

<b>INTRODUCCIÓN</b>	<b>1</b>
<b>CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA</b>	<b>5</b>
1.1. Antecedentes	5
1.2. El software educativo	6
1.2.1. Características esenciales de un <i>software</i> educativo:	7
1.2.2. Clasificación del <i>software</i> educativo según su estructura	7
1.2.3. Principales funciones del <i>software</i> educativo	8
1.2.4. Ventajas y desventajas de los <i>software</i> educativos	10
1.2.5. Características fundamentales de la multimedia educativa como <i>software</i> educativo	11
1.3. La aplicación multimedia: elementos básicos sobre su desarrollo	12
1.3.1. Principios que deben de cumplir las aplicaciones multimedia	14
1.3.2. Categorías y funciones que debe cumplir una multimedia educativa	17
1.3.3. Criterios de calidad de una aplicación multimedia	18
1.4. Herramientas a utilizar en el desarrollo de aplicaciones multimedia	19
1.5. Ventajas de <i>Neobook</i> para la construcción de multimedia	20
1.6. Gestor de base de datos: Access	21
1.7. Rational unified process (RUP)	21
Conclusiones del Capítulo	22
<b>CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA</b>	<b>23</b>
2.1. Modelo del dominio	23
2.2. Requerimientos del sistema	24
2.2.1. Requerimientos funcionales	24
2.2.2. Requerimientos no funcionales	25
2.3. Valoración de sostenibilidad del sistema propuesto	27
2.3.1. Dimensión Administrativa	27
2.3.2. Dimensión Socio-Humanista	33
2.3.3. Dimensión Ambiental	34
2.3.4. Dimensión Tecnológica	35
2.3.5. ¿Es el sistema propuesto sostenible?	35
2.4. Modelo de Casos de Uso del Sistema	36
2.4.1. Actores del sistema	36
2.4.2. Diagramas de Casos de Uso del Sistema	36

<b>2.5. Análisis</b>	<b>37</b>
<b>2.6. Modelo del Diseño</b>	<b>39</b>
<b>2.6.1. Mapa de Navegación</b>	<b>40</b>
<b>2.6.2. Diseño de la base de datos</b>	<b>40</b>
<b>2.7. Implementación de la multimedia</b>	<b>47</b>
<b>2.7.1. Herramientas utilizadas en el desarrollo del software</b>	<b>47</b>
<b>2.7.2. Peculiaridades de la programación con el uso de los script</b>	<b>47</b>
<b>2.7.3. Utilización de eventos</b>	<b>49</b>
<b>2.7.4. Utilización de recursos audiovisuales y trabajo con animaciones</b>	<b>51</b>
<b>2.7.5. Utilización de recursos matemáticos y trabajo con la base de datos</b>	<b>53</b>
<b>2.8. Prueba</b>	<b>54</b>
<b>2.8.1. Casos de Prueba</b>	<b>55</b>
<b>2.8.2. Procedimientos de Prueba</b>	<b>56</b>
<b>Conclusiones del Capítulo</b>	<b>58</b>
<b>CONCLUSIONES</b>	<b>58</b>
<b>RECOMENDACIONES</b>	<b>59</b>
<b>BIBLIOGRAFÍA</b>	<b>60</b>
<b>REFERENCIAS BIBLIOGRÁFICAS</b>	<b>64</b>
<b>ANEXOS</b>	<b>67</b>

## INTRODUCCIÓN

En las últimas décadas los graduados de la carrera Ingeniería Informática han demostrado que resulta imprescindible el uso de las mejores técnicas de Ingeniería y Gestión de Software; por lo que lograr su formación es de vital importancia. Esta preparación se concentra en los tres primeros años de la carrera en las asignaturas de la Disciplina Principal Integradora, ya que el futuro egresado no puede carecer de las mejores prácticas de la Ingeniería de Software, de ahí que resulte indispensable integrar los conocimientos y habilidades de las asignaturas de estas dos áreas.

La asignatura Fundamentos de la Informática es de gran importancia para los estudiantes, pues tiene entre sus objetivos educativos desarrollar hábitos como: estudio sistemático, consultar la bibliografía especializada, utilización de sistemas informáticos para el almacenamiento y recuperación de información, permitiendo el crecimiento de su personalidad y el mejoramiento individual en su inserción social.

Entre los problemas que impiden el cumplimiento de estos objetivos se encuentran:

- El Plan de estudio D no corresponde con los temas a tratar en la asignatura Fundamentos de la Informática.
- La información se encuentra dispersa.
- No se tiene un texto básico unificado.
- Existen limitaciones de acceso a las fuentes de información.
- Los seminarios son la tipología de clases más frecuentes por lo que se requiere de una base bibliográfica amplia.

Por lo que es necesario aglutinar toda esta gama de información en un soporte digital atractivo que permita la perdurabilidad de la información en el tiempo y la interacción de la misma.

Teniendo en cuenta las situaciones anteriores, se define como **Problema Científico de la Investigación:**

¿Cómo organizar y publicar los materiales didácticos y la bibliografía digital de la asignatura Fundamentos de la Informática, de forma tal que puedan ser consultados, favoreciendo la interacción, el trabajo colaborativo y la evaluación de las habilidades definidas?

El **objeto de la investigación se centra en:** El proceso de enseñanza-aprendizaje de la asignatura Fundamentos de la Informática, en la carrera Ingeniería Informática de la Universidad de Granma.

Enmarcado en el siguiente **campo de acción:** Sistema informático para la enseñanza de la asignatura Fundamentos de la Informática, en la carrera Ingeniería Informática de la Universidad de Granma.

**La idea a defender** que orientará a desarrollar esta investigación es que: Con el desarrollo de un *software* educativo para la asignatura Fundamentos de la Informática en la Universidad de Granma, contribuirá a elevar el rendimiento académico y la motivación de los estudiantes.

Para contribuir a transformar el objeto de estudio y solucionar el problema planteado se define como **objetivo general:**

Desarrollar un *software* educativo que permita el almacenamiento de recursos bibliográficos y documentales relevantes sobre la asignatura Fundamentos de la Informática, la cual fomentará los hábitos de estudio.

### **Objetivos específicos**

1. Analizar la situación actual del Proceso Docente Educativo de la asignatura Fundamentos de la Informática en la carrera Ingeniería Informática de la Universidad de Granma.
2. Implementar un *software* educativo para la asignatura Fundamentos de la Informática, en la carrera Ingeniería Informática de la Universidad de Granma.

## **Tareas de la investigación:**

1. Identificar los principales requerimientos y necesidades para la gestión de la información.
2. Seleccionar las herramientas adecuadas para la implementación del software educativo a partir del análisis realizado.
3. Realizar el diseño del sistema.
4. Implementar el software educativo siguiendo el diseño realizado.

## **Métodos Científicos de Investigación**

Para dar cumplimiento a las tareas anteriores se emplearon los siguientes métodos y técnicas:

### **Método teórico**

**Análisis y síntesis:** Para el desarrollo del sistema se realizó una investigación previa de los procesos que intervienen en el desarrollo de *software* y los principales elementos que integran las metodologías para su construcción.

**Histórico – lógico:** Se utilizó para el estudio del desarrollo histórico del proceso de control de la asistencia personal y de los antecedentes de investigaciones relacionadas con la gestión de información.

### **Método empírico:**

#### **Observación**

Se utilizó en la recogida de información y es de gran utilidad en las etapas de análisis y diseño, donde se realiza el diagnóstico del problema a investigar.

La posibilidad de diseñar y elaborar un producto multimedia con fines educativos, que aglutine toda la información necesaria de la asignatura Fundamentos de la Informática, y estando al alcance de los estudiantes contribuya a elevar su nivel de conocimiento, su formación integral de la profesión, a través de la interacción y evaluación de los conocimientos, constituye el principal **aporte e importancia de esta investigación.**

El presente trabajo está estructurado en introducción, dos capítulos, bibliografía, referencias bibliográficas, conclusiones, recomendaciones y anexos.

El capítulo 1 describe cuál es la situación existente con los materiales referentes a la asignatura Fundamentos de la Informática en la carrera de Ingeniería Informática de la Universidad de Granma y aborda los principales fundamentos teóricos necesarios del universo de la tecnología multimedia sobre los que se sustenta el desarrollo de esta investigación, constituyendo el marco teórico y referencial de la misma.

El capítulo 2 describe el desarrollo del sistema en la fase de análisis, diseño e implementación del mismo. En los que se expone la captura de requisitos, el modelado del sistema, la confección del guión multimedia y en lo concerniente a la implementación, se aborda el proceso de programación del software, los principios paradigmáticos de las aplicaciones multimedia y se detallan sus principales funcionalidades.

Finalmente se concluyen los resultados del trabajo, se realizan las recomendaciones que se consideran pertinentes y se expone la bibliografía consultada.

# CAPÍTULO 1.FUNDAMENTACIÓN TEÓRICA

## Introducción

La inserción y proliferación del desarrollo alcanzado por las Tecnologías de la Información y las Comunicaciones (TIC) en todas las esferas de la sociedad ha provocado cambios sustanciales en la educación, llegando a modificar incluso los modelos tradicionales de enseñanza. Los profundos cambios tecnológicos que experimenta actualmente la sociedad amplían las opciones metodológicas que influyen sobre la práctica educativa a todos los niveles de la enseñanza.

En este capítulo se realiza un estudio bibliográfico actualizado sobre las principales tendencias, tecnologías, herramientas, conceptos y relaciones que apoyan el desarrollo de este trabajo, constituyendo el marco teórico-referencial.

### 1.1. Antecedentes

El uso de la computadora ha tenido gran impacto en el proceso de la enseñanza, llegando a muchos lugares del mundo por su valor e importancia en la obtención de conocimientos. Conseguir en los estudiantes que el autoaprendizaje se convierta en una actividad consciente, permanente y sistemática, compone el elemento fundamental del triunfo de estos métodos didácticos.

Actualmente existen varios *software* que pueden visualizar códigos, datos y algoritmos como son el PV Prototype, MacGnome, y TANGO mas no existe un programa que contenga toda esta información en un soporte digital atractivo, agradable y de fácil uso. Es elemental tener en cuenta que el solo hecho del uso adecuado de estas tecnologías por sí solas, no avala la superación en la enseñanza; haciendo necesario establecer la relación tecnología-enseñanza. Dar calidad al proceso de enseñanza de *software* educativos, favorece tanto a los educadores como a los educandos.

## 1.2. El software educativo

En la década de los años 60 aparecen en Estados Unidos (EEUU) las primeras ideas sobre el desarrollo de *software* educativo, tomando mayor auge después de la aparición de las microcomputadoras a fines de los años 80. Los primeros pasos de este desarrollo fueron dados por el Instituto Tecnológico Massachusetts(MIT), empleándolos como material didáctico en el lenguaje LOGO en un sentido constructivista del aprendizaje, extendiéndose posteriormente a numerosas escuelas y universidades de EEUU.

Los estudios realizados acerca del educativo han permitido obtener diversas definiciones. Sánchez, en su libro: “Construyendo y Aprendiendo con el Computador” (Sánchez, 1999), define el concepto de *software* educativo como: “cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar”. Un concepto más restringido de *software* educativo es: “aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender”. (Marqués,1995).

Atendiendo a las definiciones antes mencionadas un **software educativo es todo aquel programa para computadora que se desarrolla con la finalidad específica de ser utilizado como recurso didáctico en el proceso de enseñanza aprendizaje.**

Un software educativo bien elaborado y utilizado contribuye en gran medida a alcanzar los objetivos educativos propuestos. Para su elaboración se debe tener en cuenta:

- Las características de la interfaz de comunicación, deberá estar diseñada de acuerdo con la teoría comunicacional aplicada y con las diferentes estrategias para el desarrollo de determinadas habilidades mentales.
- El contenido del aprendizaje debe ordenarse de tal manera que los conceptos más generales e inclusivos se presenten al principio, favoreciendo la formación

de conceptos incluyentes en la estructura cognoscitiva de los alumnos facilitando el aprendizaje significativo de los otros elementos del contenido.

### **1.2.1. Características esenciales de un *software* educativo:**

- Son materiales elaborados con una finalidad didáctica.
- Utilizan el ordenador como soporte en el que los alumnos realizan las actividades que ellos proponen.
- Permiten un diálogo y un intercambio de información entre el ordenador y los estudiantes.
- Individualizan el trabajo de los estudiantes, adaptándolos al ritmo de trabajo de cada uno, así como sus actividades según las actuaciones de los alumnos.
- Son fáciles de utilizar.

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

### **1.2.2. Clasificación del *software* educativo según su estructura**

Según (Marqués,1995) atendiendo a su estructura el *software* educativo se clasifica en:

- Programas tutoriales.
- Programas de ejercitación.
- Simuladores.
- Base de datos.
- Constructores.
- Programas herramientas.

**Programas tutoriales:** presentan contenidos y proponen ejercicios autocorrectivos. Pretenden que a partir de una información los estudiantes pongan en práctica determinadas capacidades: que aprendan, refuercen conocimientos y habilidades. Si utilizan técnicas de Inteligencia Artificial se denominan tutoriales expertos.

**Programas de ejercitación:** responden a la necesidad de realizar actividades específicas sencillas. Se diseñan con diferentes niveles de complejidad, en dependencia de su aplicación (reproductiva o productiva) y su principal objetivo es que los estudiantes adquieran habilidades que los conduzcan a la reafirmación y consolidación de los conocimientos.

**Simuladores:** presentan un modelo dinámico, que facilitan su exploración, los estudiantes pueden realizar aprendizajes inductivos o deductivos mediante la observación y manipulación de la estructura del *software*. Así pueden descubrir sus elementos y adquirir experiencia directa frente a situaciones de la vida real poco frecuentes, peligrosas o costosas.

**Base de datos:** presentan datos organizados en un entorno estático, mediante unos criterios que facilitan su exploración y consulta selectiva para resolver problemas, analizar, relacionar datos, comprobar hipótesis y extraer soluciones.

**Constructores o talleres creativos:** facilitan aprendizajes heurísticos y presentan entornos programables que les permite a los estudiantes convertirse en profesores del ordenador.

**Programas herramientas:** presentan un entorno instrumental que facilita la realización de trabajos de tratamiento de la información, tales como escribir, organizar, calcular, etc.

### 1.2.3. Principales funciones del *software* educativo

- El empleo del *software* educativo será eficiente en dependencia del uso que se le proporcione, y para ello es fundamental conocer las funciones que ha de cumplir. Dependiendo del tipo de *software*, pueden realizar funciones básicas propias de los materiales educativos, en algunos casos pueden cumplir funciones específicas. Según (Marqués,1995) un *software* educativo puede cumplir con las siguientes funciones:

- **Instructiva:** Es una de las principales funciones que posee, debido a las características y propósitos que persigue todo *software*. En el proceso de aprendizaje la computadora actúa como mediador en la construcción del conocimiento, promoviendo actividades interactivas a través del *software*.
- **Informativa:** El *software* educativo como cualquier material educativo, es un elemento que proporciona información a los estudiantes, además de servir como un material complementario para la labor pedagógica de los docentes.
- **Motivadora:** La introducción del computador en el proceso de instrucción, por sí sola, es un elemento de motivación intrínseca, que hace que los estudiantes se sientan atraídos e interesados. Además, los programas llevan inmersos sistemas de motivación con el propósito de captar la atención y aumentar el interés en el aprendizaje. La función motivadora, por ello, se constituye en una de las principales características del *software* educativo.
- **Evaluadora:** Los *softwares* educativos por lo general poseen sistemas de registros de usuarios, con el propósito de registrar las acciones y los logros de los estudiantes. Además la retroinformación de los logros se produce en el acto, propiciando en el caso de los errores nuevas secuencias de aprendizaje. La evaluación puede ser de dos tipos:
 

*Implícita*, cuando el estudiante detecta sus errores y se evalúa a partir de comparar sus respuestas con las que le presenta la computadora. (el sistema puede emplear sonidos para indicar errores o generar información de retorno).

*Explícita*, cuando el *software* presenta los informes globales del logro de las metas establecidas.
- **Investigadora:** Los programas, especialmente las bases de datos, simuladores y programas constructores, ofrecen a los estudiantes interesantes entornos donde investigar: buscar determinadas informaciones y cambiar los valores de las variables de un sistema; además con los programas herramientas, pueden proporcionar a los profesores y estudiantes instrumentos de gran utilidad para el desarrollo de trabajos de investigación

que se realicen con dificultades cuando se hacen al margen de los computadores.

- **Expresiva:** Dado que las computadoras son máquinas capaces de procesar los símbolos mediante los cuales las personas representan sus conocimientos y se comunican, debido a que sus posibilidades como instrumento expresivo son muy amplias.
- **Desde el aspecto informático:** A través del *software* educativo los estudiantes se expresan y se comunican con la computadora y con otros compañeros a través de las actividades de los programas y especialmente, cuando utilizan lenguajes de programación, procesadores de textos, editores de gráficos, etc.
- **Lúdica:** Trabajar con el computador realizando actividades educativas es una labor que a menudo tiene connotaciones lúdicas y festivas para los estudiantes. Además, algunos programas refuerzan su atractivo mediante la inclusión de determinados juegos o entretenimientos, con lo que potencian aún más esta función.
- **Innovadora:** Aunque no siempre sus planteamientos pedagógicos resulten innovadores, los programas educativos se pueden considerar material didáctico con esta función, ya que utilizan una tecnología recientemente incorporada a los centros educativos y en general, suelen permitir muy diversas formas de uso.

#### 1.2.4. Ventajas y desventajas de los *software* educativos

##### Ventajas del *software* educativo:

- Enriquece el campo de la Pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza - aprendizaje.
- Pueden adaptar el *software* a las características y necesidades de su grupo teniendo en cuenta el diagnóstico en el proceso de enseñanza - aprendizaje.
- Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.

- Permiten elevar la calidad del proceso docente - educativo.
- Permiten controlar las tareas docentes de forma individual o colectiva.
- Muestran la interdisciplinariedad de las asignaturas.
- Marca las posibilidades para una nueva clase más desarrolladora.

#### **Desventajas del software educativo:**

- Adicción: El *software* educativo interactivo resulta motivador, pero un exceso de motivación puede provocar adicción.
- Distracción: Los alumnos a veces se dedican a jugar en vez de trabajar.
- Ansiedad: La continua interacción ante el ordenador puede provocar ansiedad en los estudiantes.
- Aprendizajes incompletos y superficiales: La libre interacción de los alumnos con estos materiales (no siempre de calidad) a menudo proporciona aprendizajes incompletos con visiones de la realidad simplista y poco profunda.
- Cansancio visual y otros problemas físicos: Un exceso de tiempo trabajando ante el ordenador o malas posturas pueden a la larga provocar diversas dolencias.

#### **1.2.5. Características fundamentales de la multimedia educativa como *software* educativo**

Las multimedias educativas entran dentro de la categoría del *software* educativo, y por tanto cumplen con sus características y propiedades. A continuación se analizan aquellas que se consideran necesarias para sustentar la construcción del producto de esta investigación.

Las multimedia educativas pueden presentar diferentes materias de formas muy heterogéneas (a través de información estructurada a los estudiantes y simulando fenómenos o procesos), brindan un entorno de trabajo asequible, ameno a los

estudiantes y con posibilidades de interacción; sin embargo, comparten características esenciales comunes (Acosta, 2009), tales como:

- Son materiales elaborados con una finalidad educativa.
- Son interactivos, contestando inmediatamente las acciones de los estudiantes.
- Individualizan el trabajo de los estudiantes, pues se acomodan al ritmo de trabajo de cada uno y pueden ajustar sus actividades según las actuaciones de los alumnos.
- Constituyen agentes de motivación para que el estudiante pueda interesarse en el contenido objeto de estudio.
- Son fáciles de usar, pues los conocimientos informáticos necesarios para utilizar la mayoría de estos programas son mínimos.

### **1.3. La aplicación multimedia: elementos básicos sobre su desarrollo**

Según el Diccionario Larousse el significado de la palabra **multimedia** es: Se aplica al elemento o equipo informático que reúne diversos medios como audio, video o televisión interaccionados. Diversos autores dan conceptos parecidos de multimedia como que: se asocia a cualquier objeto computacional que usa de forma integradora diferentes formas de contenido informativo como textos, sonidos, imágenes, animaciones y videos para informar o entretener al usuario (Bou, 2003).

Por lo tanto, se define multimedia: el *software* que utiliza diferentes medios de comunicación como imágenes, animaciones, textos, videos, sonidos para informar, enseñar y entretener al usuario mediante un ambiente agradable y sencillo.

En el trabajo “Nuevas tecnologías de la información y de la comunicación aplicadas a la educación”, José Manuel Ríos y Manuel Cebrián, plantean que la **importancia actual de las multimedias** están dadas por:

- Presentan las ventajas comunes a todas las tecnologías, permitiendo además una mayor interacción.

- Ofrecen la posibilidad de controlar el flujo de la información.
- Gracias a la información almacenada en un disco óptico, ofrecen gran rapidez de acceso y durabilidad.
- Unen todas las posibilidades de la Informática y de los medios audiovisuales.
- La información audiovisual que contiene un disco óptico puede ser utilizada para varias finalidades.
- Un programa multimedia bien diseñado no corre el peligro de obsolescencia, puesto que pueden actualizarse constantemente los contenidos con pequeños cambios en el *software*.

“De acuerdo a un grupo de investigadores, el ser humano tiene la capacidad de retener:

- Un 20% de lo que escucha.
- Un 40% de lo que ve y escucha.
- Un 75% de lo que ve, escucha y practica.”

Actualmente el estudiante universitario debido a su inquietud y curiosidad, si se les proporcionan múltiples materiales atractivos, preparados con fines instructivos y se les dan los acomodos necesarios para que se muevan por un mundo de información que no les resulte tedioso explorar, se puede esperar que esto de en ellos por resultado el mantenimiento del dinamismo, consiguiendo un alto nivel de retención de la información que desemboca en la formación del conocimiento que se espera.

Si estos materiales son programas de computadora, documentos o presentaciones que combinan adecuadamente los medios, se mejora notablemente la atención, la comprensión y el aprendizaje, acercándose a la manera habitual en que los seres humanos se comunican cuando emplean varios sentidos para comprender un mismo objeto, proceso, fenómeno o concepto. Así funcionan los llamados sistemas multimedia, aplicaciones multimedia o simplemente multimedia. (Acosta, 2009).

El desarrollo actual de la tecnología multimedia es el resultado lógico del progreso concurrente de la tecnología informática (Rojas, José, 2008) que se resume en:

- **Acelerado desarrollo del hardware:** caracterizado en lo fundamental por el incremento de las capacidades de almacenamiento de los soportes de información, la fabricación de microprocesadores mucho más rápidos y el perfeccionamiento de los dispositivos destinados a la captura, la digitalización y la compresión de la información, así como de los diferentes dispositivos periféricos (memorias RAM, discos duros, memorias flash, CD, DVD).
- **Multiplicidad y diversificación utilitaria en el desarrollo del software:** surgimiento de nuevas interfaces gráficas y el desarrollo de sistemas de autor cada vez más interactivos.

### 1.3.1. Principios que deben de cumplir las aplicaciones multimedia

En el diseño de una aplicación multimedia es indispensable tomar en consideración un conjunto de requisitos, que conjugados, aseguran la efectividad de la aplicación multimedia en el proceso de aprendizaje y estos requisitos son los nombrados: “Principios que deben de cumplir las aplicaciones multimedia”: (Chaveco, J. R. y Rojas, J. R., 2004)

1. **Principio de necesidad:** A excepción de notables casos particulares, todas las aplicaciones deben regirse por el principio de necesidad: deben ser necesarias. Esto quiere decir que, para su diseño, se debe partir a priori de:
  - La aplicación sirve para algo (necesidad de la existencia de la aplicación).
  - La aplicación debe ser multimedia (necesidad de ser diseñada bajo este enfoque).
2. **Principio de vitalidad:** Ante todo hay que dejar bien claro que las aplicaciones multimedia deben ser dinámicas y es uno de los indicadores principales para distinguir entre trabajos bien hechos y mal acabados o poco profesionales. Debemos conseguir que la pantalla “esté viva”, utilizando por ejemplo mascotas o iconos en movimiento en los momentos en que el usuario

piensa, así como la utilización de técnicas de animación a la hora de presentar un contenido.

**3. Principio de retroalimentación:** La retroalimentación, como su nombre lo indica, consiste en que una aplicación utiliza la información que obtiene del usuario para la corrección y mejora de su propio funcionamiento. Según (Bou, 2003), para el caso de las aplicaciones multimedia hay que tener en cuenta las siguientes cuatro interrogantes:

- ¿Qué información se recoge?
- ¿Cómo se presenta?
- ¿A quién se dirige?
- ¿Cómo se procesó?

En una aplicación multimedia, al analizar las respuestas dadas por el usuario se obtiene información importante para el propio guionista. Esta información debe utilizarse para revertirla en el progreso del propio usuario. Por ejemplo, en una aplicación educativa se debe disponer de un mecanismo que les informe a los usuarios de sus errores, indique cómo corregirlos y oriente sobre los progresos conseguidos desde que empezaron a estudiar.

**4. Principio de la múltiple entrada:** A diferencia de los *software* de amplio espectro a los que se está acostumbrado (procesador de textos, procesador de datos, etc.), que se diseñan pensando en la coordinación pero no en el perfil (conocimientos, expectativas, sentimientos) del usuario, las aplicaciones multimedia normalmente son diseños que se conciben para ser utilizados por un tipo determinado de usuario. Es de vital importancia tener en cuenta estos aspectos que radican en establecer los nexos de comunicación de manera tal que sean empleados todos los canales coherentemente sincronizados, estableciendo consistentes enlaces entre los medios.

- 5. Principio de interactividad:** La interactividad es un recurso propio de los sistemas informáticos especialmente importante (de entrada, constituye la ventaja principal de las aplicaciones actuales sobre los productos de video tradicional). La aplicación multimedia debe ser concebida como refuerzo potencial del mensaje, debe contener en su diseño posibilidades de participación activa o grupal y evitar las zonas en pantalla que recuerdan que no se puede interactuar. El diseño de la interacción debe asumirse como una tarea diferenciada (aunque no separada o independiente del resto) dentro de una aplicación multimedia.
- 6. Principio de la seguridad:** La seguridad en la gestión de la información se basa en la accesibilidad, si una información es de difícil acceso, es como si se hubiera perdido.
- 7. Principio de libertad:** La información presentada deberá generar en el estudiante una aparente libertad y descubrimiento de la información. El término aparente indica que esta "libertad" es realmente el resultado de la apariencia del diseño, de manera que el proceso de aprendizaje se precise la obtención de conocimientos progresivamente.
- 8. Principio de la usabilidad:** El ambiente debe ser flexible con las necesidades físicas y psíquicas del usuario. El hecho de que el *software* sea una aplicación multimedia no garantiza el *Ipsa Facto*, porque deben tenerse en cuenta aspectos tales como: que en ocasiones resulta más cómodo oprimir una tecla que hacer clic en un pequeño botón, sobre todo si dicha operación es necesario realizarla reiteradamente o que la presencia de sonidos y música puede perturbar la concentración de determinados tipos de usuarios. La instalación del programa en el ordenador debe ser sencilla, rápida y transparente.
- 9. Principio de atención:** Para que el aprendizaje significativo se realice es necesario que el contenido sea potencialmente necesario, importante para el estudiante y que éste tenga la voluntad de aprender significativamente,

relacionando los nuevos contenidos con el conocimiento almacenado en sus esquemas mentales. Los buenos programas tienen en cuenta las características iniciales de los estudiantes a los que van dirigidos (desarrollo cognitivo, capacidades, intereses, necesidades, etc.) y los progresos que vayan realizando. Cada sujeto construye sus conocimientos sobre los esquemas cognitivos que posee y utiliza determinadas técnicas. Los buenos programas multimedia utilizan potentes recursos didácticos para facilitar los aprendizajes de sus usuarios.

### 1.3.2. Categorías y funciones que debe cumplir una multimedia educativa

Diferentes autores consultados (De la Cruz, 2007), (Marqués, 1995), (Rojas, José, 2008) y (Valdés, 2002) concuerdan en que el *software* educativo debe satisfacer tres condiciones (las que se tornan en categorías): funcionalidad, usabilidad y fiabilidad. En consecuencia, la multimedia educativa también las cumple.

- **La funcionalidad:** es la capacidad del producto para proveer funciones que cumplan con las necesidades específicas o implícitas cuando es utilizado bajo ciertas circunstancias.
- **La usabilidad:** se refiere a la capacidad para ser atractivo, asequible, comprensible, aprendido y empleado por el usuario bajo condiciones específicas. Es decir, debe tener en cuenta la posibilidad de que el usuario novel entienda cómo utilizar inicialmente el sistema y cómo llegar a un máximo nivel de conocimiento y uso del mismo (facilidad de aprendizaje), la diversidad de vías en las que se establece la comunicación (sistema – usuario), la propiedad de la interacción usuario – sistema permite lograr los objetivos previstos y su asesoramiento (solidez y flexibilidad), los recursos de ayuda y forma en que el usuario puede utilizarlos (mecanismos de soporte), que el programa debe despertar y mantener la curiosidad y el interés de los estudiantes hacia los contenidos que se presentan (capacidad de motivación).

- **La fiabilidad:** es la capacidad del *software* para mantener un nivel expreso de rendimiento cuando es utilizado bajo condiciones especificadas. El empleo del *software* educativo será eficiente en dependencia del uso que se le suministre, y para ello es fundamental conocer las funciones que ha de cumplir. Dependiendo del tipo de *software*, pueden realizar funciones básicas propias de los materiales educativos, en algunos casos pueden cumplir funciones específicas.

### 1.3.3. Criterios de calidad de una aplicación multimedia

Los productos formativos multimedia de calidad, son ante todo materiales interactivos y flexibles en su forma y momento de utilización, que pueden proporcionar una formación personalizada y un seguimiento continuo de las actividades y los progresos que realizan sus usuarios, optimizando el proceso de aprendizaje, reduciendo costes económicos y temporales, así como obviando algunos problemas de la enseñanza convencional derivados del espacio y del tiempo (Bou, 2003).

En este marco un producto formativo de calidad deberá ofrecer ante todo una alta funcionalidad a sus usuarios y por lo tanto deberá resultar eficaz, eficiente y relevante para ellos, proporcionando:

**Eficacia didáctica:** La razón de ser de los materiales formativos consiste en facilitar el aprendizaje a sus usuarios; su eficacia se medirá en función de su contribución al logro de estos objetivos. Por otra parte se espera que proporcionen aprendizajes significativos, completos y transferibles a diversas situaciones prácticas, contribuyendo además a desarrollar la capacidad de autoaprendizaje de los usuarios.

**Eficiencia económica:** Los productos multimedia educativos también deben resultar eficientes, reduciendo los costos de formación, el tiempo invertido y el esfuerzo desarrollado por sus usuarios, estudiantes, profesores y gestores de los sistemas de formación.

**Relevancia para los usuarios:** La relevancia se refiere al interés de los destinatarios por el producto, en virtud de que les resuelva un problema importante para ellos o les resulte útil para sus actividades.

#### **1.4. Herramientas a utilizar en el desarrollo de aplicaciones multimedia**

Para desarrollar aplicaciones multimedia se requiere disponer de tiempo y recursos suficientes, por lo que entraña una responsabilidad, perseverancia y constancia. En función de optimizar tanto tiempo como recursos, se debe escoger el mejor sistema que sirva como herramienta de programación adecuado y que se ajuste a los requerimientos de la aplicación que se va a desarrollar.

En la actualidad se cuenta con una gran variedad de sistemas de autor, desde los tradicionales lenguajes de programación hasta las herramientas de diseño especial para el manejo de menús.

Lenguajes de programación como Pascal, C, C++, *VisualBasic*, *Delphi*, exigen grandes habilidades del programador para escribir y desarrollar una buena aplicación, además de que, por sus características propias de desarrollarse a través de códigos, la construcción de la aplicación se hace “lenta”, por lo que no son idóneos para elaborar estos tipos de *software* cuando se precisa de rapidez.

Más factibles son los lenguajes de autor, que también requieren códigos, pero han sido cardinalmente escritos con la intención de provocar “aprendizajes asistidos por computadoras” (CAL: *Computer Aided Learning*, según sus siglas en inglés) de modo que los códigos sean sencillos.

Los sistemas de autor se manipulan, casi siempre, a través de menús y el código se genera automáticamente. Son paquetes de fácil uso, pero a su vez son los menos flexibles (por ejemplo: *AuthorWare*, *Mediator*, *MacromediaDirector*, *ToolBook*, *Neobook*, *Multimedia Builder*, entre otros). En la práctica muchos de los sistemas también tienen *scripting languages* que permiten adicionar facilidades (Acosta, 2009).

## Clasificación de los sistemas de autor

Los sistemas de autor presentan varias clasificaciones, que permiten determinar el uso de los mismos de acuerdo al producto que se quiera elaborar. Estos por su forma de trabajo en la elaboración de aplicaciones, pueden clasificarse según (Acosta, 2009), de la siguiente forma:

- **Sistemas de autor basados en fichas:** en estos sistemas la aplicación consta de unidades llamadas fichas o páginas, como las páginas de un libro. Las fichas se apilan juntas para formar un libro o pila y se puede navegar por página, buscando un tema específico o a través de hiperenlaces. *Neobook* es el representante más conocido de los Lenguajes Autor agrupados en este paradigma.
- **Sistemas de autor basados en íconos:** en un sistema basado en íconos, estos son arrastrado desde una paleta para formar un diagrama de flujo. *IconAuthor* y *Authorware* de Macromedia son sistemas basados en íconos. Una vez que el diagrama de flujo ha sido diseñado, el mismo se puede utilizar para producir una aplicación diferente editando el contenido de los íconos.
- **Sistemas de autor basados en línea de tiempo:** Estas herramientas proporcionan una interfaz para el manejo de eventos multimedia en el tiempo, para controlar el flujo de la aplicación. Macromedia *Flash* y *Adobe Director* son ejemplos de sistemas basados en línea de tiempo.

### 1.5. Ventajas de *Neobook* para la construcción de multimedia

Teniendo en cuenta los criterios antes mencionados se empleará el lenguaje de autor *Neobook V5* como lenguaje principal integrador. El mismo es capaz de brindar un tratamiento especial en la vinculación de sonidos, imágenes, textos, música, animaciones, controles interactivos, etc.

*Neobook* soporta la mayoría de los formatos existentes e incluye un gran repertorio de efectos para el tratamiento de animaciones. *Neobook* no tiene como objetivo igualar o superar sistemas específicos existentes para tratamiento de imágenes, animación 3D, edición de audio, etc., sino que se hace compatible a éstos para

manipular en sus presentaciones o páginas, los productos ya elaborados y perfeccionados por estas herramientas, convirtiéndose en una herramienta pequeña y liviana. Esta aplicación de autor, se destaca también porque incrementa su potencia brindando la posibilidad de usar código externo estandarizado, como es el uso de funciones de bibliotecas existentes (DLL) y los agregados o *Plugins* (Herold, 2002).

La posibilidad de accionar sobre herramientas e interfaces adaptadas de fácil utilización para leer, escuchar, buscar y comunicar, fácil acceso, control de Bases de Datos brindando ya las herramientas y funciones para su manejo hace de *Neobook* una herramienta muy poderosa.

#### **1.6. Gestor de base de datos: Access**

La base de datos está implementada en el gestor de Base de Datos (B.D) *Access*, el cual es multiusuario. Este gestor es una implementación cliente/servidor donde sus principales virtudes son la gran velocidad, robustez y facilidad. En el mundo 9 de cada 10 computadoras usan *Windows* como su sistema operativo, el 98 % de las computadoras que usan *Windows* tienen el *Microsoft Office* instalado, por lo tanto, el Gestor de base de datos *Access* está en casi todas las computadoras del mundo, es decir al diseñar una multimedia interactiva con una base de datos en *Access*, le da la posibilidad de que se pueda ejecutar en casi todas las computadoras.

#### **1.7. Rational Unified Process (RUP)**

RUP es un proceso de desarrollo de *software* y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. La metodología RUP define claramente quién, cuándo, cómo y qué debe hacerse.

Sus principales características son:

- Forma disciplinada de asignar tareas y responsabilidades.
- Desarrollo iterativo.
- Administración de requisitos.

- Uso de arquitectura basada en componentes.
- Control de cambios.
- Modelado visual del *software*.
- Verificación de la calidad del *software*.

RUP es una metodología iterativa e incremental, está centrada en la arquitectura y guiado por los casos de uso. Incluye artefactos y roles. El Proceso Unificado se repite a lo largo de una serie de ciclos que constituyen la vida de un sistema. Cada ciclo concluye con una versión del producto para los clientes.

Cada ciclo consta de cuatro fases: inicio, elaboración, construcción y transición. Cada fase se subdivide a su vez en iteraciones y en cada iteración se invierte esfuerzo en varias disciplinas (o flujos de trabajo) tales como:

- Modelación del negocio.
- Captura de Requerimientos.
- Análisis y diseño.
- Implementación.
- Prueba.

## **CONCLUSIONES DEL CAPÍTULO**

En este capítulo se describió la situación existente con los materiales referentes a la asignatura Fundamentos de la Informática en la carrera de Ingeniería Informática de la Universidad de Granma y se abordaron los principales fundamentos teóricos necesarios del universo de la tecnología multimedia sobre los que se sustentó el desarrollo de esta investigación, constituyendo el marco teórico y referencial de la misma.

## CAPÍTULO 2. ANÁLISIS, DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

### Introducción

Para el desarrollo de cualquier sistema forma parte imprescindible el profundo análisis de las características del mismo a implementar y constituye un eslabón fundamental la comprensión de los procesos existentes en el dominio del problema a resolver. En este capítulo se listan los requerimientos funcionales y no funcionales que debe tener el sistema propuesto, se muestran los diagramas de los casos de usos y se describen los actores que intervienen en dicho sistema. Por último se hace una descripción del diseño, implementación y prueba.

### 2.1. Modelo del dominio

El modelo de dominio es una representación visual estática del entorno real objeto del proyecto. Es decir, un diagrama con los objetos que existen (reales) relacionados con el proyecto que se va a acometer y las relaciones que hay entre ellos.

Se llama "de Dominio" para distinguirlo del Modelo de Negocio que en RUP es un concepto más amplio. El Modelo de Dominio se centra en una parte del negocio, la relacionada con el ámbito del proyecto. En este contexto el término "dominio" representa una parte del "negocio".

Se dice que es estática porque no representa la interacción en el tiempo de los objetos, sino que representa una visión "parada" de las clases y sus interacciones.

El Modelo de Dominio queda recogido en el proyecto en un diagrama de clases del dominio que se muestra a continuación en la figura 1:

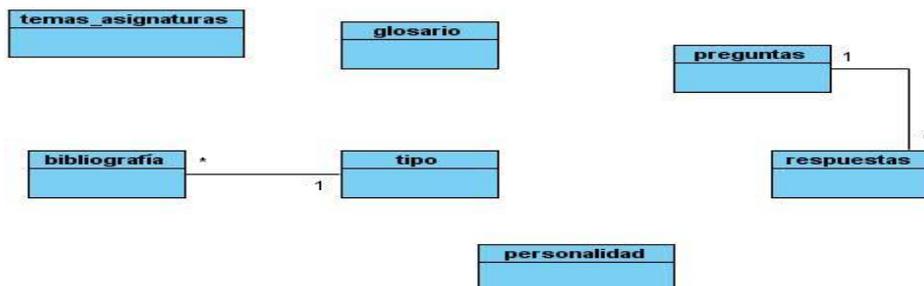


Figura 1: Diagrama de clases del dominio

## **2.2. Requerimientos del sistema**

Un requerimiento es una descripción de una condición o capacidad que debe cumplir un sistema, ya sea derivada de una necesidad de usuario identificada, o bien, estipulada en un contrato, estándar, especificación u otro documento formalmente impuesto al inicio del proceso. La especificación de requerimientos es la base que permite verificar si se alcanzaron o no los objetivos establecidos en el proyecto ya que estos son un reflejo detallado de las necesidades de los clientes o usuarios del sistema y es contra lo que se va a estar verificando si se están cumpliendo las metas trazadas. Los requerimientos se clasifican en requerimientos funcionales y no funcionales (Arias, 2005).

### **2.2.1. Requerimientos funcionales**

Los requerimientos funcionales son características requeridas por el cliente para solucionar un problema o conseguir un objetivo. Es decir, son descripciones del problema, no de la solución (Pérez, 2008).

A continuación se muestra el listado de requerimientos funcionales del sistema:

R1. Actualizar Contraseña.

R2. Mostrar el Gestionar Bibliografía.

R3. Insertar bibliografía.

R4. Modificar bibliografía.

R5. Eliminar bibliografía.

R6. Mostrar el Gestionar Aula Virtual.

R7. Insertar Aula Virtual

R8. Modificar Aula Virtual.

R9. Eliminar Aula Virtual.

- R10. Mostrar el Gestionar Evaluadores.
- R11. Insertar Evaluadores.
- R12. Modificar Evaluadores.
- R13. Eliminar Evaluadores.
- R14. Mostrar el Gestionar Materiales Auxiliares.
- R15. Insertar Materiales Auxiliares.
- R16. Modificar Materiales Auxiliares.
- R17. Eliminar Materiales Auxiliares.
- R18. Mostrar el Gestionar Personalidades.
- R19. Insertar Personalidades.
- R20. Modificar Personalidades.
- R21. Eliminar Personalidades.
- R22. Mostrar el Gestionar Temas.
- R23. Insertar Temas.
- R24. Modificar Temas.
- R25. Eliminar Temas.

### **2.2.2. Requerimientos no funcionales**

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido o confiable. En muchos casos se vinculan a requisitos funcionales y generalmente determinan el éxito del producto. Es decir,

conocido lo que el sistema debe hacer, puede saberse cómo ha de comportarse, qué cualidades debe tener o cuán rápido o grande debe ser (Hernández, 2005).

### **Apariencia o interfaz externa**

- Diseño sencillo, orientado al entorno de trabajo del cliente para que se sienta identificado con la aplicación.
- Los colores deben ser claros y agradables a la vista del usuario.
- El sistema debe tener una interfaz cómoda, facilitando la navegación y evitando la utilización en exceso de imágenes.
- El vocabulario utilizado será en idioma español exclusivamente.

### **Usabilidad**

- El sistema será de fácil uso, para todos los usuarios, aunque no tenga los conocimientos previos de aplicaciones multimedia.

### **Software**

- El Sistema podrá ejecutarse en cualquier PC con el Sistema Operativo Windows instalado y el paquete Microsoft Office 2003.

### **Hardware**

- Las computadoras deben ser Pentium a 2.67 GHz o superior.
- Las computadoras deben tener 128 MB de RAM o superior.
- Un lector de CD o entrada USB para extraer el instalador de la plataforma educativa correspondiente.

### **Seguridad**

- Garantizar que las funcionalidades del sistema se muestren de acuerdo al nivel de usuario que esté activo.
- Para garantizar la seguridad del sistema se cuenta con 2 niveles de acceso: administrador del sistema y usuarios comunes de la aplicación.
- Garantizar que las contraseñas de los usuarios se almacenen de forma segura.

## Confiabilidad

- La información almacenada en el sistema debe corresponderse con la asignatura Fundamento de la Informática de la carrera Ingeniería Informática de la Universidad de Granma.

## Ayuda

- El *software* contará con un manual de usuario que explicará como puede ser utilizado y aprovechado.

### 2.3. Valoración de sostenibilidad del sistema propuesto

La elaboración de un sistema informático puede incidir de forma positiva o negativa sobre los usuarios finales, es por eso que se hace importante realizar una valoración de su impacto en las dimensiones administrativas, socio-humanista, ambiental y tecnológica, para saber hasta qué punto es sostenible el producto informático elaborado.

#### 2.3.1. Dimensión Administrativa

Para la estimación del tiempo de desarrollo de la multimedia se utilizó la técnica de estimación por puntos de casos de uso cuyos pasos se describen a continuación:

##### 1. Cálculo de Puntos de Casos de Uso sin ajustar:

Se calcula a partir de la siguiente ecuación:  $UUCP = UAW + UUCW$  donde:

**UUCP:** Puntos de Casos de Uso sin ajustar

**UAW:** Factor de Peso de los Actores sin ajustar

**UUCW:** Factor de Peso de los Casos de Uso sin ajustar

**1.1 Factor de Peso de los Actores sin ajustar (UAW):** Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema.

Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Factor de Peso	Cant*Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, <i>Application Programming Interface</i> )	1	0*1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	2	0*2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	3	2*3
<b>Total</b>			6

$$UAW = \sum(\text{actores} * \text{Peso}) = 1*0 + 2*0 + 3*2 = 6$$

**1.2 Factor de Peso de los Casos de Uso sin ajustar (UUCW):** Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una **transacción** se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia y está representada por uno o más pasos del flujo de eventos principal del Caso de Uso, pudiendo existir más de una transacción dentro del mismo Caso de Uso. Los criterios se muestran en la siguiente tabla:

Tipo de Caso de Uso	Descripción	Factor de Peso	Cant*Peso
Simple	El Caso de Uso contiene de 1 a 3 Transacciones.	5	5*13
Medio	El Caso de Uso contiene de 4 a 7 Transacciones.	10	0*10
Complejo	El Caso de Uso contiene más de 8 Transacciones.	15	0*15

$$UUCW = \sum CU*Peso=5*12+0*10+0*15=60$$

Entonces ya tenemos los Puntos de Casos de Uso sin ajustar (UUCP) al calcular

$$UUCP = UAW + UUCW =65+6=71$$

**2. Cálculo de Puntos de Casos de Uso ajustados:** Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación: **UCP = UUCP x TCF x EF.**

Donde:

**UCP:** Puntos de Casos de Uso ajustados

**UUCP:** Puntos de Casos de Uso sin ajustar

**TCF:** Factor de Complejidad Técnica

**EF:** Factor de Ambiente

**2.1 Factor de Complejidad Técnica (TCF):** Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Factor	Descripción	Peso	Valor	Peso*Valor
T1	Sistema distribuido	2	0	2*0
T2	Objetivos de performance o tiempo de respuesta	1	5	1*5
T3	Eficiencia del usuario final	1	1	1*1
T4	Procesamiento interno complejo	1	2	1*2
T5	El código debe ser reutilizable	1	3	1*3
T6	Facilidad de instalación	0.5	5	0.5*5
T7	Facilidad de uso	0.5	5	0.5*5
T8	Portabilidad	2	0	2*0
T9	Facilidad de cambio	1	5	1*5
T10	Concurrencia	1	5	1*5
T11	Incluye objetivos especiales de seguridad	1	3	1*3
T12	Provee acceso directo a terceras partes	1	0	1*0
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	0	1*0
<b>Total</b>				<b>31</b>

El Factor de Complejidad Técnica se calcula mediante la siguiente ecuación:

$$TCF= 0.6 + 0.01 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i )=0.6+0.01*31=0.91$$

**2.2 Factor de Ambiente (EF):** Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de Ambiente. El cálculo del mismo es similar al cálculo del Factor de Complejidad Técnica, es decir,

se trata de un conjunto de factores que se cuantifican con valores de 0 a 5. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores.

Factor	Descripción	Peso	Peso*Valor
E1	Familiaridad con el modelo de proyecto utilizado	1.5	1.5*2
E2	Experiencia en la aplicación	0.5	0.5*2
E3	Experiencia en orientación a objetos	1	1*3
E4	Capacidad del analista líder	0.5	0.5*3
E5	Motivación	1	1*5
E6	Estabilidad de los requerimientos	2	2*5
E7	Personal part-time	-1	-1*0
E8	Dificultad del lenguaje de programación	-1	-1*2
<b>Total</b>			<b>21.5</b>

El Factor de Ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \Sigma (\text{Peso}_i \times \text{Valor asignado}_i) = 1.4 - 0.03 * 21.5 = 0.755$$

### 3. De los Puntos de Casos de Uso a la estimación del esfuerzo (E)

Se contabilizan cuántos factores de los que afectan al Factor de Ambiente están por debajo del valor medio (3), para los factores E1 a E6, en este caso solo **2** está por debajo de tres.

Se contabilizan cuántos factores de los que afectan al Factor de Ambiente están por encima del valor medio (3), para los factores E7 y E8, en este caso no existe ninguno que cumpla dicha condición.

$$T = 2 + 0 = 2$$

Como el total es 2 se utiliza el **Factor de Conversión 20 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 20 horas-hombre.

El esfuerzo en horas-hombre en la implementación viene dado por:  **$E = UCP \times CF$**   
donde:

**E:** esfuerzo estimado en horas-hombre

**UCP:** Puntos de Casos de Uso ajustados

**CF:** Factor de Conversión

**$UCP = UUCP \times TCF \times EF = 71 \times 0.91 \times 0.755 = 48.78$**

**$CF = 20$  horas/hombres**

**$E = 48.78 \times 20 = 975.6$  horas/hombre**

#### 4. Calcular esfuerzo de todo el proyecto

Actividad	% esfuerzo	Valor esfuerzo
Análisis	10%	243.9 horas-hombre
Diseño	20%	487.8 horas-hombre
Implementación	40%	975.6 horas-hombre
Prueba	15%	365.85 horas-hombre
Sobrecarga	15%	365.85 horas-hombre
<b>Total</b>	<b>100%</b>	<b>2439 horas-hombre</b>

Suponiendo que una persona trabaje 8 horas por día, y un mes tiene como promedio 30 días; la cantidad de horas que puede trabajar una persona en 1 mes es 240 horas

Si  **$E_T = 2439$  horas-hombre** y por cada 240 horas se tiene 1 mes esto daría un  **$E_T$  de aproximadamente 10 mes-hombre.**

Esto quiere decir que 1 persona puede realizar el problema analizado en más o menos 10 meses.

Como el equipo de desarrollo está formado por 2 personas y todas realizan el mismo esfuerzo entonces el problema analizado puede terminarse en aproximadamente **5 meses.**

Para toda la fase de diseño, implementación, y prueba se utilizará equipamiento ya existente en la UDG por lo que en este aspecto no se incurrirá en ningún gasto.

Para la implantación del sistema en la UDG no se tendrá que mejorar el equipamiento existente, ya que la aplicación se realizará en función de los requerimientos técnicos que posee el centro.

El consumo de electricidad asociado al funcionamiento del producto informático no incrementará el que ya existe en la institución, debido a que será instalado en máquinas ya existentes y de actual explotación.

No aporta ingresos directamente a la entidad ya que no se realizará con ese fin sino para facilitar el estudio de los estudiantes involucrado en el proceso de gestión de la información de la asignatura Fundamentos de la Informática en la carrera Ingeniería Informática en la Universidad de Granma.

Teniendo en cuenta los beneficios que reporta se considera que es sostenible desde el punto de vista administrativo.

### **2.3.2. Dimensión Socio-Humanista**

En Cuba el factor humano es muy importante, por lo que para cualquier institución es primordial la comodidad de los trabajadores, trayendo esto consigo un mayor rendimiento en las tareas que ellos realizan. Se puede garantizar que una vez implantado el sistema informático mejorará la calidad de vida de los estudiantes y aumentará el rendimiento de los mismos, ya que contarán con una herramienta fiable y segura.

Es importante destacar que la implantación de este nuevo sistema está diseñado para favorecer a los estudiantes en el estudio de la asignatura Fundamentos de la Informática. Además el uso de las nuevas tecnologías ayuda a elevar el nivel profesional y cultural.

El sistema informático contribuye en gran medida con la informatización que el país está llevando a cabo en estos momentos en todas las esferas sociales. El mismo posee la facilidad de adaptarse a todas aquellas facultades que tienen similitudes en cuanto a los procesos docentes con la Facultad de Ciencias Técnicas de la UDG, por las características de flexibilidad ante cambios que posee es posible la generalización del sistema a todas las universidades del país.

Por lo antes expuesto se concluye que el sistema informático es sostenible, de acuerdo a la dimensión socio-humanista.

### **2.3.3. Dimensión Ambiental**

La implantación del sistema ahorra recursos que generan daños al medio ambiente, porque contribuye en la disminución del consumo de artículos que tienen su base en los recursos de la naturaleza, tal es el caso de los materiales de oficina, principalmente el papel que se obtiene a partir de una serie de procesos que inician con la tala de árboles.

El sistema brinda al usuario una interfaz agradable, cuidando en todo momento el uso de los colores, los cuales están en la gama de los azules y los grises siendo estos colores familiares para el usuario debido a que la mayoría de las aplicaciones con las que trabaja se encuentra en esta gama de colores y no resultan dañinos a la vista. No son colores muy brillosos, ni oscuros, más bien son colores frescos, que le brindan al usuario una sensación de relajamiento y tranquilidad, de forma que se disminuye la posibilidad de desarrollar alguna enfermedad ocular.

Desde el punto de vista psicológico el *software* influye mucho sobre el usuario, ya que anteriormente debía emplear una gran cantidad de tiempo para obtener los reportes e informaciones que necesitaba, pues se hacía a mano y ahora el sistema lo resuelve en algunos segundos.

En todo momento se tuvieron en cuenta las necesidades de los usuarios, se utilizó un tamaño de letra adecuado para la correcta visualización del contenido. Se recomienda a los usuarios una correcta postura en las sillas, usar protectores de

pantalla para cuidar la visión, así como poner el monitor a la altura de los ojos y frente a ellos, para evitar los problemas de visión, de columna y de cervical.

La implantación del *software* no contribuye de modo alguno al deterioro del medio ambiente por causa de contaminación por ruido, interferencia, etc. Todo lo contrario, asegura un paso más en el cuidado del mismo, por lo que se considera que es sostenible teniendo en cuenta la dimensión medio ambiental.

#### **2.3.4. Dimensión Tecnológica**

La Facultad de Ciencias Técnicas de la UDG cuenta con la infraestructura electrónica necesaria para la correcta implantación y uso del *software*, pero se hace necesaria la capacitación del personal que interactuará con el mismo, mediante el uso de la Ayuda del Sistema.

Existe una estrecha comunicación entre los desarrolladores de la aplicación y los usuarios finales, logrando así que el sistema satisfaga las necesidades y expectativas de los mismos.

Los códigos están debidamente comentados, lo que facilita su entendimiento. Se utiliza la programación orientada a objetos, dándole a cada clase y funciones un nombre intuitivo, que con sólo ver el nombre, el que lo lea sabe de qué trata la porción del código. También se adjuntarán documentos que describen como está confeccionado el sistema y la Ingeniería de Software.

Por todo lo anteriormente expresado se determina que el sistema es sostenible desde el punto de vista tecnológico.

#### **2.3.5. ¿Es el sistema propuesto sostenible?**

Después de realizado el análisis de sostenibilidad del producto según las dimensiones administrativa, socio-humanista, ambiental y tecnológica se llega a la conclusión de que la herramienta propuesta trae consigo un grupo de aportes prácticos a la gestión administrativa de los procesos docentes en la Facultad de Ciencias Técnicas de la UDG y cumple con todos los requerimientos antes

señalados, por lo que se puede decir que constituye un producto informático sostenible.

## 2.4. Modelo de Casos de Uso del Sistema

Dado que el sistema constituye un producto sostenible se pasa a representar los requerimientos funcionales del sistema mediante un diagrama de casos de uso utilizando las facilidades que brinda el UML. Para ello se definen cuales son los actores que van a interactuar con el sistema y los casos de uso que representan las funcionalidades.

### 2.4.1. Actores del sistema

Los actores del sistema no son parte de él. Es decir, pueden realizar un intercambio de información con el sistema, así como ser un recipiente pasivo de información y pueden representar el rol que juega una o varias personas, un equipo o un sistema automatizado.

En la tabla 2.1 se muestran las descripciones de los actores del sistema.

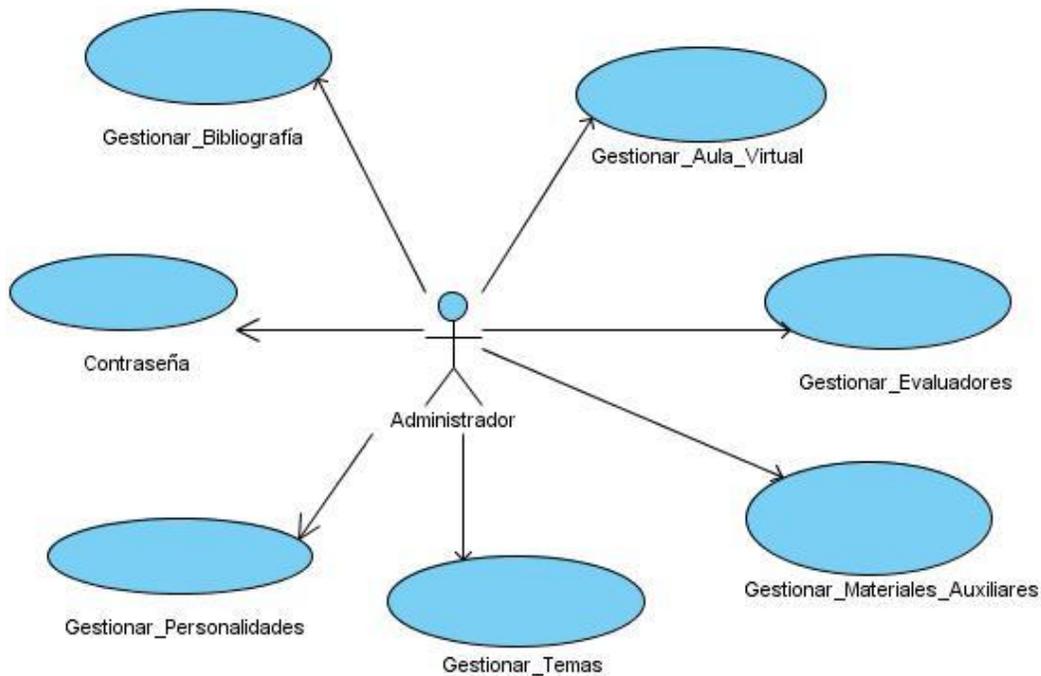
**Tabla 2.1** Descripciones de los actores del sistema.

Actores	Descripción	Requerimientos asociados
Administrador	Podrá acceder a todas las opciones del usuario, además de la actualización de la base de datos, modificando, insertando o	
Usuario	Podrá interactuar con el <i>software</i> y tendrá acceso a todas las opciones del sistema: secciones, ayuda, evaluadores, cambiar audio, acceder a la galería, navegar mediante los botones.	

### 2.4.2. Diagramas de Casos de Uso del Sistema

Los casos de uso son una técnica para especificar la forma en que se comporta un sistema teniendo como punto de partida la identificación de los requerimientos del sistema.

En la Figura 2 se puede apreciar el diagrama de caso de uso del sistema para el usuario administrador.



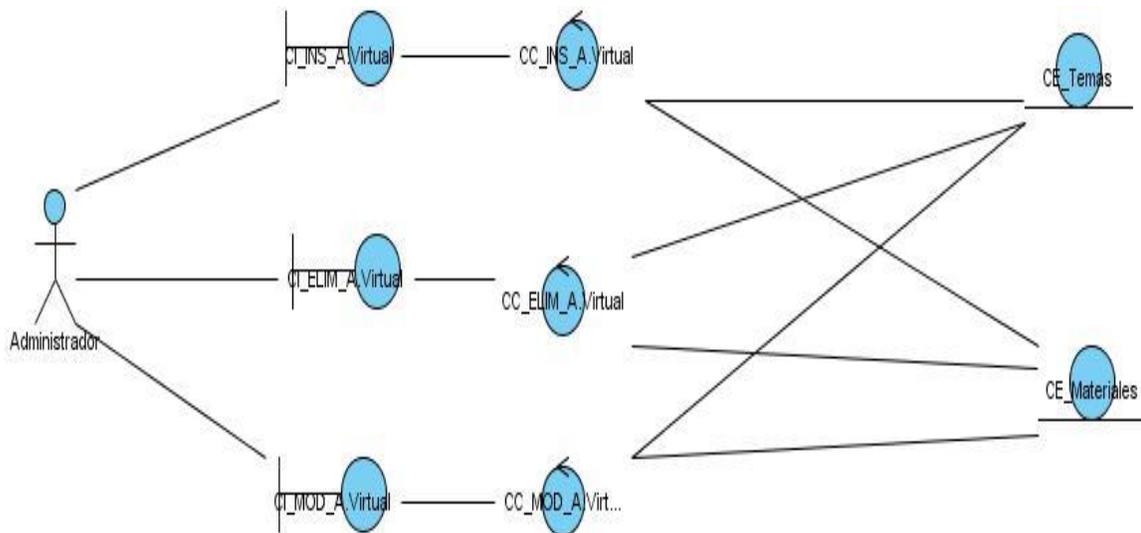
**Figura 2** Diagrama de caso de uso del sistema para el usuario administrador

## 2.5. Análisis

El modelo de análisis es fundamental en el proceso de estructuración y refinamiento de los requisitos, permite razonar sobre los aspectos internos del sistema, incluyendo los recursos compartidos internos, proporciona un mayor poder expresivo y una mayor formalización así como una arquitectura centrada en el mantenimiento de aspectos tales como la flexibilidad ante los cambios y la reutilización. Esta estructura no solo es útil para el mantenimiento de los requisitos del sistema como tal, sino que también se utiliza como entrada en las actividades de diseño y de implementación (Jiménez, 2009).

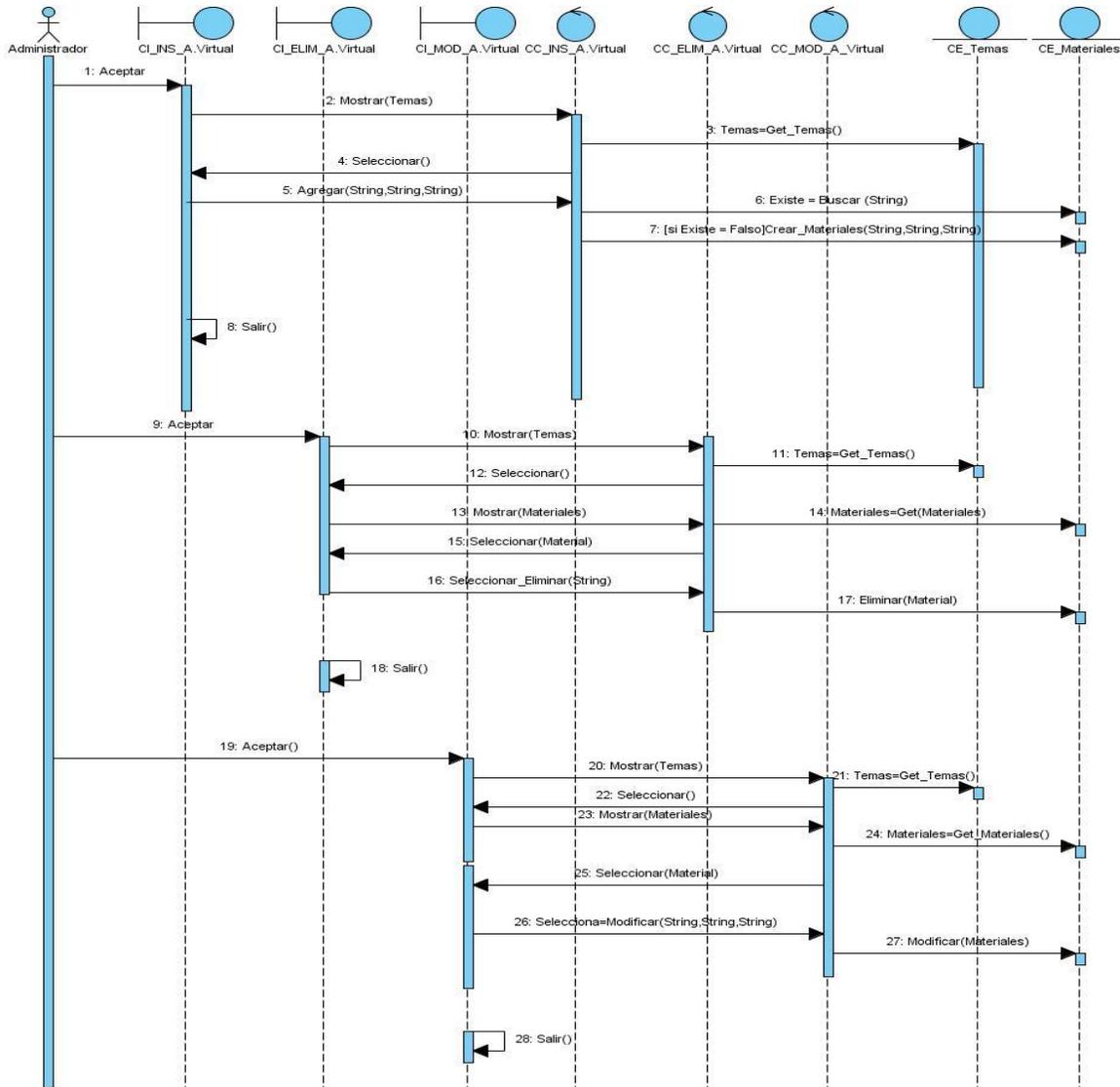
Entre los artefactos del modelo del análisis se encuentran las clases del análisis, los diagramas de clases del análisis, diagramas de interacción (secuencia o colaboración) y la descripción del flujo de sucesos-análisis que completa los diagramas de colaboración.

En la Figura 3 muestra el diagrama de clase de análisis del caso de gestionar aula virtual.



**Figura 3** Diagrama de análisis del caso de uso gestionar aula virtual.

Para llevar a cabo este flujo en el desarrollo de la multimedia se realizaron los diagramas de colaboración, en las Figura 4 se pueden apreciar el diagrama de secuencia del caso de uso gestionar aula virtual.



**Figura 4** Diagrama de secuencia del caso de Gestionar aula virtual

El resultado obtenido en el análisis constituye una entrada esencial en el diseño pues proporciona una comprensión detallada de los requisitos e impone una estructura de la aplicación que en el modelo del diseño se conserva lo más fielmente posible.

## 2.6. Modelo del Diseño

El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso el cual tiene la responsabilidad de modelar el sistema y encontrar la forma para que este soporte todos los requisitos necesarios para cumplir su objetivo.

### 2.6.1. Mapa de Navegación

Un mapa de navegación es la representación gráfica de la organización de la información. Expresa todas las relaciones de jerarquía y secuencia, y permite elaborar escenarios de comportamiento de los usuarios (Romero, 2009).

En la figura 5 se muestra el mapa de navegación de la multimedia

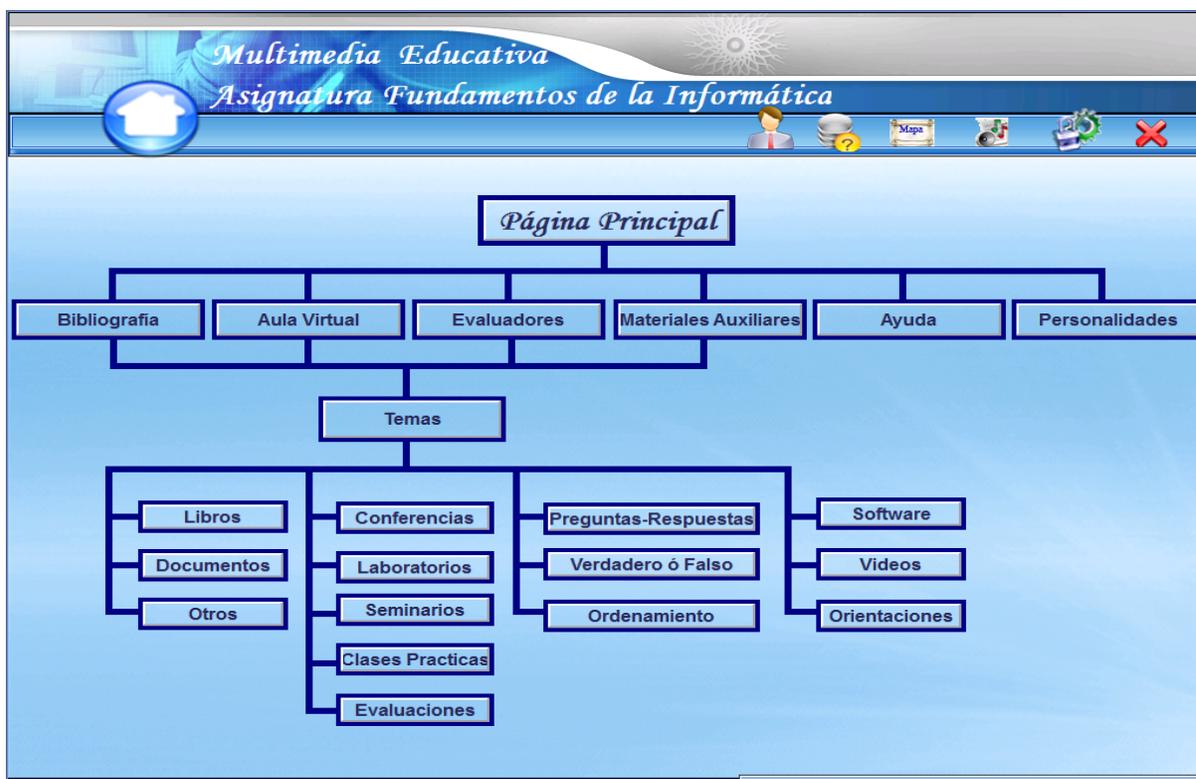


Figura 5 Mapa de Navegación

### 2.6.2. Diseño de la base de datos

El diseño de la base de datos es una de las etapas más importantes en el ciclo de vida de desarrollo del *software*. Aquí se definen las tablas, sus atributos y las relaciones entre las tablas. Teniendo como objetivo asegurar que los datos persistentes se almacenen de forma consistente y eficientemente, definiendo el comportamiento que debe ser implementado en la base de datos.

El proceso de normalización es un estándar que consiste, básicamente, en un proceso de conversión de las relaciones entre las entidades, evitando:

- La redundancia de los datos: repetición de datos en un sistema.

- Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.
- Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.
- Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos (Jacobson, *et al.*, 2000).

**1ra Forma Normal:** Una varrel está en 1ra Forma Normal si y solo si, en cada valor válido de esa varrel, toda tupla contiene exactamente un valor para cada atributo.

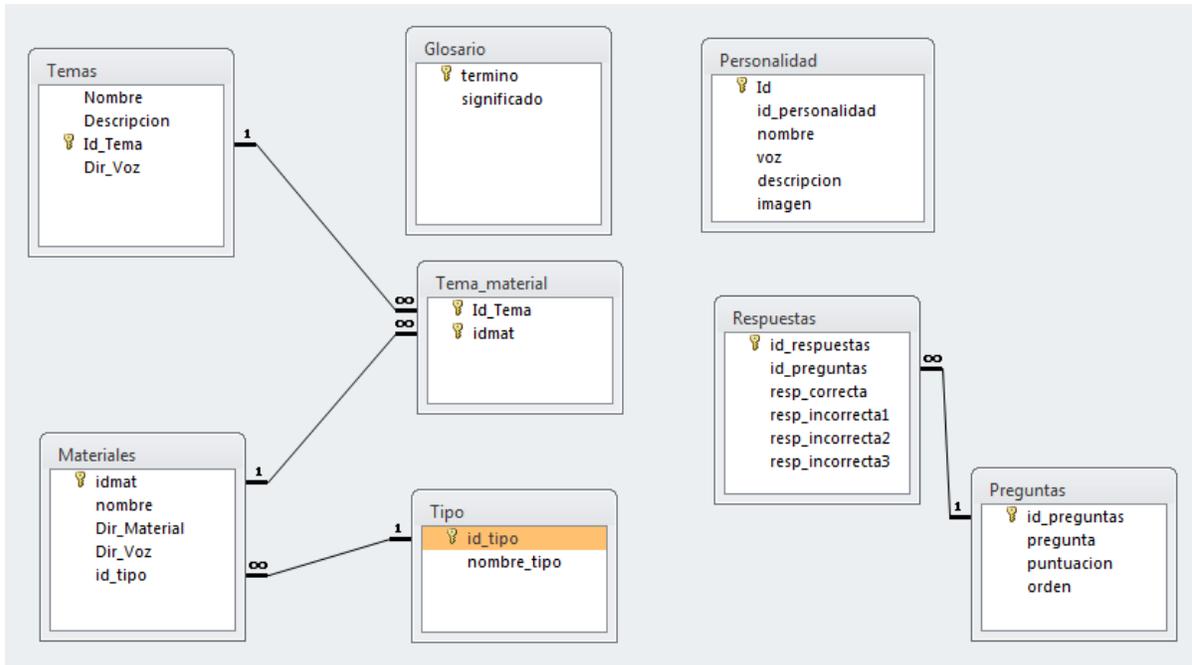
**2da Forma Normal:** Una varrel está en 2da Forma Normal, si y solo si está en 1ra Forma Normal y todo atributo que no sea clave, es dependiente irreduciblemente de la clave primera (Mejía, 2007).

**3ra Forma Normal:** Una varrel está en 3ra Forma Normal, si y solo si está en 2da Forma Normal, y todos los atributos que no son clave, son dependientes de forma no transitiva de la clave primaria.

**Forma Normal de Boyce-Codd:** Una varrel está en Forma Normal de Boyce-Codd si y solamente si toda dependencia funcional no trivial, irreducible a la izquierda, tiene una clave candidata como su determinante (Date, 2003).

Analizando las definiciones descritas anteriormente se llegó a la conclusión de que la base de datos en cuestión ya está normalizada debido a que cumple con todas las reglas de normalización, empezando por la primera forma normal y terminando con la forma normal de Boyce –Codd.

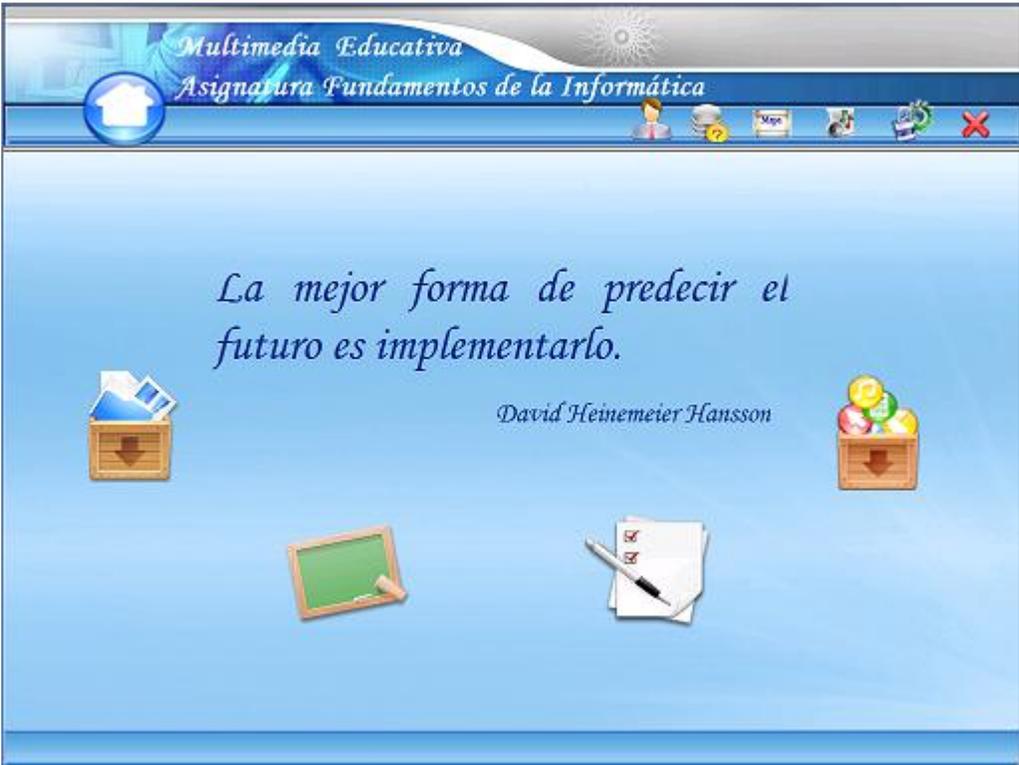
A continuación se muestra el diagrama entidad-relación de la base de datos del sistema.



**Figura 6** Diagrama entidad-relación de la base de datos.

### 2.7.3 Guión para el software multimedia

El guión multimedia es un documento que contiene una descripción de todas y cada una de las escenas del producto audiovisual. Estas escenas están compuestas por los elementos multimedia que en su integración constituyen las distintas pantallas que se muestran en la aplicación.

<p>Recursos Gráficos</p>	<p>Imágenes que le dan al usuario una noción sobre el tema abordado en la multimedia en las distintas secciones que se pueden visitar, así como botones para mostrar las secciones complementarias.</p> 
<p>Sonido</p>	<p>Música de fondo MP3.</p>
<p>Texto</p>	<p>Nombre del sistema, guías sobre lo que hacen las distintas opciones mostradas para navegar por la multimedia y una frase relacionada con el software.</p>

Acción	<p>Se mostrarán en la pantalla botones que le permitirán al usuario navegar por la multimedia en dependencia de la opción seleccionada.</p> <p>Botón para visitar la sección Bibliografía.</p> <p>Botón para acceder a la sección Aula Virtual.</p> <p>Botón para ingresar a la sección Evaluadores.</p> <p>Botón para visitar la sección Materiales Auxiliares.</p>
--------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

<p>Figura 8. Aula Virtual</p>	
<p>Recursos Gráficos</p>	<p>Muestra el tema seleccionado con anterioridad y las distintas tipologías de información (Conferencias, Laboratorios, Seminarios, Clases Prácticas y Evaluaciones) que se pueden visitar para acceder a toda su información, así como botones para mostrar las secciones complementarias.</p>  <p>The screenshot shows a web-based interface for 'Multimedia Educativa' focused on the course 'Asignatura Fundamentos de la Informática'. The interface includes a navigation bar with icons for home, user, and other resources. Below this, there are tabs for 'Conferencias', 'Laboratorios', 'Seminarios', 'Clases Prácticas', and 'Evaluaciones'. The main content area displays 'TEMA 2. Conceptos básicos de Hardware y Software: familiarización con la computadora.' with sub-sections for 'Contenido' and 'Objetivos'. The 'Contenido' section lists three bullet points: 'Definir el término computadora. Clasificación.', 'Definir el término hardware. Tipos de hardware.', and 'Definir el término software. Tipología de software.'. The 'Objetivos' section is partially visible at the bottom. The interface is designed with a blue and white color scheme and includes various icons for navigation and user interaction.</p>

Sonido	Música de fondo MP3 o la voz del texto que se está mostrando en pantalla.
Texto	Se mostrarán al ingresar a la pantalla el nombre del sistema y luego el texto según la opción que se ejecute.
Acción	El usuario podrá luego de haber seleccionado un tema obtener todos los materiales acerca del mismo. Botones para navegar y controlar el audio.

Figura 9. Administrar Multimedia

Recursos Gráficos	<p>Muestra los botones para mostrar las secciones complementarias.</p> 
Sonido	Música de fondo MP3 o la voz del texto que se está mostrando en pantalla.

<p>Texto</p>	<p>Se mostrará una frase al ingresar en la pantalla, el nombre del sistema y luego el texto según la opción que se ejecute.</p>
<p>Acción</p>	<p>El usuario podrá seleccionar una sección al cual podrá tener acceso y obtener información acerca del mismo.</p> <p>Botones para navegar, controlar el audio.</p>

Figura 10. Añadir al Aula Virtual

<p>Figura 10. Añadir al Aula Virtual</p>	
<p>Recursos Gráficos</p>	<p>Muestra las distintas opciones a realizar con los documentos pertenecientes a dicha sección, así como botones para mostrar las secciones complementarias.</p> 
<p>Sonido</p>	<p>Música de fondo MP3 o la voz del texto que se está mostrando en pantalla.</p>

Texto	Se mostrará el nombre del sistema y luego el texto según la opción que se ejecute.
Acción	El usuario podrá seleccionar el tipo de material y el tema, cargar el material, la voz y por último añadirlo a la base de datos.  Botones para navegar y controlar el audio.

## 2.7. Implementación de la multimedia

En la etapa de implementación se explica cómo fueron realizados algunos de los artificios utilizados en el sistema y se describen las principales funcionalidades que ofrece la multimedia en correspondencia con los objetivos trazados, el cumplimiento de los principios multimedia y la validación de la factibilidad del *software*.

### 2.7.1. Herramientas utilizadas en el desarrollo del software

Para la implementación de la multimedia se utilizó como lenguaje de autor principal Neobook 5.6.2 explotando las posibilidades que brinda el mismo para la creación de aplicaciones multimedia. Es una herramienta multimedia orientada a eventos con excelente capacidad para interactuar con el usuario.

### 2.7.2 Peculiaridades de la programación con el uso de los script

El presente sistema multimedia es un *software* orientado a eventos, que lo capacita para interactuar con el usuario y cuyo formato está organizado a modo de páginas, lo que le proporciona organización. La programación en cada uno de estos eventos se construye mediante el uso de scripts. Los scripts en Neobook no son más que líneas de código o secuencia de acciones que se ejecutarán una tras otra en cada uno de los eventos en que estén implementadas. En el desarrollo de nuestra aplicación, utilizando Neobook 5.6.2 y sus scripts característicos, tuvo un singular peso el conocimiento de determinados conceptos, los cuales se exponen a continuación:

## **Variables y subrutinas**

El concepto y las funcionalidades de una variable en *NeoBook*, no difiere a los de lenguaje de propósito general, simplemente es un área de la memoria del ordenador que puede usarse para almacenar información temporalmente, mientras se está ejecutando su publicación.

El tratamiento de variables es de gran importancia en el desarrollo de cada uno de los eventos que posee la multimedia porque mediante éstas se puede asegurar el estado de los mismos y la correcta navegación por el sistema. Constituye el eslabón fundamental en la óptima utilización de las subrutinas implementadas. La mayoría de los objetos usan variables para almacenar información acerca de su estado o el contenido a mostrar, tales como los botones de opción, Entrada de texto y Artículos, los que han sido de vital importancia en el trabajo con los ejercicios de autoevaluación y el despliegue de información.

La subrutina es una herramienta que tiene mucho peso en aras de lograr una mayor eficiencia y modularidad en la programación con script. Las mismas no son más que segmentos de código que se almacenan en una librería y que pueden ser llamadas desde cualquier punto del código script. Su función fundamental es evitar la repetición innecesaria de código.

Durante la implementación del sistema esta herramienta significó un ahorro sustancial de tiempo y esfuerzo; a continuación se muestra una de las subrutinas utilizadas.

```

Math "[pag]+1" "" "[temp]"
dbpGotoRecord "bd" "Personalidades" "[temp]"
SetVar "[titulo]" "[bd.viuwperiodopersonalidades.nombres y apellidos]"
SetVar "[documento]" "[bd.viuwperiodopersonalidades.biografia]"
endif
endif
endif
Return

:cargardocdocumentacion
SetVar "[material]" ""
dbpGotoRecord "bd" "viuwtipomaterial" "[temp]"
SetVar "[material]" "[bd.viuwtipomaterial.material]"
Return

:cargardescripsoftware
SetVar "[descripsoftware]" ""
dbpGotoRecord "bd" "viuwtiposoftware" "[temp]"
SetVar "[descripsoftware]" "[bd.viuwtiposoftware.Descripcion]"
Return

```

**Figura 11.** Implementación de la subrutina cargar documentación

La subrutina cargar documentación dado un tipo de documento almacenado en la variable [temp] (libro, carta, tesis, etc.) carga la información del mismo desde la base de datos para mostrarla al usuario, evitando así repetir éste fragmento de código para cada tipo de documento.

```

DisableObject "B1"
EnableObject "B2"
EnableObject "B3"
EnableObject "B4"
EnableObject "B5"
EnableObject "B6"
EnableObject "B7"

Math "[pag]+1" "" "[temp]"
GoSub "cargardocdocumentacion"

```

**Figura 12.** Utilización de la subrutina Cargar documentación

### 2.7.3 Utilización de eventos

Los eventos pueden ser de páginas o de objetos. Los eventos de objeto más utilizados en nuestro trabajo son los siguientes:

1. LeftClick: Ocurre cuando se produce un clic izquierdo encima del objeto.

2. Mouse Enter: Ocurre cuando el mouse se sitúa dentro del objeto.
3. Mouse Exit: Es provocado cuando el mouse abandona el perímetro del objeto.

### **Los eventos de página son:**

- Page Enter: Es provocado cuando se ejecuta una página.
- Page Exit: Es provocado cuando se abandona la página actual.

Las acciones son un conjunto de operaciones que se ejecutan al ocurrir un evento determinado. Por ejemplo:

### **Acciones de navegación:**

- GotoPage: Es utilizado cuando se quiere dirigir el curso de la navegación hacia una página especificando su nombre.
- GotoNextPage: Se utiliza para visitar la página siguiente a la actual.
- ReturnToPage: Se usa para retornar al la página visitada anteriormente.

### **Acciones propias de los objetos:**

- ShowObject: mostrar el objeto.
- HideObject: Ocultar el objeto.
- SetObjectFont: Cambia las propiedades de la fuente de un objeto.

### **Acciones para la gestión de Archivos:**

- FileCopy: Copia un archivo externo.
- FileErase: Borra un archivo externo.
- FileExists: Determina si existe un archivo externo.

### **Acciones para el manejo de las bases de datos:**

- dbpOpen: Abre una base de datos de extensión .dbp.
- dbpCreate: Crea y abre una base de datos en una ruta especificada, si esta ya existe la abre simplemente.
- dbpFirst: Muestra el primer registro de la base de datos especificada.
- dbpNext: Muestra el registro siguiente a partir de la posición actual de la base de datos especificada.
- dbpFind: Muestra la primera coincidencia en dependencia del criterio especificado.
- dbpFindNext: Muestra la siguiente coincidencia referente a un criterio anterior.
- dbpGotoRecord: Se mueve en la base de datos al número de fila que se le pasa.
- dbpSort: Ordena una base de datos en dependencia a un criterio especificado.

#### **2.7.4. Utilización de recursos audiovisuales y trabajo con animaciones**

El entendimiento de *Neobook V5* con varios formatos como imágenes, videos y sonidos permite que una misma información se presente de múltiples maneras, utilizando cadenas de asociaciones de ideas similares a las que emplea la mente humana, al no brindar herramientas potentes para dar tratamientos a estos formatos se necesita la ayuda de algunos sistemas especializados como:

1. Adobe PhotoShop CS: Sobre este se construyeron y editaron las imágenes utilizadas en la aplicación.
2. TextAloud: Aquí se construyeron todos los ficheros de audio.
3. Xara Webstyle 4.0: Fue clave en la construcción de prototipos de botones, gif, animados e iconos.

Una de las ventajas de la utilización de una multimedia como medidor didáctico radica en la muestra de la información por varios canales. En este sentido se trabajó en la confección de un reproductor de video, con las funciones estándar de una herramienta de este tipo y que permitiera la reproducción de los videos existentes sin necesidad de abandonar la aplicación; además se manipuló el curso de la reproducción de los archivos de audio y música de fondo con opciones de stop y play.

Los gif animados y las películas flash fueron utilizadas indistintamente explotando fundamentalmente su efecto estético y su dinamismo; a fin de lograr un rendimiento óptimo de las mismas se utilizaron herramientas que brinda Neobook, como son los bucles (permiten la repetición de los efectos visuales), cambio de características, activación o desactivación, entre otras. La utilización de animaciones para mostrar contenidos favorece el proceso de enseñanza y aprendizaje; es por eso la muy frecuente utilización de este recurso como medio de transmisión de información en el sistema “Ponderación de la Historia de Cuba. Acercamiento a la Universidad de Oriente”. En este aspecto el Neobook brinda un conjunto fuerte de herramientas para el trabajo con animaciones entre las que se destacan:

- Ocultar y mostrar componentes.
- Habilitar y deshabilitar componentes.
- Opciones de transición.
- Cambio de tamaño y color de los componentes.
- Retardo de tiempo y cronómetro.
- Opciones de movimiento de los componentes.
- Agrupamiento de componentes.

A la hora de implementar una animación es vital tener en cuenta el factor tiempo como guía; fue por esto la importancia de la utilización de los retardos de tiempos y cronómetros utilizados en la implementación de las animaciones en la

aplicación. Estos, asociados a técnicas visuales y la utilización de recursos audiovisuales como las imágenes, películas flash, gif animados, ficheros de audio, entre otros, garantizaron un correcto y efectivo mensaje por parte de cada una de las animaciones implementadas en ésta multimedia. Es importante destacar que las animaciones fueron programadas con el código script de Neobook, y que son invocadas en cada uno de los eventos que fueron considerados como óptimos.

### **2.7.5. Utilización de recursos matemáticos y trabajo con la base de datos**

Durante la implementación del software se utilizaron cálculos matemáticos, pero el peso fundamental de estos recayó en la creación de los ejercicios de autoevaluación.

Uno de los aportes fundamentales de la multimedia es su capacidad para evaluar y visualizar el resultado de los ejercicios respondidos previamente por el usuario. Para esto fue necesaria la implementación del sistema de evaluación brindado por los especialistas. Para la programación de estos métodos se utilizaron las funciones matemáticas básicas brindadas por el Neobook V5 (suma, resta, producto, cociente, parte entera, entre otras), así como la información brindada por objetos (botón de opción, botón de confirmación) que en conjunto con la definición de variables globales tiene como objetivo:

- Validar que se haya dado respuesta a cada una de las preguntas propuestas.
- Contar la cantidad de respuestas correctas.
- Utilizar criterio de especialista para asignar y visualizar su calificación correspondiente.
- Actualizar en pantalla las respuestas correctas.
- Almacenar calificación de cada una de las preguntas por temas.
- Promediar y visualizar calificación global por tema.

Para la realización de este método, y teniendo en cuenta los objetivos que se perseguían, se decidió enfocar el trabajo en dos áreas fundamentales:

1. Dotar al sistema con una base de datos que permita almacenar toda el material a mostrar en la multimedia y los diferentes módulos de evaluación con el objetivo de que este sea actualizable sin la ayuda del especialista que creó el software.
2. Programar el método matemático para cada uno de los diferentes módulos de evaluación

```
Suspend "True"
If "[DBOpen]" "<>" "True"
  FileExists "E:\YudithTesis\BD\bdYudith.mdb" "[Found]"
  If "[Found]" "=" "False"
    dbpCreateAccessDatabase "E:\YudithTesis\BD\bdYudith.mdb" "Password="
    dbpOpenAccessDatabase "bd" "[PubDir]BD\bdYudith.mdb" ""
  Else
    dbpOpenAccessDatabase "bd" "[PubDir]BD\bdYudith.mdb" ""
  EndIf
  SetVar "[DBOpen]" "True"
EndIf
SetVar "[publicacion]" "[PubDir]"

dbpShowErrors "No"
Suspend "False"
```

**Figura 13** .Fragmento de código que pertenece a la creación o apertura de la base de datos.

## 2.8. Prueba

Un aspecto crucial en el control de calidad del desarrollo de *software* son las pruebas y dentro de estas, las pruebas funcionales, en las cuales se hace una verificación dinámica del comportamiento de un sistema, basada en la observación de un conjunto seleccionado de ejecuciones controladas o casos de prueba. Para hacer pruebas funcionales se requiere una planificación que consiste en definir los aspectos a chequear y la forma de verificar su correcto funcionamiento, punto en el cual adquieren sentido los casos de prueba y estos solo se pueden realizar teniendo en cuenta las clases de equivalencia (Ponce, *et al.*, 2010).

Las pruebas evalúan el producto para determinar que cumple con el objetivo previsto, por lo que es necesario diseñar un plan de pruebas que se adapte y sea coherente con la metodología de desarrollo, que proporcione un enfoque de fácil acceso a la estructura para verificar los requisitos y cuantificar su rendimiento, y que identifique las diferencias entre los resultados previstos y los reales (errores o fallas).

Se utilizaron las pruebas de caja blanca, denominadas a veces prueba de caja de cristal, es un método de diseño de casos de prueba que usa la estructura de control del diseño procedimental para obtener los casos de prueba (Pressman, 2002).

### 2.8.1. Casos de Prueba

Un caso de prueba específica una forma de probar el sistema, incluyendo la entrada o el resultado con que se ha de probar y las condiciones bajo las que ha de probarse. Es una forma de organizar las pruebas del sistema relacionadas con la funcionalidad que tenga, es por eso que los casos de prueba están directamente relacionados con los casos de uso que prueban. Es un conjunto de entradas y resultados esperados que ejercitan a un componente con el propósito de causar fallas y detectar defectos.

En la Tabla 2 se muestra el caso de prueba realizado al caso de uso gestionar Aula Virtual.

**Tabla 2.** Caso de prueba: Entrada correcta de los datos del Aula Virtual

Caso de uso:	Gestionar Aula Virtual.
Caso de Prueba:	Entrada correcta de los datos del Aula Virtual.
Entrada:	El administrador introduce correctamente los datos del Aula Virtual en la base de datos.
Resultados:	El sistema muestra el mensaje: “Los datos han sido insertados correctamente”.
Condiciones:	El administrador debe estar autenticado.

**Tabla 3.** Caso de prueba: Entrada incorrecta de los datos del Aula Virtual.

Caso de uso:	Gestionar Aula Virtual.
Caso de Prueba:	Entrada incorrecta de los datos del Aula Virtual.
Entrada:	El administrador introduce incorrectamente los datos del Aula Virtual.
Resultados:	El sistema muestra el mensaje: “Datos incorrectos”.
Condiciones:	El administrador debe estar autenticado.

### 2.8.2. Procedimientos de Prueba

Un procedimiento de prueba es otro artefacto que pertenece al modelo de prueba. Especifica cómo realizar uno o varios casos de prueba o partes de estos. Constituye una guía para quien ejecutará un caso de prueba.

A continuación se muestra un procedimiento de prueba del caso de uso gestionar Eliminar Aula Virtual.

**Tabla 8.** Caso de prueba: Eliminar Aula Virtual.

Acción del actor	Respuesta del sistema
1. El administrador confirma eliminar Aula Virtual.	2. El sistema elimina el Aula Virtual, muestra el mensaje: “Los datos han sido eliminados de la base de datos” y finaliza el proceso.
Pos condiciones:	El Aula Virtual es eliminada, el sistema muestra el mensaje: “Los datos han sido eliminados de la base de datos”.

### Conclusiones del Capítulo

En este capítulo se describió el desarrollo del sistema en la fase de análisis, diseño e implementación. En los que se expone la captura de requisitos, el modelado del sistema, la confección del guión multimedia y en lo concerniente a la implementación,

se aborda el proceso de programación del software, los principios paradigmáticos de las aplicaciones multimedia y se detallaron sus principales funcionalidades.

## CONCLUSIONES

Con la realización de este trabajo:

- Se analizó la situación actual del proceso de gestión de información de la asignatura Fundamentos de la Informática en la carrera Ingeniería Informática en la Universidad de Granma.
- Se implementó una aplicación multimedia para la asignatura Fundamentos de la Informática en la carrera Ingeniería Informática en la Universidad de Granma.

## RECOMENDACIONES

- Consultar la investigación realizada como material de estudio en el desarrollo de aplicaciones similares.
- Enriquecer las utilidades que brinda la multimedia, trabajando sobre la implementación de nuevas funcionalidades.

## BIBLIOGRAFÍA

1. Alarcón, A. y Sandoval, E. (2008). Herramientas CASE para ingeniería de Requisitos. *Cultura Científica* 6.
2. Alvares, M. A. (2001). Qué es HTML, Revisado: 15 de Noviembre 2011, tomado de: <http://www.desarrolloweb.com/articulos/que-es-html.html>
3. Alvarez, M. (2010). Manual de CodeIgniter, Revisado: 21 de Noviembre 2011, tomado de: <http://www.desarrolloweb.com/manuales/manual-codeigniter.html>
4. Alvarez, R. (2001). Explicación de qué diferencia hay entre lenguajes como Java o JavaScript y otros como ASP o PHP. Pro y contras de cada tipo2, Revisado: 11 de Noviembre 2011, tomado de: <http://www.desarrollo-web.com/articulos/239.php>
5. Argudo, J. (2009). *CodeIgniter 1.7*.
6. Arias, M. (2005). La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software. *InterSedes*, 6(10).
7. Árias, M. (2009). *Carmen: una herramienta de software libre para modelos gráficos probabilistas*. Unpublished Tesis Doctoral, Universidad Politécnica de Madrid, España.
8. Becerra, P. (2004). Oficina Central de Calidad Académica y Acreditación Revisado: 24 de Octubre 2011, tomado de: <http://www.unmsm.edu.pe/occa-a/sgcunmsm.html>
9. Bell, D. (2003). *UML basics: An introduction to the Unified Modeling Language*.
10. Boehm, B., Abts, C., Horowitz, E., Brown, W., Madachy, R., Chulani, S., *et al.* (2000). *Cocoma II Model Definition Manual*: Universidad del Sur de California.
11. Briggs, O., Champeon, S., Costello, E. y Patterson, M. (2003). *Cascading Style Sheets* España.
12. Brown, S., Allen, R. y Lo, N. (Eds.). (2009). *Zend Framework in Action*: Manning Publications.
13. Cabrera, M. (2009). *CITAS2.0 Un gestor de reuniones online basado en AJAX*. Universidad Carlos III de Madrid, Leganés.
14. Carrillo, D. (2005). Desarrollo web empresarial con PHP4, tomado de: [http://gluc.unicauca.edu.co/wiki/index.php/Desarrollo\\_web\\_empresarial\\_PHP4](http://gluc.unicauca.edu.co/wiki/index.php/Desarrollo_web_empresarial_PHP4)
15. Claro, R. L. H. y Navarro, D. G. (2010). Estándares de Diseño Web. *Ciencias de la Información*, 41(2), 69 - 71.
16. Cobos, C., Zuñiga, J., Guarín, J., León, E. y Mendoza, M. (2010). CMIN - a CRISP-DM-based case tool for supporting data mining projects *Ingeniería e Investigación*. Bogotá, 30(3), 12.

17. Charte, F. (2007). *Ajax*. España.
18. Date (2003). *Introducción a los Sistemas de Base de Datos* (Félix Varela ed.). Habana.
19. Díaz, G. A. B. y Aguilar, U. R. (2011). *Sistema para la gestión de información en el área de investigación y postgrado de la Universidad de Granma*. Universidad de Granma, Bayamo.
20. Echevarría, Á. M. (1995). *Manual Práctico de HTML*. España: Escuela Técnica Superior de Ingenieros de Telecomunicación Universidad Politécnica de Madrid.
21. Echeverry, C. E. M., Trujillo, M. L. y Rueda, A. G. (2010). Interactive web Simulator For Financial Decision Making In Micro And Small Enterprises. *Scientia et Technica*, Año XVI(No 45).
22. Enrique, D. (2009). *Aplicación Web para el Control de Usuarios e Información del Laboratorio en la SUM de Cultura Física en Consolación del Sur*. Universidad de Pinar del Río Pinar del Río.
23. Fernández, I. G., Figueroa, R. C. y Pérez, D. F. (2011). Propuesta de arquitectura de una herramienta web para la administración del gestor PostgreSQL. [artículo]. *Revista Cubana de Ciencias Informáticas (RCCI)*.
24. Freitas, V. D., Vilorio, O. y Alvarez, G. (2001). Factores de interacción de la tecnología y la organización que inciden en la adopción de Ingeniería de Software. *Revista Venezolana de Gerencia (RVG)*, 6(14), 286-295.
25. Fuster, G. G. (2011). *Entrelazamiento de los aspectos estático y dinámico en las asociaciones UML* Universidad Carlos III de Madrid, España.
26. García, P. (2011). *Desarrollo de un sistema de consultas web para sistemas SAT*. Universidad de Complutense de Madrid, España.
27. Garrett, J. (2005). *Ajax: A New Approach to Web Applications*.
28. González, E. V. (2010). *La Seguridad de las Bases de Datos en la Red*. Universidad Veracruzana.
29. González, J. A. M. (1995). Los sistemas de información documental: consideraciones sobre sus características, conceptos y funciones. *Boletín Millares Carlos*, 14 137-150.
30. Haro, J. M. G. (2010). *Diseño e implementación de un marco de trabajo (framework) de presentación para aplicaciones JEE*. Universidad Oberta de Catalunya
31. Hernández, A. (2004). Identificación de procesos de negocio. *Ingeniería Industrial*, 25(3).
32. Hernández, A. (2005). Requisitos a partir del modelo del negocio. *Informática Industrial*, 26(3), 60-64.
33. Inciarte, A., Marcano, N. y Reyes, M. (2006). Gestión académico-administrativa en la educación básica. *Venezolana de Gerencia*. Maracaibo, 11(34).

34. Jacobson, I., Booch, G. y Rumbaugh, J. (2000a). *El lenguaje unificado de modelado*.
35. Jacobson, I., Booch, G. y Rumbaugh, J. (2000b). *El proceso unificado de desarrollo de software*. New York Addison Wesley.
36. Jiménez, D. (2009). *Aplicación Web para el Control de Usuarios e Información del Laboratorio en la SUM de Cultura Física en Consolación del Sur*. Unpublished Ingeniero Informático, Universidad de Pinar del Río, Cuba.
37. Jiménez, J. (2008). Ubuntu, Apache. Características, Revisado: 14 de Noviembre 2011, tomado de: [http://www.geocities.ws/jjaimes6603/fase-3/t4\\_ubuntu.html](http://www.geocities.ws/jjaimes6603/fase-3/t4_ubuntu.html)
38. Jurado, M. y Suárez, M. (2011). *Diseño e implementación de un sistema de facturación de agua potable para las parroquias rurales del Cantón Otavalo*. Universidad Técnica del Norte, Ibarra. Ecuador.
39. Mata, M. P. (2008). *Evaluación y pruebas de aplicaciones RIA con AJAX*. Unpublished Trabajo Final de Carrera, Universidad Politécnica de Catalunya.
40. Mejía, M. (2007). *Introducción al diseño de Base de Datos*. Universidad Nacional Autónoma de México, México, DF.
41. Mesa, E., Martínez, Y. y Argota, I. (2010). *Sistema para el análisis cuantitativo de los riesgos para los proyectos de producción de software*. Cuba: Universidad de las Ciencias informáticas.
42. Mesa, Y. R. (2006). De la gestión de información a la gestión del conocimiento. *ACIMED*, 14(1).
43. Moreno, M. (2008). *Proceso Unificado de Desarrollo para Aplicaciones con Arquitectura Orientada a Servicios*. México: Instituto Tecnológico de Mérida, Yucatán 97118.
44. Nobrega, M. (2005). Herramientas CASE: Rational Rose, Revisado: 14 de Diciembre 2011, tomado de: [http://curso\\_sin2.blogia.com/temas/herramientas-case.php](http://curso_sin2.blogia.com/temas/herramientas-case.php)
45. Noeda, M. S. (2009). *Tema 4. Bases de Datos (learningObject A)*: Universidad de Alicante.
46. Nogueira, R. y Góngora, N. (2000). *Evaluación de la Gestión Universitaria*. Buenos Aires: Comisión Nacional de Evaluación y Acreditación Universitaria (CONEAU).
47. Ortega, A., Morquecho, M. y Pilar, M. d. (2011). *Desarrollo de un portal Web para la gobernación del cañar*. Universidad Politécnica Salesiana. Ecuador.
48. Palladino, E. y Palladino, L. (1998). *Administración organizacional: calidad, capacitación, evaluación* (1ª ed.). Argentina: Buenos Aires Espacio, 1998.
49. Pérez, C. R. R. (2009). *Desarrollo de Software de Gestión para la Producción Bovina empleando Ajax*. Unpublished Tesis para optar al Título de: Ingeniero Civil en Informática, Universidad Austral de Chile, Chile.

50. Pérez, Y. (2008). Business Modelling and Requirement Management as Indispensable Stages of the Development of Automatic Information Systems. *Scielo*, 17(3).
51. Ponce, J., Ramos, I., Escalona, M. J., Mejías, M., Pérez, D., Aragón, G., et al. (2010). Pruebas de Aceptación en Sistemas Navegables. *Redalyc*, 6(3), 47-55.
52. Pressman, R. (2002). *Ingeniería del Software. Un enfoque práctico* (Tercera edición. Parte 1 ed.). Cuba: Editorial Félix Varela, Ciudad de la Habana
53. Prieto, F., Crespo, Y., García, F. y Laguna, M. (2000). *Construcción de frameworks basada en análisis de conceptos formales y soportada por Mecanos*.
54. Quintana, Y., Camejo, L. y Días, A. (2011). Diseño de la Base de Datos para Sistemas de Digitalización y Gestión de Medidas. *Revista de Informática Educativa y Medios Audiovisuales*, 8(15), 17-25.
55. Riba, J. M. C. (2006). *Bases de datos distribuidas con una solución LAMP (Linux, Apache, MYSQL, PHP)*. Univesidad Autónoma del Estado de Hidalgo. México.
56. Rojas, M. (2009). Diseño de un Sistema de Control de Gestión para la Facultad de Ingeniería, Ciecias y Administración de la Universidad de la Frontera, Revisado: 10 de Noviembre 2011, tomado de: [http://ceur-ws.org/Vol-558/Art\\_14.pdf](http://ceur-ws.org/Vol-558/Art_14.pdf)
57. Romero, M. B. A. (2000). A retrospective view of CASE tools adoption. *ACM SIGSOFT Software Engineering*, 25(2), 46-50.
58. Romero, R. (2009). Nivel 1 del mapa de navegación, Revisado: 16 de febrero 2012, tomado de: <http://blog.juventudrebelde.cu/sitemap/nivel-1-del-mapa-de-navegacion/>
59. Rosabal, A. (2011). Sistema para el control de actividades científicas. *Cuadernos de Educación y Desarrollo*, 3(27).
60. Sada, L. (2011). Plan de Trabajo, Revisado: 7 de Noviembre 2011, tomado de: <http://www.scn.org/mpfc/key/key-ps.htm>
61. Schmitt, C. (2007). *Curso de CSS*. España.
62. Sordo, V. y Suárez, A. (2006). *Sistema de Gestión Administrativa. Facultad 4*. Unpublished Trabajo de Diploma para optar por el título de Ingeniero Informático, Universidad de Ciencias Informáticas, La Habana.
63. Suárez, C. (2009). *Desarrollo de software orientado a objetos*. Perú: Universidad Nacional del santa.
64. Tamayo, S. C., Campaña, M. P. y Expósito, F. R. (2007). Alternativa para el proceso de planificación de horarios docentes de una Universidad. *Revista Trimestral. Holguín*, 13(4).
65. Townsend, W. (2010). An Application Of The Rational Unified Process® For Requirements Analysis. *Journal Of Business Case Studies*, 6(1), 19- 26.

66. Trilles, J. (2011). Reglas de Negocio (BR) y Gestión por Procesos de Negocio (BPM), Revisado: 6 de Enero 2012, tomado de: <http://www.club-bpm.com/Noticias/art0007.htm>
67. Valdelli, I. (2006). Aspectos y características generales de Javascript, Revisado: 12 de Diciembre 2011, tomado de: <http://www.htmlpoint.com/cd-javascript/corso/index.html>
68. Valdés, D. P. (2007). Los diferentes lenguajes de programación para la web, Revisado: 16 de Noviembre 2011, tomado de: <http://www.maestros-del-web.com/principiantes/los-diferentes-lenguajes-de-programacion-para-la-web/>

## REFERENCIAS BIBLIOGRÁFICAS

1. Acosta.(2009).*Multimedia” La FEU de Oriente, tan joven como entonces sigue sumando”*.Universidad de Oriente.
2. Arias, M. (2005). *La ingeniería de requerimientos y su importancia en el desarrollo de proyectos de software*.*InterSedes*, 6(10).
3. Bou, G. (2003).*El guión multimedia*. ED. Anaya Multimedia, España.
4. Cantón, N. , JOSÉ C y SILVA LEÓN, ARNALDO.*Historia de Cuba(1959-1999)*.La Habana : Pueblo y Educación, 2009.[3]
5. Chaveco, J. R. y Rojas, J. R.(2004). *Multimedia y Software Educativo*. Managua : UPONIC.
6. Date (2003). *Introducción a los Sistemas de Base de Datos* (Félix Varela ed.). Habana.
7. De la Cruz, O.(2007). *Multimedia Educativa para la Enseñanza y el Aprendizaje de la Programación (SIMAP)*. Boletín de la Sociedad Cubana de Matemática y Computación, Volumen 5, Número Especial. 2007. ISSN 1728-6042.
8. Hernández, A. (2005). *Requisitos a partir del modelo del negocio*.*Informática Industrial*, 26(3), 60-64.
9. Herold G, (2002).*Folleto Gestión de Bases de Datos en la Ingeniería de Sistemas*. Managua.
- 10.Jacobson, I., Booch, G. y Rumbaugh, J. (2000a). *El lenguaje unificado de modelado*.
- 11.Jacobson, I., Booch, G. y Rumbaugh, J. (2000b). *El proceso unificado de desarrollo de software*. New York Addison Wesley.
- 12.Jiménez, D. (2009). *Aplicación Web para el Control de Usuarios e Información del Laboratorio en la SUM de Cultura Física en Consolación del Sur*.Unpublished Ingeniero Informático, Universidad de Pinar del Río, Cuba.
- 13.Marqués P.(1995).*Software Educativo: guía de uso y metodología de diseño*. Barcelona. Estel.

14. Mejía, M. (2007). *Introducción al diseño de Base de Datos*. Universidad Nacional Autónoma de México, México, DF.
15. Pérez, Y. (2008). *Business Modelling and Requirement Management as Indispensable Stages of the Development of Automatic Information Systems*. *Scielo*, 17(3).
16. Ponce, J., Ramos, I., Escalona, M. J., Mejías, M., Pérez, D., Aragón, G., et al. (2010). *Pruebas de Aceptación en Sistemas Navegables*. *Redalyc*, 6(3), 47-55.
17. Rojas C., JOSÉ R. (2008). **SIMAP**: *Propuesta de un mediador didáctico como parte de la dinámica del Proceso de Enseñanza – Aprendizaje de la Programación*. Tesis en opción al Título de Máster. Universidad de Oriente.
18. Romero, R. (2009). *Nivel 1 del mapa de navegación, Revisado: 16 de febrero 2012, tomado de: <http://blog.juventudrebelde.cu/sitemap/nivel-1-del-mapa-de-navegacion/>*
19. Sánchez, J. (1999). *Construyendo y Aprendiendo con el Computador*.
20. Valdés Pardo, V.G. et al. (2002) *Aplicaciones de las NTIC a la Educación Superior: visión y acción en la UCLV*. Monografía en formato electrónico. Editorial Feijoó, Universidad Central de Las Villas. Cuba. ISBN: 959-250-048-7.