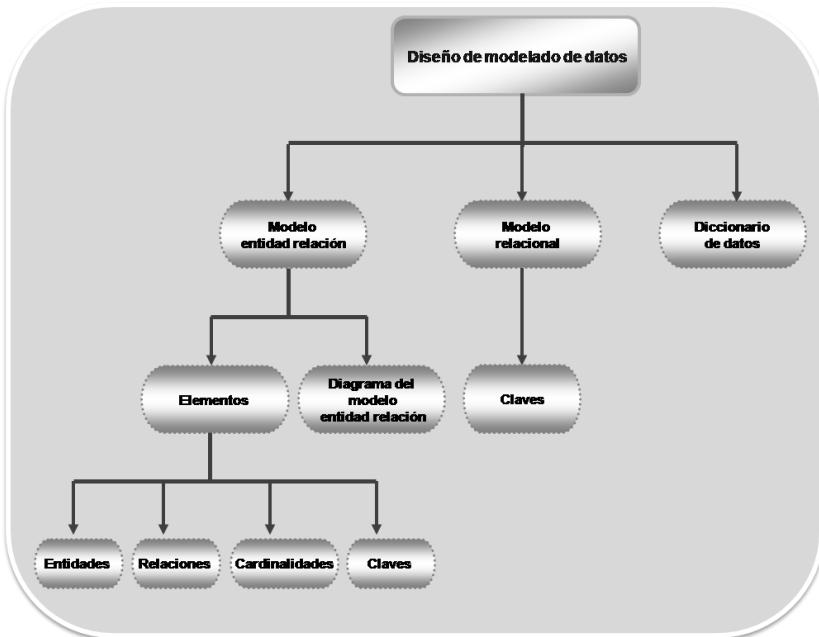


Figura 39. Diseño del modelado del sistema de conocimiento/bases de datos

## 4.7.1 Modelo Entidad Relación

El modelo de datos entidad-relación (E-R) está basado en una percepción del mundo real que consta de un conjunto de objetos llamados entidades y de relaciones entre estos objetos. Se desarrolló para facilitar el diseño de bases de datos permitiendo la especificación de un esquema de la empresa que representa la estructura lógica completa de una base de datos<sup>65</sup>.

### 4.7.1.1 Elementos

- **Entidades:** Una entidad tiene un conjunto de propiedades y los valores para algún conjunto de propiedades pueden identificar una entidad de forma única.

Una entidad se representa mediante un conjunto de atributos. Los atributos describen propiedades que poseen cada miembro de un conjunto de entidades. La designación de un atributo para un conjunto de entidades expresa que la base de datos almacena información similar concerniente a cada entidad del conjunto de entidades; sin embargo, cada entidad puede tener su propio valor para cada atributo<sup>66</sup>.

- **Relaciones:** una relación es una asociación entre diferentes entidades. La función que desempeña una entidad en una relación se llama papel de la entidad. Debido a que los conjuntos de entidades que participan en un conjunto de relaciones son generalmente distintos, los papeles están implícitos y no se especifican normalmente. Sin embargo, son útiles cuando el significado de una relación necesita aclaración.

Una relación puede también tener atributos descriptivos. Considérese un conjunto de relaciones impositor con un conjunto de entidades cliente y cuenta. Se podría asociar el atributo fecha-acceso a esta relación para especificar la fecha más reciente en que un cliente accedió a una cuenta<sup>67</sup>.

- **Cardinalidades:** La correspondencia de cardinalidades o razón de cardinalidad, expresa el número de entidades a las que otra entidad puede estar asociada vía un conjunto de relaciones.

<sup>65</sup> Ibid. BOOC. Pág. 15

<sup>66</sup> Ibid. BOOC. Pág. 15-16

<sup>67</sup> Ibid. BOOC. Pág. 18-19

La correspondencia de cardinalidades es la más útil describiendo conjuntos de relaciones binarias, aunque ocasionalmente contribuye a la descripción de conjuntos de relaciones que implican más de dos conjuntos de identidades. Para un conjunto de relaciones binarias R entre los conjuntos de entidades A y B, la correspondencia de cardinalidades debe ser una de las siguientes:

- **Uno a Uno:** Una entidad en A se asocia con a lo sumo una entidad en B, y una entidad en B se asocia con a lo sumo una entidad en A<sup>68</sup>.
- **Uno a Varios:** Una entidad en A se asocia con cualquier número de entidades en B. Una entidad en B, sin embargo, se puede asociar con a lo sumo en una entidad en A<sup>69</sup>.
- **Varios a Uno:** Una entidad en A se asocia con a lo sumo una entidad en B. Una entidad en B, sin embargo, se puede asociar con cualquier número de entidades en A<sup>70</sup>.
- **Varios a Varios:** Una entidad en A se asocia con cualquier número de entidades en B, y una entidad en B se asocia con cualquier número de entidades en A<sup>71</sup>.

➤ **Claves:** Es importante ser capaz de especificar como las entidades dentro de un conjunto de relaciones dado son distinguibles. Conceptualmente, las entidades y relaciones individuales son distintas; desde una perspectiva de bases de datos, sin embargo, la diferencia entre ellas se debe expresar en término de sus atributos. El concepto de clave permite hacer tales distinciones<sup>72</sup>.

- **Superclave:** Es un conjunto de uno o más atributos que, tomados colectivamente, permiten identificar de forma única una entidad en el conjunto de entidades. El atributo dni del conjunto de entidades cliente es suficiente para distinguir una entidad cliente de las otras. Así, dni es una superclave. Análogamente, la combinación de nombre-cliente-dni es una superclave del conjunto de entidades cliente. El atributo nombre-cliente de cliente no es una superclave, porque varias personas podrían tener el mismo nombre<sup>73</sup>.
- **Claves candidatas:** Es posible que conjunto distintos de atributos pudieran servir como claves candidatas. Supóngase que una combinación de nombre-cliente y calle-cliente es suficiente para distinguir entre los miembros del conjunto de entidades cliente. Entonces, los conjuntos {dni} y {nombre-cliente, calle-cliente} son claves candidatas.
- **Clave primaria:** este término se utiliza para denotar una clave candidata y un conjunto de relaciones. Se usara el término clave primaria para denotar una clave candidata que es elegida por el diseñador de la base de datos como elemento principal para identificar las entidades dentro de un conjunto de entidades. Una clave (primaria, candidata y superclave) es una propiedad del conjunto de identidades, más que las entidades individuales.

La composición de la clave primaria para un conjunto de relaciones depende de la estructura de los atributos asociados al conjunto de relaciones R. Si el conjunto de relaciones R no tiene atributos asociados, entonces el conjunto de atributos forma una superclave para el conjunto de relaciones<sup>74</sup>.

**4.7.1.2 Diagrama del modelo Entidad-Relación:** la estructura lógica general de una base de datos se puede expresar gráficamente mediante un diagrama E-R. La simplicidad relativa o la claridad pictórica de esta técnica de diagrama puede ser en gran parte la causa del uso ampliamente extendido del uso del modelo E-R. Tal diagrama consta de los siguientes componentes principales:

➤ **Rectángulos:** representan conjunto de entidades.

---

<sup>68</sup> Ibid. BOOCH. Pág. 21

<sup>69</sup> Ibid. BOOCH. Pág. 22

<sup>70</sup> Ibid. BOOCH.

<sup>71</sup> Ibid. BOOCH.

<sup>72</sup> Ibid. BOOCH. Pág. 23

<sup>73</sup> Ibid. BOOCH.

<sup>74</sup> Ibid. BOOCH. Pág. 24

- **Elipses:** representan atributos.
- **Rombos:** representan relaciones.
- **Líneas:** Unen atributos a conjuntos de entidades y conjuntos de entidades a conjuntos de relaciones.
- **Elipses dobles:** representan atributos multivalorados.
- **Elipses discontinuas:** denotan atributos derivados.
- **Líneas dobles:** indican participación total de una entidad en un conjunto de relaciones<sup>75</sup>.

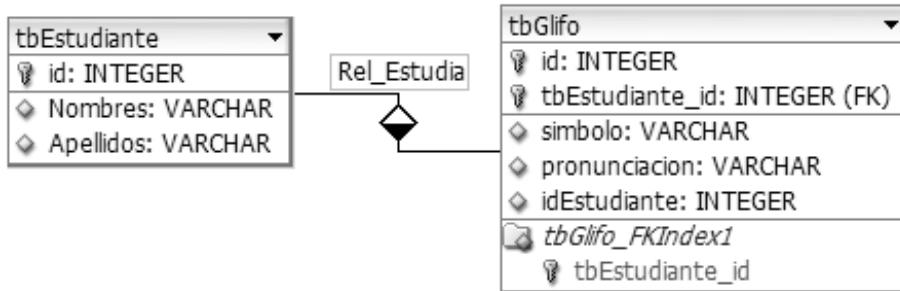


Figura 40. Vista parcial de un MER. (Software Lenguaje Maya - Edupmedia-2008)

#### 4.7.2 Modelo Relacional

El modelo relacional se ha establecido como el principal modelo de datos para las aplicaciones de procesamiento de datos. Una base de datos relacional consiste en un conjunto de tablas, a cada una de las cuales se le asigna un nombre exclusivo. Cada fila de la tabla representa una relación entre un conjunto de valores. Dado que cada tabla es un conjunto de dichas relaciones, hay una fuerte correspondencia entre el concepto de tabla y el concepto matemático de relación, del que toma su nombre el modelo de datos relacional<sup>76</sup>.

**4.7.2.1 Claves:** El esquema de una base de datos relacional se basa en las tablas derivadas de un esquema E-R, es posible determinar la clave primaria del esquema de una relación a partir de las claves primarias de los conjuntos de entidades o de relaciones de los que se deriva el esquema<sup>77</sup>.

#### 4.7.3 Diccionario De Datos

Un sistema de bases de datos relacionales necesita tener datos sobre las relaciones, como el esquema de las mismas. Esta información se denomina diccionario de datos o catálogo del sistema. Entre estos tipos de información que debe guardar el sistema figuran los siguientes:

- Los nombres de las relaciones.
- Los nombres de los atributos de cada relación.
- Los dominios y las longitudes de los atributos.
- Los nombres de las vistas definidas en la base de datos y las definiciones de esas vistas.
- Las ligaduras de la integridad (por ejemplo, las ligaduras de las claves).

<sup>75</sup> Ibid. BOOC.

<sup>76</sup> Ibid. BOOC. Pág. 45

<sup>77</sup> Ibid. BOOC. Pág. 50

Además, muchos sistemas guardan los datos siguientes de los usuarios del sistema:

- Los nombres de los usuarios autorizados.
- La información de las cuentas de usuarios.

Además, se puede guardar información estadística y descriptiva sobre estos asuntos:

- Número de tuplas de cada relación.
- Método de almacenamiento utilizado para cada relación (por ejemplo, las ligaduras de las claves)<sup>78</sup>.

---

<sup>78</sup> Ibid. BOOCH. Pág. 235-236

## **PUNTOS CLAVE**

- La fase de diseño computacional detalla las funcionalidades que se implementarán teniendo como fundamento los criterios definidos en el diseño educativo y multimedial.
- La selección y descripción del proceso de desarrollo busca tener un proceso bien definido que sirva como marco de trabajo para los integrantes del equipo de trabajo, donde una de las premisas principales es la calidad.
- La descripción del sistema debe ilustrar a las personas de una forma clara sobre la funcionalidad del software.
- En la fase de Análisis de Requerimientos busca establecer los requerimientos a partir de la necesidad que se presentó en el proceso de enseñanza y aprendizaje, debe tener en cuenta la población educativa a quien va dirigida.
- El Diagrama de Casos de Uso se utilizan para describir la forma como el usuario interactúa con el sistema. Los casos de uso ayudan a definir los requisitos funcionales del sistema y proporcionar una base para la validación de las pruebas. Los formatos de casos de uso describen cada caso de uso, el cual se representa en el diagrama de casos de usos.
- En el Diseño de Modelado de Datos, el diseñador retoma todos los datos encontrado hasta ahora en el sistema y los describe con diagramas y formatos específicos que representen la persistencia en el sistema. Se puede plantear la solución con el uso de base de datos relacionales o con estructura XML.

## **REFERENCIAS**

- ARLOW, Jim. UML and the Unified Process. Editorial Addison Wesley. 2002.
- BOOCH, G. Lenguaje de Modelado Unificado. Editorial Addison Wesley. Madrid. 2001.
- ERIKSSON, Hans y otros. UML2 Toolkit. Editorial Wiley Publishing. 2004.
- FLOWER, Martin. UML gota a gota. Editorial Addison Wesley. 1999.
- ISO 12207. Disponible en: <http://www.abelia.com/docs/12207cpt.pdf>. Consultado el 23 de febrero de 2009
- PRESSMAN, Roger. Ingeniería del Software. Editorial Mc Graw Hill. 2002. Ed. 5
- PRESSMAN, Roger. Ingeniería del Software. Editorial Mc Graw Hill. 2005. Ed. 6
- SCHMULER, Joseph. Aprendiendo UML en 24 horas. Editorial Prentice Hall. 2001.
- SOMMERVILLE, Ian. Ingeniería del Software. Editorial Addison Wesley. 2005. Ed. 7

**Capítulo V**  
**PRODUCCIÓN**

## 5. Fase IV: Producción

En la fase de producción se agrupan todos los materiales obtenidos en el desarrollo de las fases anteriores, por tal motivo es aquí donde se realiza la implementación de todos los documentos obtenidos en las fases de diseños que serán codificados por un especialista en programación permitiendo el desarrollo final del software, que conllevará a la presentación de un prototipo el cual será evaluado para determinar si cumplió con los objetivos propuestos. Además como último paso se realizará un manual de usuario para que los usuarios puedan navegar fácilmente por el software.



Figura 40. Fase IV. Producción

### 5.1 Selección de herramienta de desarrollo

Para llevar a cabo la codificación es necesario seleccionar una herramienta o lenguaje de programación que sea acorde con las especificaciones del software desarrollado, es importante tener en cuenta que la persona encargada de la codificación domine la herramienta o lenguaje seleccionado para que se facilite la programación de las actividades y contenidos del software, permitiendo el cumplimiento de las metas propuestas en términos de eficacia y eficiencia.

La selección de la herramienta de desarrollo (*véase norma ISO/IEC 12207 proceso de desarrollo, tarea seleccionar, adaptar, y usar normas, métodos, herramientas y lenguajes de programación; tarea desarrollar y preparar planes para preparar las actividades del proceso de desarrollo*) permite conocer el lenguaje de programación seleccionado, las categorías implicadas en este proceso son:



Figura 41. Selección de la herramienta de desarrollo

### 5.1.1 Selección del lenguaje de programación

(Véase norma ISO/IEC 12207 proceso de desarrollo, tarea diseño de la arquitectura del software)

**5.1.1.1 Características del lenguaje de programación:** En este enunciado se describirán las especificaciones que contiene el lenguaje seleccionado para la programación del software, como los entornos de funcionamiento, flexibilidad, elementos y sus atributos.

**5.1.1.2 Sintaxis del lenguaje de programación:** En este bloque se detallaran la estructura del código del lenguaje de programación seleccionado determinando las etiquetas, clase, palabras reservadas, operadores, estructura y el significado de sus elementos.

## 5.2 Codificación

Esta etapa ayuda a que el desarrollo del software con respecto a la programación tenga una presentación correcta, entendible y con buen estilo, la aplicación de estos parámetros permitirá fácilmente el mantenimiento del software para poder corregir los errores y modificar algún requisito cuando sea necesario.

Las categorías del proceso que se llevan a cabo para desarrollar la subfase de la codificación (véase norma ISO/IEC 12207 proceso de desarrollo, tareas integración del sistema, diseño detallado del software, codificación y pruebas del software) comprenden:

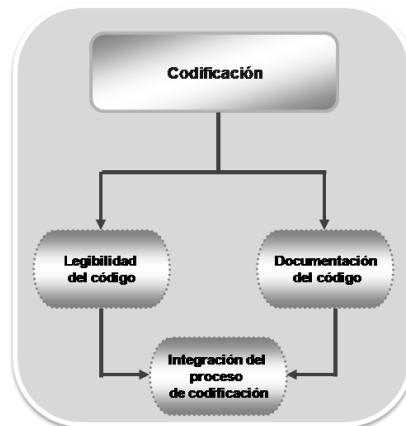


Figura 42. Codificación

### **5.2.1 Legibilidad del código**

Lo legible de la codificación de un programa implica: (1) Usar denominaciones significativas para las variables y procedimientos, (2) Indentar la codificación del programa según los niveles de ejecución que tengan asociadas las instrucciones<sup>79</sup>.

La legibilidad comprende la semántica utilizada para definir los nombres de las variables, las palabras reservadas, nombre de procedimientos, de funciones, de constantes, de unidades y etiquetas que constituyen el cuerpo del código. Al declarar los calificativos de los distintos métodos o procedimientos se debe tener en cuenta que estos nombres guarden relación con la función que realizan.

### **5.2.2 Documentación del código**

La documentación consiste en hacer comentarios de todas las partes del código para que se pueda entender y detallar los componentes y las funciones que realizan cada una de las unidades del programa. A demás de esto se realiza un manual para mantenimiento que incluye una estructura global del código donde se enumera la forma como se encuentra organizada las unidades del programa.

### **5.2.3 Integración del proceso de codificación**

El grupo de trabajo está conformado por diferentes especialistas que aportan todos los conocimientos que se requieren para el desarrollo del software. El grupo a cargo de la codificación después de haber culminado con las tareas asignadas a cada uno de los miembros del equipo, se procede a integrar cada una de las partes del código para formar un todo que permita visualizar la terminación del software (*véase norma ISO/IEC 12207 proceso de desarrollo, tarea integración del software*).

## **5.3 Evaluación de la versión inicial**

La evaluación de la versión inicial se realiza antes de la presentación de la prueba modelo, para identificar en qué condiciones se encuentra el software después de haber integrado las fases descritas con los procesos que dieron lugar a la codificación. Esta evaluación pretende demostrar si hasta estos términos el sistema lleva una secuencia de los requerimientos suministrados por los usuarios, además si presenta errores cual es el grado de complejidad que estos manifiestan, ya que dependiendo del resultado de este estudio se determinará si los objetivos propuestos se están alcanzando (*véase norma ISO/IEC 12207 proceso de desarrollo, tarea codificación y pruebas del software*).



Figura 43. Evaluación de la versión inicial

<sup>79</sup> GALVIS PANQUEVA, Álvaro H. Ingeniería de Software Educativo. Pág. 218

### 5.3.1 Evaluación realizada por los expertos

La evaluación de la versión inicial es realizada por el grupo de trabajo (*véase la subfase de planificación del proceso de producción*) conformado por especialistas en diferentes áreas, que tienen como tarea realizar un conjunto de pruebas a través de formatos evaluativos, que dará lugar a un diagnóstico del estado actual de la versión inicial del software (*véase norma ISO/IEC 12207 proceso de desarrollo, tarea pruebas de calificación del software*).

### 5.3.2 Ajustes a la versión inicial

Al terminar la evaluación realizada por el grupo de trabajo, tomando como referencia el diagnóstico obtenido en la etapa anterior, se procederá a efectuar los ajustes precisos para lograr la corrección de los errores que se manifiestan en esta primera versión del software. Este ajuste conllevará a la disminución de las posibles fallas encontradas, que dará lugar a la preparación de una prueba modelo más sólida y efectiva (*véase norma ISO/IEC 12207 proceso de operación, tarea establecer procedimientos para la resolución de problemas*).

En el formato dado a continuación, el significado de cada una de las abreviaturas de calificación es el siguiente:

- **TA:** Totalmente de acuerdo                  -- **TD:** Totalmente en desacuerdo
- **AC:** De Acuerdo                  -- **NA:** No aplica
- **DA:** En desacuerdo

FORMATO EVALUACIÓN OBJETIVA DE EXPERTOS						
ASPECTOS PEDAGÓGICOS		TA	AC	DA	TD	NA
<i>Educativo</i>	<i>El objetivo del software está bien definido</i>					
	<i>Alcanza el objetivo propuesto</i>					
	<i>El propósito educativo se infiere claramente en el material</i>					
	<i>Resuelve una necesidad educativa</i>					
	<i>El modelo pedagógico seleccionado se evidencia claramente en el software</i>					
	<i>Emplea un enfoque pedagógico acorde a la necesidad de la población</i>					
	<i>Las actividades y los recursos didácticos se ajustan al modelo pedagógico seleccionado.</i>					
	<i>Facilita el desarrollo de los contenidos</i>					
<i>Motivación</i>	<i>El software está adaptado al entorno donde se desenvuelven los usuarios.</i>					
	<i>El uso del software resulta motivador</i>					
	<i>El contenido motiva y mantiene el interés</i>					
	<i>El diseño en pantalla motiva y mantiene el interés</i>					
<i>Atención</i>	<i>La calidad técnica ayuda a motivar y mantener el interés</i>					
	<i>Fomenta la iniciativa y el auto aprendizaje</i>					
	<i>El contenido logra mantener la atención</i>					
	<i>El diseño logra mantener la atención</i>					
	<i>La calidad técnica permite mantener la atención</i>					

<b>Interactividad</b>	<i>Los recursos didácticos del software permiten su manipulación</i>				
	<i>Posibilidad de controlar la secuencia del programa.</i>				
	<i>Navegación libre con posibilidad de retornar.</i>				
	<i>La interactividad es de fácil manejo para el tipo de usuario para el cual fue diseñado.</i>				
	<i>Facilidad de volver al menú principal.</i>				
	<i>Facilita la construcción activa de conocimiento en el estudiante, o es simplemente repetitiva</i>				
<b>Creatividad</b>	<i>Estimula al aprendizaje</i>				
	<i>Estimula procesos creativos y divergentes</i>				
	<i>Ayuda a aprender de los errores</i>				
	<i>Tiene originalidad</i>				
<b>ASPECTOS METODOLÓGICOS</b>		<b>TA</b>	<b>AC</b>	<b>DA</b>	<b>TD</b>
<i>Buen manejo de los recursos didácticos</i>					
<i>La metodología favorece que el usuario participe activamente en el aprendizaje</i>					
<i>Está fundamentada en una didáctica apropiada para lo que se desea enseñar</i>					
<i>Mantiene el interés por lograr los objetivos con un buen nivel de eficacia</i>					
<b>CONTENIDOS</b>		<b>TA</b>	<b>AC</b>	<b>DA</b>	<b>TD</b>
<i>La información mostrada es suficiente para comprender el tema.</i>					
<i>Posee un alto valor educativo.</i>					
<i>Es pertinente para la población objeto.</i>					
<i>La presentación es multimedial</i>					
<i>Es coherente con los objetivos que se buscan</i>					
<i>El lenguaje utilizado es coherente con los usuarios</i>					
<i>La información es clara y concisa.</i>					
<i>Los contenidos presentados están en conformidad con el currículo.</i>					
<i>Es atractivo</i>					
<i>El volumen de información es suficiente para la comprensión y dominio de los contenidos.</i>					
<i>La información es actualizada.</i>					
<i>Permite proponer y enfrentar situaciones de variado nivel de complejidad.</i>					
<i>Los niveles de complejidad están acorde con los usuarios.</i>					
<i>Tiene significado para los usuarios</i>					
<i>Es relevante para lo que se desea que el alumno aprenda.</i>					
<b>DISEÑO</b>		<b>TA</b>	<b>AC</b>	<b>DA</b>	<b>TD</b>
<i>La interfaz inicial estimula el interés por la exploración del software.</i>					
<i>Existe un mapa de navegación.</i>					
<i>La estructura del mapa de navegación está acorde con la estructura del software.</i>					
<i>La participación del usuario es activa y relevante.</i>					
<i>Existe una continuidad estándar en las interfaces.</i>					
<i>El tamaño y el tipo de letra permiten leer en forma rápida y comprensiva.</i>					
<i>Los iconos utilizados corresponden a los de la temática.</i>					
<i>Integración adecuada y agradable de los diversos medios (imagen, sonido, video, texto,</i>					

<i>audio).</i>				
<i>Las interfaces no están recargadas de información.</i>				
<i>El material se presenta atractivo para el estudiante.</i>				
<i>El diseño motiva a conocer el tema.</i>				
<i>Integración armónica y adecuada de los elementos gráficos y audiovisuales.</i>				
<i>Facilita el acceso a la información.</i>				
<i>Facilidad de uso del software.</i>				
<i>Existe un ícono que identifique la salida del programa</i>				
<i>El usuario tiene control de las ventanas</i>				
<i>Claridad en los iconos que se ofrecen al usuario en cualquier parte del programa.</i>				
<i>El programa ofrece las respuestas a las actividades en tiempo real.</i>				
<i>La forma de presentación de los contenidos y actividades motiva al estudiante.</i>				
<b>ASPECTOS TÉCNICOS</b>	<b>TA</b>	<b>AC</b>	<b>DA</b>	<b>TD</b>
<i>Hay un manual sobre la instalación y la puesta en marcha del programa.</i>				
<i>El manual es suficientemente completo para dar mantenimiento al programa.</i>				
<i>Se especifica las características mínimas necesarias para su correcta operación.</i>				
<i>Los usuarios a los que se dirige el software pueden utilizar el programa con facilidad y de forma independiente.</i>				
<i>Los profesores pueden utilizar el programa con facilidad.</i>				
<i>El programa aprovecha los recursos que brinda el computador.</i>				
<i>Facilidad para el usuario de iniciar el programa sin ayuda.</i>				
<i>Posee errores de funcionamiento</i>				
<i>La calidad del audio es buena.</i>				
<i>El tiempo de acceso al programa es corto.</i>				
<i>Permite alternar la utilización de diferentes periféricos (teclado, ratón, joystick...) para la interacción con el programa.</i>				
<i>El código fuente está disponible.</i>				
<i>Calidad del programa respecto a la utilización del audio, imágenes estáticas, en movimiento y las animaciones.</i>				
<i>El programa permite ser utilizado sin conocimientos previos en informática.</i>				
<i>Es fácil de instalar y desinstalar.</i>				
<i>Identificación del programa (título de la aplicación, los datos del autor o empresa).</i>				
<b>ACTIVIDADES</b>	<b>TA</b>	<b>AC</b>	<b>DA</b>	<b>TD</b>
<i>Tienen relación directa con los objetivos y contenidos.</i>				
<i>Son motivadoras.</i>				
<i>Fomentan la creatividad y la exploración.</i>				
<i>Da el programa información sobre los errores cometidos.</i>				

Formato 17. Ejemplo de un formato de la evaluación realizada por expertos.

## **5.4 Presentación de la prueba modelo**

La prueba modelo es la primera versión que se obtiene al culminar el diseño y desarrollo del software, esta presentación se realiza para verificar si el software cumplió con las especificaciones planteadas en la necesidad educativa, para esto se selecciona un grupo representativo de la población objeto de estudio a quienes se les mostrará y aplicará el prototipo.

La presentación de la prueba modelo incluye dos etapas indispensables que conjugadas entre sí permiten la aplicación de la prueba. Éstas son:

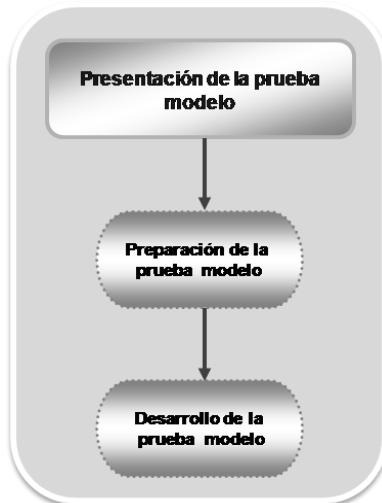


Figura 44. Presentación de la prueba modelo

### **5.4.1 Preparación de la prueba modelo**

La prueba piloto incluye la selección de la muestra, el diseño y prueba de los instrumentos de recolección de información, y el entrenamiento de quienes van administrar la prueba del material<sup>80</sup>.

### **5.4.2 Desarrollo de la prueba modelo**

Al terminar la elaboración de la prueba modelo se efectúa la ejecución de esta prueba a la población seleccionada, iniciando con la interacción del usuario con la primera versión del software para luego realizar una prueba escrita donde se medirá la interactividad, la navegabilidad, el diseño de las interfaces del software y el dominio que presentan los estudiantes al manipular la aplicación (*véase norma ISO/IEC 12207 proceso de operación, tarea establecer procedimientos para probar el producto*).

## **5.5 Evaluación de la prueba modelo**

Al conocer los resultados de la evaluación desarrolladas a la prueba modelo se reúne el grupo de trabajo para realizar un análisis de las evaluaciones efectuadas, en donde se determinará dependiendo de los errores, si es necesario realizar ajustes al software.

La subfase evaluación de la prueba modelo (*véase norma ISO/IEC 12207 proceso de desarrollo, tarea pruebas de calificación del sistema*) tiene dos fases, una de estas depende de la otra para su ejecución, las cuales serán especificadas a continuación:

---

<sup>80</sup> Ibíd. Pág. 78

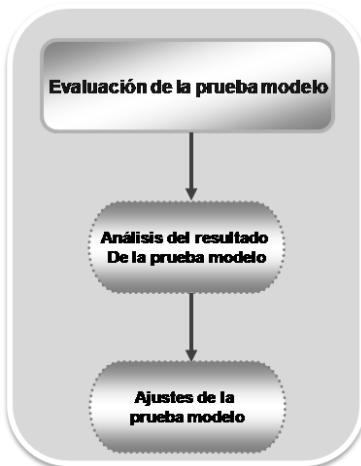


Figura 45. Evaluación de la prueba modelo

### 5.5.1 Análisis del resultado de la prueba modelo

Una vez concluido las pruebas efectuadas de la prueba modelo a la población seleccionada, es determinante llevar a cabo un estudio a fondo de los resultados obtenidos en dicha prueba para establecer si se encontraron errores, dificultades u omisiones de detalles al interactuar los usuarios con el software y a través de la prueba escrita. Si el resultado es positivo es decir sin fallas el software está dispuesto para ser aplicado a toda la población objeto de estudio.

### 5.5.2 Ajustes de la prueba modelo

Después de obtener el diagnóstico de la prueba modelo y tener claridad de los problemas presentados en el software se hace indispensable realizar ajustes o correcciones necesarias para dar solución a las dificultades detectadas en esta primera versión. Se debe tener en cuenta que esta etapa es opcional, es decir que depende del análisis del resultado de la prueba modelo (*véase norma ISO/IEC 12207 proceso de operación, tarea establecer procedimientos para la resolución de problemas*).

## 5.6 Manual de usuario

Pequeño libro que recibe el usuario junto con el sistema de composición, ya sea con referencia al hardware o al software, que le permite aprender las particularidades del equipo o del sistema y la forma de utilizarlo. Una vez familiarizado con el mismo, pasa a ser un instrumento de consulta para cuando se ha olvidado algún aspecto o se presenta alguna duda<sup>81</sup>.

El manual de usuario es de vital importancia al momento de desarrollar un software debido a que este es el soporte escrito que permite guiar a los usuarios en la utilización del software, estos detalles se explicarán con más claridad en las siguientes categorías:

---

<sup>81</sup> <http://www.proyectosfindecarrera.com/definicion/manualdelusuario.htm>

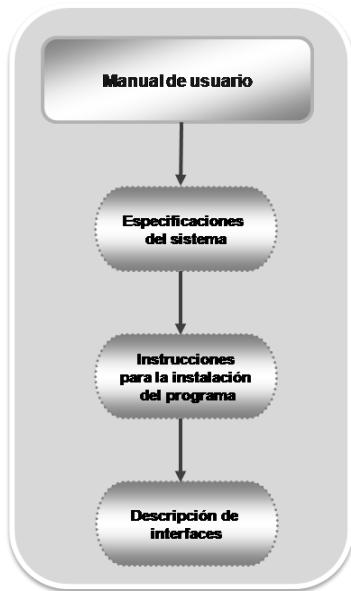


Figura 47. Manual de usuario

### **5.6.1 Especificaciones del sistema**

Antes de la utilización de un software es importante que los usuarios conozcan los detalles que se deben tener en cuenta para que la aplicación funcione correctamente, estos detalles o especificaciones corresponden a las características del hardware y software (*véase norma ISO/IEC 12207 proceso de adquisición, tarea definición del sistema de requerimientos; también vea el proceso de desarrollo, las tareas requisitos del sistema y diseño de la arquitectura del sistema*).

### **5.6.2 Instrucciones para la instalación del programa**

Es indispensable que los usuarios que utilizarán el software conozcan cuales son los procedimientos a tener en cuenta para realizar la instalación adecuada de la aplicación, después de haber logrado la instalación exitosa del programa se debe explicar paso a paso el ingreso al software.

### **5.6.3 Descripción de las interfaces**

La descripción de las interfaces es muy importante en un manual de usuario, debido a que se explica detalladamente las acciones que puede realizar el usuario. En la descripción que se hace se incluye los mensajes, los botones y las características que ofrece cada una de las ventanas del software.

## PUNTOS CLAVE

- En la fase de producción se agrupan todos los materiales obtenidos en el desarrollo de las fases anteriores y se realiza la implementación de todos los documentos que serán codificados por un especialista en programación permitiendo el desarrollo final del software.
- Es necesario seleccionar una herramienta o lenguaje de programación que sea acorde con las especificaciones del software desarrollado, es importante tener en cuenta que la persona encargada de la codificación domine la herramienta o lenguaje seleccionado.
- En la etapa de codificación se aplican unos parámetros para que la programación tenga una presentación correcta, entendible y con buen estilo, lo que permitirá fácilmente el mantenimiento del software para poder corregir los errores y modificar algún requisito cuando sea necesario.
- La evaluación de la versión inicial se realiza antes de la presentación de la prueba modelo, para identificar en qué condiciones se encuentra el software y dependiendo del resultado de este estudio se determinará si los objetivos propuestos se están alcanzado.
- La prueba modelo es la primera versión que se obtiene, esta presentación se realiza para verificar si el software cumplió con las especificaciones planteadas en la necesidad educativa, para esto se selecciona un grupo representativo de la población objeto de estudio a quienes se les mostrará y aplicará el prototipo.
- La evaluación de la prueba modelo consiste en realizar un análisis por parte de un grupo de trabajo del resultado de las evaluaciones efectuadas, en donde se determinará dependiendo de los errores, si es necesario realizar ajustes al software.
- El manual de usuario es de vital importancia al momento de desarrollar un software debido a que este es el soporte escrito que permite guiar a los usuarios en la utilización del software.

## REFERENCIAS

- GALVIS PANQUEVA, Álvaro H. Ingeniería de Software Educativo. Pág. 218
- <http://www.proyectosfindecarrera.com/definicion/manualdelusuario.htm>

**Capítulo VI**

**APLICACIÓN**

## 6. Fase V: Aplicación

Al finalizar todas las fases anteriores del modelo MODESEC, donde cada fase aporta para la configuración del software. La fase I *diseño educativo*, como su nombre lo dice está referida a toda la parte educativa del software como lo es el análisis de la necesidad educativa, a su vez es el punto de partida, el planteamiento de los fines educativos, del sistema de competencias que es el eje central del modelo, el diseño de los contenidos, el diseño pedagógico y el diseño de aprendizaje, los cuáles serán abordados en el software. La fase II del modelo es la parte llamativa del software, es decir la parte atractiva del software, debido a que es en esta fase donde se diseñan cada una de las interfaces que harán parte del programa.

La fase III es la encargada del análisis y modelado del sistema, es en esta fase donde se selecciona el proceso de desarrollo, se identifican los requerimientos, se realizan los casos de uso, el modelado de datos, todo esto para seguir a la fase IV, donde se integran todos los componentes que se necesitan para completar el software, es decir que en esta fase el software queda listo para la siguiente y última fase que permitirá que el software sea probado y luego ser utilizado.

La fase de aplicación es la etapa final donde se mostrará el software terminado y funcionando a cabalidad. Esta comprende la aplicación del software a toda la población objeto de estudio en condiciones normales del aprendizaje para comprobar si el programa en realidad cumplió con los requisitos establecidos en los objetivos propuesto en la necesidad educativa. En esta fase se realiza el mantenimiento del software, ya sea para actualizar, corregir o cambiar requisitos o funciones del software.



Figura 48. Fase V. Aplicación

### 6.1 Utilización del software

Para utilizar un software es necesario conocer cuáles son las condiciones que se establecen para que los usuarios puedan interactuar sin mayor dificultad con el software, esto permitirá que la aplicación tenga el efecto esperado para lo cual fue desarrollada.

La subfase utilización del software permite plantear el entorno propicio para el manejo del software, estas condiciones se establecen de la siguiente forma:

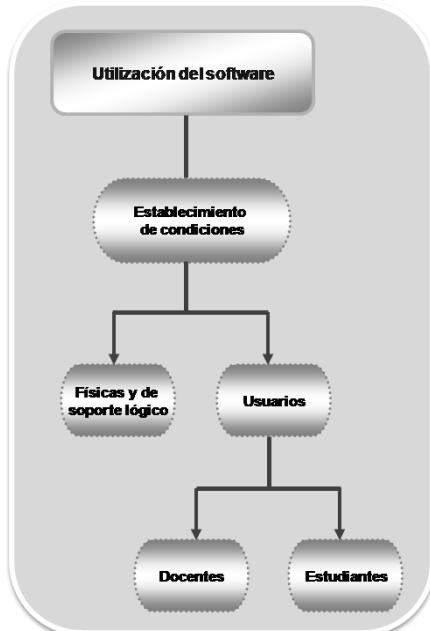


Figura 49. Utilización del software

### 6.1.1 Establecimiento de condiciones

(Véase norma ISO/IEC 12207 proceso de desarrollo, tareas requisitos del sistema, diseño de la arquitectura del sistema, requisitos del sistema)

**6.1.1.1 Físicas y de soporte lógico:** Las condiciones físicas y de soporte lógico hace referencia a la infraestructura de la sala de cómputo, la cual debe estar adecuada con las especificaciones del sistema (hardware-software) establecidas en el manual de usuario.

#### 6.1.1.2 Usuarios

- **Docentes:** Antes de realizarle la presentación de la aplicación a los estudiantes, como primera medida es conveniente preparar a los docentes para la correcta utilización del software, de tal forma que estos puedan entender las funciones y características que contiene la aplicación, para que resuelvan las dudas e inconvenientes que se les puedan presentar a los estudiantes al momento de utilizar el software.
- **Estudiantes:** Los estudiantes son los que permitirán evidenciar si el software ha cumplido con la función principal para lo cual ha sido creado, por tal motivo antes de que estos interactúen con la aplicación se debe hacer una preparación de la prueba para motivar y apoyar a los estudiantes en la utilización del software. A parte de la prueba computacional se debe hacer una escrita donde se evalúe los conocimientos adquiridos a partir de la interacción con la aplicación.

## 6.2 Obtención de resultados

En esta fase se realiza el análisis de los resultados obtenidos en la etapa de utilización del software. A partir de este estudio se logrará determinar si el software aplicado fue eficaz, eficiente y cuenta con los requisitos necesarios para seguir con su utilización en el aula de clases o por lo contrario, éste requiere que se realicen ajustes necesarios para cumplir con la necesidad para lo cual fue diseñado.

La etapa de obtención de resultados es la fase en donde se visualizará la importancia o impertinencia del software en el proceso de enseñanza-aprendizaje pero para determinar estas conclusiones se deben realizar los componentes siguientes:

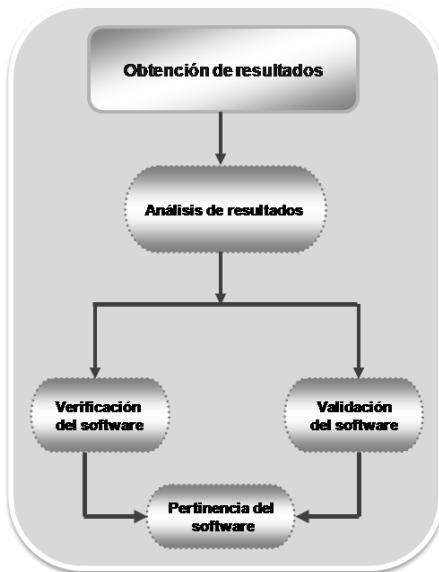


Figura 50. Obtención de resultados

### 6.2.1 Análisis de resultados

**6.2.1.1 Verificación del software:** La verificación de un software es un proceso continuo que conlleva a la evaluación de cada fase que contiene el desarrollo de un software, ésta evaluación es realizada por un grupo de personas que fueron seleccionadas en la fase de Planificación del Proceso de Producción; para llevar a cabo este estudio de revisión de calidad de los procesos es necesario implementar categorías que permitan efectuar una verificación eficaz.

Estas categorías se desarrollan partiendo de la necesidad educativa (véase el análisis de la necesidad educativa), la cual permite decir si el sistema cumple con las funciones requeridas para suplir dicha necesidad. Esto conlleva a saber si el software incluye todas las funciones o requisitos solicitados por los usuarios. Es indispensable realizar una valoración del presupuesto y la tecnología disponibles, que permitan conocer si los requisitos dispuestos sean los adecuados para tal propósito.

**6.2.1.2 Validación del software:** La validación de un software permite saber si los requisitos desarrollados en el sistema cumplen con las especificaciones dispuestas por los usuarios. Esto ayudará a conocer si el programa presentó algún error el cual podrá ser solucionado con anticipación evitando así costos innecesarios.

**6.2.1.3 Pertinencia del software:** Despues de haber analizado los resultados obtenidos se tomarán decisiones acerca de la pertinencia del software, teniendo en cuenta si los resultados del análisis son convenientes no se hará ninguna modificación o por lo contrario se debe realizar los ajustes necesarios para el funcionamiento exitoso del software. En esta fase también se debe prever los cambios que pueda tener el currículo y por ende los objetivos de aprendizaje validando la vigencia de la necesidad que dio origen al software.

### 6.3 Mantenimiento

Esta fase es complementaria a la fase *Obtención de resultados*, debido a que en la fase de *Mantenimiento* (véase norma ISO/IEC 12207 *proceso de desarrollo, tarea asegurar la operación y mantenimiento del software entregable, vea también el proceso de mantenimiento*), es donde se realiza todas las mejoras, las

modificaciones o actualizaciones que requieren las actividades, los procesos o cambios en los requerimientos del software para verificar si se cumplió con la necesidad para lo cual fue diseñado.

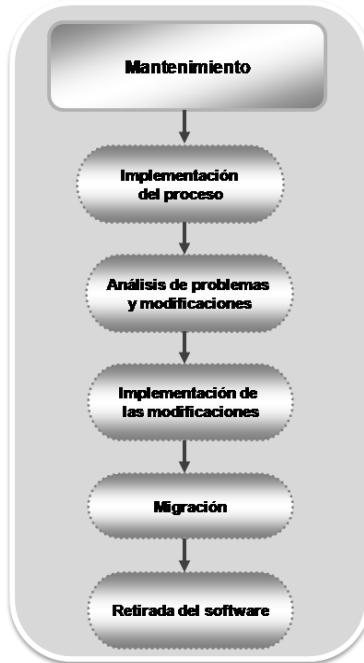


Figura 51. Mantenimiento

### 6.3.1 Implementación del proceso

Esta fase incluye todos los procesos que se necesitan para preparar, documentar y ejecutar las actividades que se necesitan para el mantenimiento. La persona encargada de mantenimiento debe determinar los procedimientos para realizar los seguimientos a los informes de los problemas que se presentan en el software.

### 6.3.2 Análisis de problemas y modificaciones

Es en esta fase donde se analiza de una forma detallada los problemas o las fallas que se presentan en el software, el encargado del mantenimiento debe comprobar el problema, después de esto se debe realizar las alternativas de solución, haciendo la documentación correspondiente del problema y las modificaciones realizadas.

### 6.3.3 Implementación de las modificaciones

En la implementación se lleva a cabo las acciones correctivas y las pruebas necesarias de las modificaciones que se requiere para que las funciones del software funcionen cabalmente y cumpla con los propósitos que llevaron a su creación. Todas las actividades realizadas contienen su respectiva documentación.

### 6.3.4 Migración

La migración se lleva a cabo cuando se requiere trasladar los datos, procesos de un entorno de operación del software antiguo a otro completamente nuevo, para ello se ejecuta un plan que incluye en primera instancia la preparación, la documentación y la ejecución. Al realizar este proceso se deberá notificar a los usuarios las modificaciones que se hicieron.

### **6.3.5 Retirada del software**

Después de hacer los diferentes estudios con respecto a los problemas presentados se ejecutará un plan, el cual será preparado y documentado, para tomar la decisión de retirar totalmente o parcialmente el software según sea el caso de igual manera se harán los avisos correspondientes a los usuarios.

### **PUNTOS CLAVE**

- La fase de aplicación es la etapa final donde se mostrará el software terminado y funcionando a cabalidad. Esta comprende la aplicación del software a toda la población objeto de estudio en condiciones normales del aprendizaje para comprobar si el programa en realidad cumplió con los requisitos establecidos en los objetivos propuesto en la necesidad educativa.
- La subfase utilización del software permite plantear el entorno propicio para el manejo del software, para lo cual se establecen unas condiciones.
- La etapa de obtención de resultados es la fase en donde se visualizará la importancia o impertinencia del software en el proceso de enseñanza-aprendizaje pero para determinar estas conclusiones se deben realizar ciertos procesos.
- La subfase de Mantenimiento es donde se realiza todas las mejoras, las modificaciones o actualizaciones que requieren las actividades, los procesos o cambios en los requerimientos del software para verificar si se cumplió con la necesidad para lo cual fue diseñado.

## **Capítulo VII**

### **APLICACIÓN Y VALIDACIÓN DEL MODELO MODESEC-ISO/IEC 12207**

## **7.1 COMPARACIÓN CON OTROS MODELOS**

En este apartado primero se procederá a realizar una breve descripción de los modelos existentes en la actualidad para el desarrollo de software educativo y luego se presentarán las diferencias entre el modelo propuesto y los ya existentes a modo de cuadro comparativo.

### **7.1.1 Modelos para el desarrollo de software educativo**

#### **7.1.1.1 Ingeniería de software educativo con modelaje orientado por objetos: un medio para desarrollar micromundos interactivos<sup>82</sup>.**

La integración de Ingeniería de Software Educativo (ISE), con el modelaje O. O, permite hacer uso efectivo de la información recolectada en la fase de análisis y diseño educativo; por esta razón es necesario su inclusión en todas las etapas del ciclo de desarrollo y así unificar los términos en los que se habla en cada etapa, estableciendo un modelo del mundo del problema y de su comportamiento; de este modo se hace referencia a objetos presentes en el modelo, extendiendo así su funcionalidad. Al llegar a la implementación, los resultados obtenidos se transcriben al lenguaje de programación escogido, cambiando la sintaxis en que se expresa el modelo, más no la semántica.

---

<sup>82</sup> Gómez, R., Galvis, A., Mariño, O. Ingeniería de software educativo y modelaje orientado a objetos. Un medio para desarrollar micro mundos interactivos. Universidad de Los Andes. Bogotá. Colombia. 2000.

#### **7.1.1.2 Metodología Dinámica para el Desarrollo de Software Educativo<sup>83</sup>.**

La metodología está compuesta por cuatro fases (Diseño Educativo, Producción, Realización e Implementación.) y un eje transversal que es la Evaluación. No se requiere la culminación de una fase para pasar a la otra, es posible obtener rápidamente un prototipo que permita hacer validaciones parciales y correcciones de ser requeridas.

#### **7.1.1.3 Metodología THALES<sup>84</sup>**

La metodología de THALES, representa un modelo híbrido no lineal. Esta metodología consta de seis fases: Planeación, Diseño, Producción, Prueba Piloto, Evaluación y Mejoramiento. Habiendo un ciclo en las fases diseño-producción-prueba piloto y luego culminada la versión 1 un ciclo evaluación-mejoramiento. En futuras evaluaciones puede tomarse la decisión de revisar los documentos desde el principio, es decir retomar la fase de planeación.

#### **7.1.1.4 MOSCA - Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica<sup>85</sup>.**

---

<sup>83</sup> Molina, R. Modelo de desarrollo de software educativo. Argentina. 2006.

<sup>84</sup> Marqués, P. Selección y Uso de Recursos Multimedia. Diseño de Actividades. Usos en el Aula Informática. 1999. Disponible en: <http://dewey.uab.es/pmarques/interven.htm>. Consultado el 03 de abril de 2009.

<sup>85</sup> Mendoza, L., Pérez, M., Grimán., A. Prototipo de modelo sistémico de calidad (MOSCA) del software. Universidad Simón Bolívar. Caracas Venezuela. 2004.

Este modelo mide la calidad sistémica de una organización desarrolladora de software partiendo de la calidad de su producto y de la calidad del proceso de desarrollo del mismo. El modelo MOSCA especifica los procesos que se deben mejorar en la empresa y las características que no son satisfechas por el producto de software desarrollado.

## ANÁLISIS COMPARATIVO

TABLA COMPARATIVA				
Indicadores de comparación	Modelo propuesto	Modelos existentes		
	<b>MODESEC</b> (Modelo para el desarrollo de software educativo basado en competencias)	Metodología Dinámica para el Desarrollo de Software Educativo	Metodología THALES	Ingeniería de software educativo con modelaje orientado por objetos: un medio para desarrollar micro mundos interactivos
Adaptado al sistema de competencias colombiano				
Personas que intervienen en el proyecto (menos de 10 personas)				
Fases e implementación que comprenden los distintos modelos	Cinco	Cuatro	Seis	Cinco
Norma o Estándar				
Disponibles software hechos con cada modelos				
Origen	Colombia	Argentina	Venezuela	Colombia

Tipos de modelos	Lineal solapado	Lineal solapado	Cíclico	Cíclico
------------------	-----------------	-----------------	---------	---------

Tabla 5. Tabla comparativa entre los modelos de diseño de software educativo estudiados.

Se puede apreciar en la tabla que los otros modelos estudiados no cuentan con indicaciones precisas acerca de cómo involucrar las competencias que se esperan que el estudiante desarrolle con el proceso de diseño del software educativo. En el caso de los modelos extranjeros, se deduce que no soportan esta capacidad debido a estar inmersos en el contexto educativo local de cada país en el que se han propuesto.

Otros modelos como el Mosca y el O.O se basan más en actividades ingenieriles que le van a dificultar a los docentes de las instituciones de educación colombianas utilizarlos para implementar soluciones a las necesidades educativas que en ellas se puedan presentar.

## 7.2. APLICACIÓN Y VALIDACIÓN DEL MODELO

El modelo propuesto en este libro ha sido validado con éxito en los trabajos de grado de la línea de Desarrollo de Software educativo que han sido orientados y/o dirigidos por el grupo de investigación EDUPMEDIA. Disponibles en CD se encuentran Edulógica, para refuerzo de la lógica matemática; Brújula, para tratar el programa de la disgrafía en niños y Zenulandia, para la enseñanza de la cultura Zenú en Colombia.

Otra validación ha sido la realizada en la asignatura de Diseño de Software Educativo, de la cual se disponen en CD las aplicaciones: Lenguaje Wayuunaiki, La conquista de América, Los Mayas, Learning mandarín y Los Aztecas. Este último será el utilizado en el presente apartado como caso de estudio de la aplicación del modelo<sup>86</sup>.

A continuación se presentará la descripción de un software educativo realizado con el modelo propuesto.

FICHA DEL SOFTWARE	
TÍTULO	SOFTWARE AZTECA
Objetivo General	
Desarrollar un software educativo que facilite a los estudiantes de noveno grado de las instituciones educativas	

<sup>86</sup> Caro, M., Alvarez, L., Ome., L. Software Azteca. Edupmedia. Universidad de Córdoba. Montería. 2008.

de Córdoba, conocer y aprender de las grandes culturas Americanas en especial la civilización Azteca.
<b>Objetivos Específicos</b>
Despertar el interés y motivación de los estudiantes para aprender sobre la civilización Azteca. Mejorar el grado de comprensión de las temáticas referentes a la civilización Azteca. Lograr que los estudiantes conozcan la historia y evolución del imperio Azteca, desde su fundación en 1325 hasta su caída en 1521.
<b>CREADORES</b>
LUIS FERNANDO ÁLVARES FAJARDO LILIANA OME MORELO ASESORÍA: MANUEL CARO PIÑERES UNIVERSIDAD DE CÓRDOBA 2008

Tabla 6. Ficha de presentación general del software.

### 7.2.1 Estudio de la necesidad educativa

ANÁLISIS DE LA NECESIDAD EDUCATIVA	
<b>Tipo: sentida</b>	Según Burton y Merryl, la necesidad a tratar con el diseño de nuestro producto es de tipología sentida, ya que, está temática fue sugerida por la institución educativa, en base a las falencias que presentan los estudiantes de noveno grado en las diferentes instituciones del departamento de Córdoba.
<b>Identificación del aprendizaje ideal</b>	<ul style="list-style-type: none"> <li>✓ Ubicación geográfica de la civilización azteca</li> <li>✓ Historia de la fundación y consolidación del imperio Azteca</li> <li>✓ Aspectos culturales de los Aztecas (economía y religión)</li> <li>✓ Lenguaje Azteca. (diccionario)</li> <li>✓ Conquista y caída del imperio Azteca.</li> </ul>
<b>Población</b>	<p><b>Rango de edad:</b> Las edades de los estudiantes de noveno grado de las instituciones educativas del departamento de Córdoba oscilan entre los 13 y 17 años de edad.</p> <p><b>Escolaridad:</b> Estudiantes de noveno grado de las instituciones educativas del departamento de Córdoba.</p> <p><b>Conocimiento que posee:</b> Los estudiantes deben tener conocimiento de las temáticas que en los colegios se dan antes de esta.</p> <p><b>Intereses y expectativas: (Población)</b></p> <ul style="list-style-type: none"> <li>➤ Desarrollar actitudes que le permitan manejar el software.</li> <li>➤ Que el software les facilite la comprensión de la temática.</li> </ul>

	<p><b>Intereses y expectativas: (Creadores)</b></p> <ul style="list-style-type: none"> <li>➤ Desarrollar actitudes en los estudiantes que le permitan manejar el software.</li> <li>➤ Que los estudiantes a través del software conozcan parte de la cultura suramericana, como es la civilización azteca, de una manera más interactiva.</li> </ul>
<b>Área de formación</b>	<b>Área del saber:</b> Sociales e Historia
	<b>Área de contenido:</b> Grandes culturas Americanas (los Aztecas)
<b>Estado actual</b>	<p><b>Diagnóstico:</b></p> <p>A partir observaciones y encuestas a docentes y estudiantes, de las distintas instituciones educativas del departamento de Córdoba. Queda al descubierto que en estas se vienen presentando problemas de aprendizaje en las distintas áreas del saber, uno de los problemas más grandes, se da en el área de sociales e historia donde los estudiantes se muestran cansados, fatigados o simplemente desinteresados, durante el desarrollo de las actividades académicas en las que trata, sobre las grandes culturas Americanas, en especial la civilización Azteca.</p>
<b>Necesidad</b>	Mediante el diagnóstico realizado nos podemos dar cuenta del desinterés de los estudiantes por el estudio de las grandes culturas Pre-Hispánicas Americanas y en especial la civilización Azteca.
<b>Causas</b>	<ul style="list-style-type: none"> <li>➤ La falta de material didáctico para el desarrollo de estas clases.</li> <li>➤ Los grandes volúmenes de contenido que tienen estas temáticas.</li> <li>➤ La aplicación por parte del docente de una metodología, muy poco atractivas para los estudiantes.</li> <li>➤ La falta de creatividad y recursividad de los maestros para el desarrollo de su actividad académica.</li> </ul>
<b>Soluciones</b>	<ul style="list-style-type: none"> <li>➤ Diseñar y crear herramientas didácticas, ya sean software educativo, videos, materiales multimediales o entre otros, con el fin de que las clases sean más amenas.</li> <li>➤ Incluir en este software educativo o material multimedial el contenido de esta temática.</li> <li>➤ El uso de una metodología que sea del agrado de los estudiantes.</li> <li>➤ Capacitar a los docentes con el fin de que sean más creativos al momento de planear sus clases.</li> </ul>
<b>Conocimientos y habilidades que debe tener el estudiante</b>	<p><b>Preconceptos:</b> Tener claro los contenidos previos a esta temática.</p>
	<p><b>Precondiciones:</b></p> <ul style="list-style-type: none"> <li>➤ Manejo básico del computador</li> <li>➤ Tener habilidades psicomotoras.</li> </ul>

<b>Justificación</b>	
<p>Teniendo en cuenta que la necesidad encontrada requiere de nuevas alternativas de solución, y que dentro de estas posibles soluciones esta capacitar a los docentes para el uso de nuevas metodologías, para ello las instituciones gastarían tiempo y dinero con los cuales no cuenta actualmente, planteamos como la solución más viable la creación de un software educativo sobre la civilización Azteca en el cual los conocimientos de los estudiantes acerca del tema sean proporcionados de una manera más comprensible; a la vez se aprovecha los beneficios de las nuevas tecnologías; gracias a la interacción del estudiante con el software educativo se buscará lograr una mayor motivación y atención en ellos, pues podrán ir eligiendo la temática que más capte su atención para esto se utilizaran imágenes e interfaces más llamativas, saliéndose del ambiente tradicional del salón de clases.</p>	

Tabla 7. Ficha para diligenciar los datos de la necesidad educativa.

### 7.2.2 Diseño del sistema de competencias

En este apartado se muestra una vista parcial de las competencias tratadas en el software.

DISEÑO DE COMPETENCIAS	
Objetivos	Normas
<b>Enunciados 1:</b> Reconoce la ubicación geográfica del imperio azteca a lo largo de su historia.	<p><b>Contexto:</b> cuando se le solicite la ubicación geográfica del imperio Azteca.</p> <p><b>Recursos:</b></p> <ul style="list-style-type: none"> <li>✓ Mapa de centro América.</li> <li>✓ Imágenes del imperio azteca.</li> <li>✓ Videos del imperio azteca.</li> <li>✓ Software de la cultura Azteca.</li> </ul>
<b>Elementos</b>	<b>Indicadores</b>
1. Identifica la ubicación geográfica de los aztecas en el siglo XIV y principios de siglo XV.	<p>1.1. Relata la ubicación geográfica de los aztecas en esta época.</p> <p>1.2. Reconoce en el mapa de centro América la ubicación geográfica de los aztecas.</p> <p>1.3. Conocer la ciudad principal donde se ubicaron los aztecas mediante imágenes.</p>
2. Identifica la ubicación geográfica de los aztecas a mediados del siglo XV.	<p>2.1. Relata la ubicación geográfica de los aztecas en esta época.</p> <p>2.2. Reconoce en el mapa de centro América la ubicación geográfica de los aztecas.</p> <p>2.3. Conocer la ciudad principal donde se ubicaron los aztecas mediante imágenes.</p>
3. Identifica la ubicación geográfica de los aztecas a finales del siglo XV.	<p>3.1. Relata la ubicación geográfica de los aztecas en esta época.</p> <p>3.2. Reconoce en el mapa de centro América la ubicación geográfica de los aztecas.</p> <p>3.3. Conocer la ciudad principal donde se ubicaron los aztecas mediante imágenes.</p>
4. Identifica la ubicación geográfica de los aztecas a principios del siglo XVI.	<p>4.1. Relata la ubicación geográfica de los aztecas en esta época.</p> <p>4.2. Reconoce en el mapa de centro América la ubicación geográfica de los aztecas.</p>

	4.3. Conocer la ciudad principal donde se ubicaron los aztecas mediante imágenes.
CONCEPTOS	<ol style="list-style-type: none"> <li>1. Generalidades.           <ul style="list-style-type: none"> <li>✓ Historia</li> <li>✓ Civilización</li> <li>✓ Aztecas</li> </ul> </li> <li>2. Ciudades.           <ul style="list-style-type: none"> <li>✓ Texcoco</li> <li>✓ Tenochtitlan</li> <li>✓ Azcapotzalco</li> <li>✓ Tepanecas</li> </ul> </li> </ol>

Tabla 8. Ficha para diligenciar los aspectos de las competencias.

### 7.2.3 Diseño de contenidos

En este caso sólo se muestran los contenidos que harán parte de las estrategias de la competencia que se ha descrito en el párrafo anterior. Recordemos que la fuente de los términos e ideas que aquí se explican es la sección de *Conceptos* del formato para la formalización de competencias. Ver apartado 4.2.

DISEÑO DE CONTENIDOS		
CONCEPTOS DE LAS COMPETENCIAS		
CONCEPTOS	CARACTERÍSTICAS	DEFINICIÓN
Aztecas	<ul style="list-style-type: none"> <li>- Tribus guerreras de filiación nahua.</li> <li>- Fundó a México-Tenochtitlán.</li> <li>- Estados más extensos de Mesoamérica.</li> <li>- Cazadores y recolectores.</li> </ul>	<ul style="list-style-type: none"> <li>-Fueron un tribus guerreras de filiación nahua que fundó a México-Tenochtitlán convirtiéndose en el centro de uno de los Estados más extensos de Mesoamérica eran cazadores y recolectores también tenían tierras de producción para cubrir todos los gastos de su población.</li> </ul>
Tepanecas	<ul style="list-style-type: none"> <li>-Pueblo prehispánico de origen chichimeca.</li> <li>-Se instaló en la Cuenca de México a mediados del siglo XII de nuestra Era.</li> <li>-Se ubicaron en el año 1012 en la región poniente del lago Texcoco.</li> </ul>	<ul style="list-style-type: none"> <li>-Es el nombre de un pueblo prehispánico de origen chichimeca, que se instaló en la Cuenca de México a mediados del siglo XII de nuestra era y se ubicaron en el año 1012 en la región poniente del lago Texcoco.</li> </ul>

Tabla 9. Ficha para diligenciar el diseño de los contenidos.

### 7.2.4 Diseño pedagógico

De acuerdo con lo especificado en el modelo de desarrollo, primero se selecciona un modelo pedagógico con base en las necesidades educativas y luego se verifican las características del modelo que son susceptibles para incluirlas en el software educativo.

- ✓ **Teoría del Modelo seleccionado para el software Azteca:**  
Constructivismo social y aprendizaje significativo: Es un modelo de enseñanza el cual nos dice que el individuo aprende mientras convive o interactúa con el medio en donde habita y los sujetos que lo rodean los cuales ayudan a que este construya un nuevo conocimiento.
  
- ✓ **Principios de esta teoría aplicables al software Azteca:**
  - El aprendizaje es responsabilidad del estudiante.
  - Se debe presentar un choque cognitivo para retar al estudiante y en cierta forma hacerlo desaprender ciertos conceptos que no son los correctos.
  - Los nuevos conceptos deben ser significativos esto es relacionado con su experiencia con otros. Que ya posea que le sirvan de ancla para el nuevo aprendizaje.
  - El individuo aprende cuando interactúa con otros:  
Los individuos se reúnen en grupos con otros individuos a explorar y interactuar con el software educativo de la cultura azteca, en donde los individuos se pueden ayudar mutuamente en caso de que si uno de los individuos no entiende o presenta problemas al momento de interactuar con el Software los demás individuos del grupo le pueden explicar o resolver la inquietud que tenga acerca de la temática o el uso de este y así adquirir nuevos Conocimientos.

## 7.2.5 Diseño de las actividades de aprendizaje

Recordemos que para diseñar las actividades de aprendizaje se deben tener en cuenta las competencias especificadas y el modelo pedagógico seleccionado.

### 7.2.5.1 Problema

Esta ventana materializa la premisa estipulada en el diseño pedagógico donde se presenta una situación problemática que genere un *choque cognitivo* para retar al estudiante y en cierta forma hacerlo desaprender algunos conceptos que no son los correctos.



*Figura 52. Ventana de presentación de la situación problemática (Software Azteca – Universidad de Córdoba. 2008).*

En esta ventana a modo se historia se le introduce en la temática y el problema a resolver se le es presentado en un video y es ambientado con una animación.

### **6.2.5.2 Estrategias**

En este caso el modelo pedagógico determina la utilización de estrategias que le darán al estudiante la información que necesita para resolver el problema que se le plantea al inicio del programa. Estas estrategias son de tipo exploratorio y no implican secuencialidad ni obligatoriedad de visita, esto se hace de este modo para ir en concordancia con el modelo seleccionado.

Cada estrategia cuenta con una sección para su evaluación. Esto se hace con el fin de dividir el problema grande presentado al inicio, en varios problemas pequeños más fáciles de solucionar por parte del estudiante

#### *Estrategia 1*

<b>Nombre:</b>	Ubicación Geográfica
<b>Objetivo:</b>	Dar a conocer el lugar en donde se ubicaron los aztecas a lo largo de su evolución.

Al desplegarse la ventana, aparece un mapa interactivo que contiene la ubicación de los aztecas, este presenta una línea de tiempo que muestra la ubicación del imperio a lo largo de la historia y cuando se haga clic en la línea de tiempo aparece la imagen de la ciudad y al lado de la imagen se muestra un cuadro de texto con la explicación de la ciudad y ubicación del imperio azteca. A su vez se acompaña de audio y recursos extras como archivos de texto con extensión PDF para quienes deseen profundizar más en el tema.



*Figura 53. Ventana de la estrategia para la ubicación geográfica (Software Azteca – Universidad de Córdoba. 2008).*

#### Estrategia 2

<b>Nombre:</b>	Emperadores Aztecas Ingles /español.
<b>Objetivo:</b>	Conocer el perfil de cada uno de los emperadores y su periodo al mando en este pueblo azteca.

En la ventana cuenta con las imágenes de cada uno de los emperadores en plano medio con su texto debajo con su periodo al mando del imperio. Cuando se dé clic en las imágenes de alguno de ellos aparece su imagen en plano general y se muestra una ficha con su perfil. La ventana está diseñada a modo de galería fotográfica y cuenta con el audio para ayudar a la correcta pronunciación del nombre de cada emperador. A su vez se acompaña de audio y recursos extras como archivos de texto con extensión PDF para quienes deseen profundizar más en el tema.



Figura 54. Ventana de la estrategia para los emperadores (Software Azteca – Universidad de Córdoba. 2008).

#### Estrategia 4

<b>Nombre:</b>	Calendario Ingles /español.
<b>Objetivo:</b>	Conocer el significado de cada una de las partes del calendario y su representación.

En esta ventana el usuario se encontrará con la imagen del calendario azteca, y cuando de clic en una de las partes del mismo, este despliega un texto con su significado y la ampliación del símbolo.



Figura 55. Ventana de la estrategia para el calendario (Software Azteca – Universidad de Córdoba. 2008).

#### 7.2.5.3 Evaluación

El usuario deberá leer las instrucciones del juego, en donde indica cómo funciona, este deberá hacer clic sobre el personaje de Hernán Cortés para empezar a caminar por el escenario, cuando llegue a uno de los guardias debe hacer clic sobre este, en donde este le arrojara una pregunta que el usuario debe responder con las opciones que le den y si acierta continua caminando y seguirán las preguntas hasta llegar al emperador Moctezuma y ganar la partida, pero si se equivoca en una de las preguntas aparece un cuadro donde le dice el aspecto en que falló y en el lado de este hay un botón que lo lleva a las ventanas donde se encuentra la información para que repase todo lo referente a la pregunta y vuelvas a intentarlo.

Recordemos que la evaluación general debe estar ligada a la situación problemática planteada al inicio del software y debe abarcar todos los elementos contenidos en las estrategias de estudio.



Figura 56. Ventana de la estrategia para la evaluación general (Software Azteca – Universidad de Córdoba. 2008).

### **7.3 CONCLUSIONES**

En este libro se ha demostrado que el modelo presentado es fácil de implementar por parte de los docentes y practicantes de las instituciones de educación básica y secundaria en Colombia, en el fondo es el objetivo principal del modelo, y es ideal para elaborar proyectos de software educativos en el sistema colombiano de educación porque cumple con los lineamientos que enuncia el programa Evaluación Censal de la Calidad de la Educación creado por el Ministerio de Educación Nacional bajo la coordinación del ICFES, donde califican la calidad de la educación en función del manejo y consecución de competencias; esta es uno de los ejes que fundamenta el modelo MODESEC-ISO/IEC 12207.

La inclusión de estándares internacionales en los procesos o las fases de un modelo para el desarrollo de software educativo fortalecen y garantizan la calidad del desarrollo del mismo, debido a que estos brindan las pautas y la documentación necesaria para brindar un producto con eficiencia y eficacia. En el modelo MODESEC-ISO/IEC 12207 todas las fases se ajustaron al estándar internacional ISO/IEC 12207 que es el otro eje que lo fundamenta, buscando la organización y control de las tareas que implica cada una de las fases incluidas en la propuesta.

El modelo propuesto se convierte en una guía de fácil uso para implementar software educativo a la medida de las necesidades de una institución educativa, las cuales cuentan con profesionales en el área del problema, con expertos en pedagogía y didáctica, y con docentes expertos en informática. Con este personal mínimo se puede desarrollar un software educativo de calidad siguiendo el modelo propuesto, sin incurrir en gastos extra para la institución.

Con respecto a los otros modelos analizados, el modelo propuesto cuenta con grandes ventajas como por ejemplo ser el único que presenta una metodología para abordar de forma clara el diseño de las competencias que se van a tratar en el software educativo. Otra ventaja que presenta es que está diseñado para desarrollar software educativo con bajo presupuesto y con un equipo de trabajo que no supera las cinco personas. Estos son factores decisivos en la realidad colombiana, pues las instituciones educativas de las regiones cuentan con presupuestos limitados que no les permiten hacer grandes inversiones en este tipo de materiales.

Aunque el modelo esté planteado para el desarrollo de software educativo a la medida, esto no es impedimento para poder utilizarlo a nivel industrial en la producción en masa, ya que los principios se mantienen igual.

## PUNTOS CLAVES

- Los modelos estudiados no se ajustan al sistema educativo colombiano sino que se ajustan a las especificaciones requeridas en los contextos educativos de cada uno de los países donde se han propuesto y los modelos orientados a objetos son complejos y le van a dificultar a los docentes de las Instituciones Educativas colombianas utilizarlos al momento de emplear soluciones a las necesidades educativas que se puedan presentar.
- Este modelo ha sido implementado en distintos software realizados en trabajo de grado y los orientados por la línea de desarrollo de software educativo, a demás la validación del mismo en la asignatura de Diseño de software educativo que se da en la Universidad de Córdoba.

## REFERENTES

- CARO, Manuel., ÁLVAREZ, L., OME, L. Software Azteca. Edupmedia. Universidad de Córdoba. Montería. 2008.
- GÓMEZ, R., GALVIS, A., MARIÑO, O. Ingeniería de software educativo y modelaje orientado a objetos. Un medio para desarrollar micro mundos interactivos. Universidad de Los Andes. Bogotá. Colombia. 2000.
- MARQUÉS, Pere. Selección y Uso de Recursos Multimedia. Diseño de Actividades. Usos en el Aula Informática. 1999. Disponible en: <http://dewey.uab.es/pmarques/interven.htm>. Consultado el 03 de abril de 2009
- MENDOZA, L., PÉREZ, M., GRIMÁN, A. Prototipo de modelo sistémico de calidad (MOSCA) del software. Universidad Simón Bolívar. Caracas, Venezuela. 2004.
- MOLINA, R. Modelo de desarrollo de software educativo. Argentina. 2006.

## **REFERENCIAS**

- ALFONSO S, Illeana; GONZÁLEZ P, Troadio. Proceso de enseñanza-aprendizaje: Algunas características y particularidades. [En línea]. Disponible en: <http://www.monografias.com/trabajos7/proe/proe.shtml?relacionados>. Acceso 02 marzo de 2009.
- ALONSO PALACIO, Luz Marina & ESCORCIA DE VÁSQUEZ, Isabel. El ser humano como una totalidad. [En línea]. Disponible en: [http://www.escenf.unam.edu.ar/docs/programas/salud/Articulo\\_1SC.pdf](http://www.escenf.unam.edu.ar/docs/programas/salud/Articulo_1SC.pdf). Acceso 22 marzo de 2009.
- AMORÓS, Eduardo. Comportamiento organizacional: En busca del desarrollo de ventajas competitivas.
- ARLOW, Jim. UML and the Unified Process. Editorial Addison Wesley. 2002.
- BENEITONE, Pablo; et al. Reflexiones y perspectivas en la Educación Superior en América Latina. Informe final-Proyecto Tuning- América Latina 2004-2007. [En línea]. Disponible en: <http://tuning.unideusto.org/tunigal/index.php>. Consultado 22 marzo de 2009.
- BLAUBERG, I; KOPNIN, P & PANTIN, I. Diccionario filosófico marxista segunda edición. Armadillo ediciones. 1975
- BOOCH, G. Lenguaje de Modelado Unificado. Editorial Addison Wesley. Madrid. 2001.
- BRUNNER, José Joaquín. Competencias para la vida: Proyecto DeSeCo. Publicado en Diciembre 11, 2005. [En línea]. Disponible en: [http://mt.educarchile.cl/mt/jjbrunner/archives/2005/12/\\_deseco\\_es\\_el\\_n.html](http://mt.educarchile.cl/mt/jjbrunner/archives/2005/12/_deseco_es_el_n.html). Consultado 22 marzo de 2009.
- CARO, Manuel., ÁLVAREZ, L., OME, L. Software Azteca. Edupmedia. Universidad de Córdoba. Montería. 2008.
- DE ZUBIRÍA, J. De la escuela nueva al constructivismo. Un análisis crítico. Cooperativa Editorial. Magisterio. Bogotá. 2001.
- DÍAZ, Fernando. Módulo pedagógico Invitación a la investigación científica. Editorial Louis Argel asociados. Montería. 1999.
- Especificación del proyecto multimedia: Guión multimedia. Disponible en: [www.dsic.upv.es/~fgarcia/imm/Traspas/Especificación%20del%20proyecto%20multimedia.ppt](http://www.dsic.upv.es/~fgarcia/imm/Traspas/Especificación%20del%20proyecto%20multimedia.ppt) -. Consultado el 22 de marzo de 2009.
- ERIKSSON, Hans y otros. UML2 Toolkit. Editorial Wiley Publishing. 2004.

- FLÓREZ, Rafael. *Pedagogía del Conocimiento*, segunda edición. Editorial Nomos. Bogotá D.C. 2005
- FLOWER, Martin. *UML gota a gota*. Editorial Addison Wesley. 1999.
- GALVIS PANQUEVA, Álvaro H. *Ingeniería de Software Educativo*. Ed. Uniandes. . Bogotá D.C. 1992.
- GÓMEZ, R., GALVIS, A., MARIÑO, O. *Ingeniería de software educativo y modelaje orientado a objetos. Un medio para desarrollar micro mundos interactivos*. Universidad de Los Andes. Bogotá. Colombia. 2000.
- HOYOS, Guillermo. MARTÍNEZ, Miquel & *et al.* ¿Qué significa educar en valores hoy?. Edición Organización de Estados Iberoamericanos para la Educación, la Ciencia y la Cultura (OEI) – Octaedro. 2004. [En línea]. Disponible en: <http://www.oei.es/publicaciones/educaval1.htm>. Acceso 22 marzo de 2009.
- ISO 12207. Disponible en: <http://www.abelia.com/docs/12207cpt.pdf>. Consultado el 23 de febrero de 2009
- KAUFMAN, R.A. *Needs Assesment: What It Is How to Do It*. University Consortium on Instructional Development and Technology: San Diego, Cal, 1976.
- LOZADA A., Moreno H. *Competencias básicas aplicadas al aula*. Ediciones SEM (Servicio educativo del magisterio). Bogotá D.C. 2003.
- MARQUÈS, Pere. *El software educativo*. Universidad Autónoma de Barcelona
- MARQUÉS, Pere. *Selección y Uso de Recursos Multimedia. Diseño de Actividades. Usos en el Aula Informática*. 1999. Disponible en: <http://dewey.uab.es/pmarques/interven.htm>. Consultado el 03 de abril de 2009
- MENDOZA, L., PÉREZ, M., GRIMÁN, A. *Prototipo de modelo sistémico de calidad (MOSCA) del software*. Universidad Simón Bolívar. Caracas, Venezuela. 2004.
- MOLINA, R. *Modelo de desarrollo de software educativo*. Argentina. 2006.
- MOLINA VASQUEZ, Ruth. *Algunos aportes sobre el software educativo*. Disponible en: [http://www.karisma.org.co/documentos/softwareredp/APORTES\\_AL\\_CONCEPTO\\_DE\\_SOFTWARE\\_EDUCATIVO.doc](http://www.karisma.org.co/documentos/softwareredp/APORTES_AL_CONCEPTO_DE_SOFTWARE_EDUCATIVO.doc). Acceso 02 mayo de 2009.

- ORTIZ OCAÑA, Alexander Luís. Modelos pedagógicos: Hacia una escuela del desarrollo integral.
- PEREZ, Felipe. Guión didáctico. Disponible en: <http://laxantesaboramora.obolog.com/guion-didactico-108808>. Consultado el 03 de marzo de 2009.
- PRESSMAN, Roger. Ingeniería del Software. Editorial Mc Graw Hill. 2002. Ed. 5
- PRESSMAN, Roger. Ingeniería del Software. Editorial Mc Graw Hill. 2005. Ed. 6
- Proyecto alfa tuning – América Latina: carreras basadas en competencias. [En línea]. Disponible en: [http://www.udc.es/grupos/cndeuto/docs/Sintesis\\_del\\_proyecto.pdf](http://www.udc.es/grupos/cndeuto/docs/Sintesis_del_proyecto.pdf). Consultado 22 marzo de 2009.
- SCHMULER, Joseph. Aprendiendo UML en 24 horas. Editorial Prentice Hall. 2001.
- SOMMERVILLE, Ian. Ingeniería del Software. Editorial Addison Wesley. 2005. Ed. 7
- Valores humanos, ¿qué son?. [En línea]. Disponible en: [http://www.masalto.com/oque/template\\_oquedestacado.phtml?consecutivo=5537&subsecc=3&cat=11&subcat=31&subj=47](http://www.masalto.com/oque/template_oquedestacado.phtml?consecutivo=5537&subsecc=3&cat=11&subcat=31&subj=47) Acceso 22 marzo de 2009.

## WEBGRAFÍA

- [http://es.wikipedia.org/wiki/Mapa\\_conceptual](http://es.wikipedia.org/wiki/Mapa_conceptual).
- [http://es.wikipedia.org/wiki/Mapa\\_mental](http://es.wikipedia.org/wiki/Mapa_mental).
- <http://nodocreativo.blogspot.com/2008/04/cmo-realizar-un-guin-multimedia-parte-i.html>.
- <http://www.psicopedagogia.com/definicion/mentefacto>.
- <http://www.monografias.com/trabajos908/guion-didactico-multimedia/guion-didactico-multimedia2.shtml>
- <http://www.proyectosfindecarrera.com/definicion/manualdelusuario.htm>
- Multimedia 2. <http://www.slideshare.net/raymarq/multimedia-2-presentation>