



**INSTITUTO SUPERIOR POLITÉCNICO  
"José Antonio Echeverría"**

**Facultad de Ingeniería Industrial**

**Ingeniería Informática**

**Trabajo para optar por el título de Ingeniero Informático**

**Herramienta para la confección  
de aplicaciones multimedia educativa para niños.  
(Nanatool)**

**Autor:** Renier La Rosa González

**Tutor:** Ing. Yaillet Martínez Pérez

**Ciudad de la Habana, Junio del 2005**

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al Grupo de Gestión Universitaria del Centro de Estudios de Ingeniería de Sistemas (CEIS); así como a dicho centro para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los    días del mes de            del 2005.

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

\_\_\_\_\_

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

\_\_\_\_\_

**Opinión del usuario del trabajo de diploma.**

El Trabajo de Diploma, titulado: “Herramienta para la confección de aplicaciones multimedia educativa para niños”, fue realizado en la UCI. Esta entidad considera que, en correspondencia con los objetivos trazados, el trabajo realizado le satisface

- Totalmente
- Parcialmente en un \_\_\_\_\_ %

Los resultados de este Trabajo de Diploma le reportan a esta entidad los beneficios siguientes:

---

---

---

---

---

---

---

---

Como resultado de la implantación de este trabajo se reporta un efecto económico que asciende a :

Y para que así conste, se firma la presente a los \_\_\_\_\_ días del mes de Junio del año 2005.

\_\_\_\_\_  
Representante de la entidad

\_\_\_\_\_  
Cargo

\_\_\_\_\_  
Firma

Cuño

## **Opinión del tutor del trabajo de diploma**

Título: "Herramienta para la confección de aplicaciones multimedia educativa para niños".

Autor: Renier La Rosa González

El tutor del presente Trabajo de Diploma considera que durante su ejecución el estudiante mostró las cualidades que a continuación se detallan.

Por todo lo anteriormente expresado considero que el estudiante está apto para ejercer como Ingeniero Informático; y propongo que se le otorgue al Trabajo de Diploma la calificación de

\_\_\_\_\_  
Firma

\_\_\_\_\_  
Fecha

## Resumen

Las aplicaciones multimedia están cobrando cada vez más auge dentro del proceso educativo, esto unido al continuo desarrollo de las TIC han contribuido a un notable incremento de la producción de software educativos. La Universidad de las Ciencias Informáticas, desde su surgimiento, ha marchado a la vanguardia del movimiento de sistemas multimedia para la educación.

Este centro recibe continuamente solicitudes de varias empresas e instituciones para la elaboración de este tipo de proyectos, principalmente orientados a niños de edad comprendida entre 4 y 7 años. Dado el cúmulo de solicitudes y la necesidad de darle una respuesta satisfactoria en la mayor brevedad posible es que se hace imprescindible la implantación de una herramienta capaz de realizar la confección de productos multimedia educativa para niños con eficiencia y calidad.

Conocer la manera en que se lleva a cabo el proceso de confección y control de los productos multimedia en la Dirección de Software Educativos de la UCI y realizar el diseño y modelado de la herramienta "Nanatool", son los objetivos fundamentales del presente trabajo de diploma. En este informe se recoge la fundamentación teórica para el desarrollo de esta herramienta, así como la documentación asociada a su diseño.

Como resultado del estudio realizado en el marco del modelo de la aplicación llamada "Nanatool", se obtuvo una caracterización de las investigaciones actuales, así como una evaluación de las plataformas y herramientas de desarrollo más utilizadas en la actualidad para este tipo de aplicaciones.

# Índice

<b>INTRODUCCIÓN.....</b>	<b>1</b>
<b>CAPÍTULO 1 FUNDAMENTACIÓN DEL TEMA.....</b>	<b>5</b>
<b>1.1 INTRODUCCIÓN.....</b>	<b>5</b>
<b>1.2 PRINCIPALES CONCEPTOS ASOCIADOS AL DOMINIO DEL PROBLEMA.....</b>	<b>5</b>
1.2.1 <i>Hipertexto.....</i>	5
1.2.2 <i>Multimedia.....</i>	6
1.2.3 <i>Soporte para Multimedia.....</i>	7
1.2.4 <i>Hipermedia.....</i>	9
1.2.5 <i>Guión.....</i>	9
1.2.6 <i>Producto.....</i>	10
<b>1.3 OBJETO DE ESTUDIO Y CAMPO DE ACCIÓN.....</b>	<b>10</b>
1.3.1 <i>Descripción actual de los procesos del negocio.....</i>	11
1.3.2 <i>Situación problémica.....</i>	12
<b>1.4 ANÁLISIS COMPARATIVO DE OTRAS SOLUCIONES EXISTENTES CON LA PROPUESTA.....</b>	<b>13</b>
1.4.1 <i>Herramientas de desarrollo de Multimedia.....</i>	13
1.4.2 <i>Comparación con la propuesta.....</i>	15
<b>1.5 OBJETIVOS GENERALES Y ESPECÍFICOS.....</b>	<b>16</b>
<b>1.6 DESCRIPCIÓN DE LAS TENDENCIAS Y TECNOLOGÍAS ACTUALES SOBRE LAS QUE SE APOYA LA PROPUESTA.....</b>	<b>17</b>
<b>1.7 ¿QUÉ ES UN PROCESO DE DESARROLLO DE SOFTWARE?.....</b>	<b>24</b>
1.7.1 <i>El Proceso Unificado de Desarrollo de software (RUP).....</i>	25
<b>1.8 LENGUAJES Y GESTORES DE BASES DE DATOS UTILIZADOS.....</b>	<b>27</b>
1.8.1 <i>C#.....</i>	27
1.8.2 <i>Director y Lingo.....</i>	29
1.8.3 <i>Herramienta de modelado de Datos (ER/Studio).....</i>	30
1.8.4 <i>Gestores de Base de Datos.....</i>	31
<b>1.9 CONCLUSIONES.....</b>	<b>32</b>
<b>CAPÍTULO 2 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....</b>	<b>33</b>
<b>2.1 INTRODUCCIÓN.....</b>	<b>33</b>
<b>2.2 REGLAS DEL NEGOCIO A CONSIDERAR.....</b>	<b>33</b>
<b>2.3 DESCRIPCIÓN DE LOS PROCESOS DEL NEGOCIO PROPUESTOS.....</b>	<b>34</b>
2.3.1 <i>Descripción de los Actores y Trabajadores del negocio.....</i>	35
2.3.2 <i>Diagrama de casos de uso del negocio.....</i>	36
2.3.3 <i>Diagrama de Actividades.....</i>	37

<b>2.4 DIAGRAMA DE CLASES DEL MODELO DE OBJETOS.....</b>	<b>38</b>
<b>2.5 REQUISITOS FUNCIONALES.....</b>	<b>39</b>
<b>2.6 REQUISITOS NO FUNCIONALES.....</b>	<b>40</b>
<b>2.7 DESCRIPCIÓN DEL SISTEMA PROPUESTO.....</b>	<b>43</b>
<b>2.7.1 Concepción general del sistema.....</b>	<b>43</b>
<b>2.7.2 Modelo de casos de uso del sistema.....</b>	<b>45</b>
<b>2.7.3 Descripción de los actores del sistema.....</b>	<b>46</b>
<b>2.7.4 Paquete Administración.....</b>	<b>46</b>
<b>2.7.5 Paquete Diseño.....</b>	<b>47</b>
<b>2.7.6 Paquete Usuario.....</b>	<b>48</b>
<b>2.7.7 Descripción de los Casos de uso.....</b>	<b>49</b>
<b>2.8 CONCLUSIONES.....</b>	<b>57</b>
<b>CAPITULO 3 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA .....</b>	<b>58</b>
<b>3.1 INTRODUCCIÓN.....</b>	<b>58</b>
<b>3.2 DIAGRAMA DE CLASES.....</b>	<b>58</b>
<b>3.2.1 Paquete Autenticación.....</b>	<b>59</b>
<b>3.2.2 Paquete Plantillas.....</b>	<b>60</b>
<b>3.2.3 Paquete Pantallas.....</b>	<b>61</b>
<b>3.2.4 Paquete Producto.....</b>	<b>62</b>
<b>3.3 DISEÑO DE LA BASE DE DATOS.....</b>	<b>63</b>
<b>3.3.1 Diagrama de Clases Persistentes Plantillas.....</b>	<b>63</b>
<b>3.3.2 Diagrama de Clases Persistentes Pantallas.....</b>	<b>64</b>
<b>3.3.3 Modelo de Datos Plantillas.....</b>	<b>65</b>
<b>3.3.4 Modelo de Datos Pantalla.....</b>	<b>66</b>
<b>3.4 ESTÁNDARES DE CODIFICACIÓN.....</b>	<b>67</b>
<b>3.5 CONCLUSIONES.....</b>	<b>67</b>
<b>CAPÍTULO 4 ESTUDIO DE FACTIBILIDAD .....</b>	<b>68</b>
<b>4.1 INTRODUCCIÓN.....</b>	<b>68</b>
<b>4.2 PLANIFICACIÓN.....</b>	<b>68</b>
<b>4.3 CÁLCULO DE LAS INSTRUCCIONES FUENTES, ESFUERZO, TIEMPO DE DESARROLLO, CANTIDAD DE HOMBRES Y COSTO.....</b>	<b>71</b>
<b>4.4 BENEFICIOS TANGIBLES E INTANGIBLES.....</b>	<b>73</b>
<b>4.5 ANÁLISIS DE COSTOS Y BENEFICIOS.....</b>	<b>73</b>
<b>4.6 CONCLUSIONES.....</b>	<b>73</b>
<b>CONCLUSIONES.....</b>	<b>74</b>

<b>RECOMENDACIONES.....</b>	<b>75</b>
<b>BIBLIOGRAFÍA .....</b>	<b>76</b>
<b>GLOSARIO DE TÉRMINOS .....</b>	<b>78</b>

## Introducción

“Guardería” es un proyecto que consiste en la implementación de un conjunto de productos de software multimedia con varios cuentos infantiles para niños en la edad comprendida entre 4 y 7 años.

La Dirección de Software Educativo de la UCI persigue incrementar los servicios que se ofrecen a los niños que visiten nuestro país, añadir valor a las guarderías con el uso de las nuevas tecnologías de la información además de promocionar y comercializar el software educativo desarrollado en Cuba, mostrando de esa manera los logros de la Industria Cubana del Software.

Se ha previsto la ubicación de puestos de trabajo en Guarderías, Instalaciones turísticas, aeroportuarias y otras, incluyendo: mobiliario, diseño del local y equipos de cómputo especialmente diseñados para niños, con software preinstalado y de fácil manejo para permitir trabajar de forma independiente y autodidacta.

La dificultad fundamental que presenta el proceso de desarrollo de software para niños de corta edad es que se pierde o malgasta mucho tiempo en desarrollar estos productos que son relativamente sencillos y fáciles de resolver. Las herramientas que se encuentran actualmente en el centro son demasiado generales y poco específicas para este tipo de trabajo.

El desarrollo de productos multimedia resulta ser un trabajo muy reiterativo, ya que continuamente se tienen que programar métodos que guardan una gran similitud entre los proyectos de esta rama, la mayoría de éstos se pueden reutilizar de un trabajo a otro. Esto trae consigo que se invierta mucho tiempo en repetir actividades antes descritas y elaboradas. Como consecuencia, el tiempo de confección de un producto es muy grande con respecto a su complejidad. La elaboración de los productos requiere de personal altamente capacitado en técnicas y lenguajes de programación y se malgastan recursos y elementos haciendo más costoso el proceso de fabricación. El

producto final carece de flexibilidad y la creación de otro producto similar requiere la reprogramación íntegra del mismo.

Esta situación nos lleva a la necesidad de desarrollar una herramienta capaz de generar productos seriados sin la necesidad de reprogramarlos completamente brindando así una respuesta óptima al proceso de confección de productos multimedia para niños. Este es precisamente el objetivo que se persigue con la implantación de la herramienta llamada "Nanatool".

Se espera, como aportes prácticos, dotar al centro de ésta aplicación informática para concluir los proyectos de multimedia educativa en tiempo y con la calidad requerida.

En este trabajo fue objeto de estudio fundamentalmente la gestión de planificación y control del trabajo multimedia en la Universidad de las Ciencias Informáticas (UCI). La cual comprende tres etapas o niveles jerárquicos: Planificación, Montaje y Revisión, cada uno con sus correspondientes grados de consolidación y detalle de la información.

En la Planificación se escoge una facultad en dependencia de la cantidad de proyectos que ya tenga asignados, se selecciona un Líder de proyecto que se encarga de gestionar y controlar el proceso de confección. La Dirección de Diseño se encarga de tratar la información correspondiente a la estructura visual del producto y la Dirección de Comunicación Audiovisual gestiona los elementos de media.

En la etapa de Montaje se implementa el producto que se corrige más tarde en el nivel de Revisión.

El objetivo principal de este trabajo es: Conocer la manera en que se lleva a cabo el proceso de gestión y producción de software educativo en la Universidad de las Ciencias Informáticas y diseñar una herramienta para la confección de los productos multimedia desarrollados en la entidad vinculados al proyecto "Guardería".

Entre los objetivos específicos se encuentran:

1. Realizar un estudio detallado del proceso de gestión de software educativo en el centro.
2. Realizar un estudio de las tendencias y tecnologías actuales para presentar una base sólida en la solución de la tarea que nos atañe.
3. Diseñar una base de datos capaz de almacenar de manera eficiente la información correspondiente a los productos y elementos multimedia.
4. Diseñar una base de datos que contenga la información inherente a las plantillas que sirven de guía a las pantallas del producto final y a los usuarios que las manipulan.
5. Diseñar y modelar una herramienta que permita disminuir el tiempo máximo de elaboración de cada proyecto de multimedia educativa para niños, así como hacer más ágil y amena la realización de los productos ofreciendo un ambiente de trabajo más personalizado y cómodo.

Para dar cumplimiento a los objetivos descritos se desarrollaron las siguientes tareas:

- Realizar un estudio de los problemas que presenta la elaboración de productos multimedia en la Universidad de las Ciencias Informáticas.
- Comunicar las mejoras propuestas a los encargados del control de los productos.
- Determinar requerimientos funcionales y no funcionales que debe cumplir el sistema para su óptimo funcionamiento.
- Estudiar las tendencias y tecnologías, así como las metodologías y software usados en la actualidad para lograr una correcta selección de los mismos
- Determinar los procesos del negocio a automatizar.
- Ejecutar el modelo del negocio.
- Realizar el modelo de análisis y diseño.
- Diseñar las bases de datos necesarias para la aplicación.

En la ejecución de este trabajo se examinó abundante bibliografía como el libro “El Proceso Unificado de Desarrollo de Software” de los autores Ivar Jacobson, Grady Booch y James Rumbaugh. Se consultaron diversos sitios de Internet y documentación

actualizada sobre Ingeniería de Software disponible en la red, esta documentación fue desarrollada fundamentalmente por profesores de la universidad.

El presente documento se divide en varios capítulos:

El Capítulo 1 “*Fundamentación del tema*” aborda las cuestiones relacionadas con los problemas que dieron pie a la concepción de la herramienta, los principales conceptos asociados con la multimedia y el soporte para la misma. Se describe además los lenguajes de programación y sistemas para realizar el análisis y diseño de la aplicación así como las tendencias y tecnologías actuales sobre las que se basa.

En el Capítulo 2 “*Descripción de la solución propuesta*” se da a conocer la modelación del negocio con sus actores y trabajadores del negocio, los casos de uso del negocio, el diagrama de actividades correspondiente, los diagramas de clases de objetos, los requerimientos del sistema y los casos de uso del sistema. Además se ofrece una descripción del mismo. En este capítulo se realiza todo el diseño del sistema.

El tercer Capítulo “*Construcción de la solución propuesta*”, como su nombre lo indica, está dedicado principalmente a la construcción del sistema a desarrollar. El mismo contiene el diagrama de clases del sistema representado por los paquetes correspondientes. Se puntualiza además el diseño de las bases de datos apoyados en el modelo de datos generado a partir del diagrama de clases persistentes. Quedan descritos en el capítulo los estándares de codificación que se usaron durante el diseño del sistema.

Este trabajo incluye además las conclusiones, recomendaciones y referencias a la bibliografía utilizada en su elaboración.

# **Capítulo 1 Fundamentación del tema.**

## ***1.1 Introducción***

Los objetivos de este capítulo son identificar los problemas que dieron pie a la concepción de una herramienta capaz de agilizar el trabajo con multimedia, descubrir el problema en sí, describir los principales conceptos asociados con la multimedia y el soporte para la misma. Se presenta una breve comparación de otras soluciones existentes con la propuesta planteada. Se abordan temas relacionados con el uso de aplicaciones Windows, lenguajes de programación, y sobre los sistemas para realizar el análisis y diseño de la aplicación. Además se ofrece una descripción de las tendencias y tecnologías actuales sobre las que se basa la propuesta.

## ***1.2 Principales conceptos asociados al dominio del problema.***

A continuación se describen los principales conceptos asociados al dominio del problema, los cuales son de vital importancia para su entendimiento. Es oportuno destacar las diferencias entre hipertexto, multimedia e hipermedia ya que hoy en día sus conceptos se confunden e identifican entre sí.

### **1.2.1 Hipertexto**

“El hipertexto no es más que una nueva forma de organizar la información contenida en los textos.”[2]

La utilización de nuevas técnicas con las computadoras permitió el desarrollo del hipertexto, una manera de ligar temas con palabras en los textos, de modo que se pueda acceder a contenidos de interés específico en uno o varios documentos sin tener que leerlos completamente. El hipertexto utiliza “palabras calientes” contenidas en el cuerpo del documento para acceder a otros textos relacionados con dicha palabra. Así se controla el orden de lectura y la aparición de los datos en la pantalla de una manera más parecida a nuestro modo de relacionar pensamientos, en el que el cerebro va respondiendo por libre asociación de ideas, y no siguiendo un hilo único y lineal. [1]

Pero si algo caracteriza al hipertexto es el dinamismo y la interactividad que conlleva frente a lo estático del texto impreso. El hipertexto se presenta como un medio de organización más dinámico, ya que la información puede cambiar de lugar, cambiar su presentación, estructurarla en varios niveles, y sobre todo ofrecer diversos caminos de lectura en función del interés del usuario. Además es interactivo, en cuanto que permite dialogar con el texto superando las limitaciones del papel. Así, las posibilidades de comunicación entre usuarios y entre usuario y ordenador aumentan considerablemente [2].

### **1.2.2 Multimedia**

Todo el mundo tiene su propia impresión de lo que es “Multimedia”. Algunos piensan en ello como en un conjunto de muestras de música, otros lo ven como un quiosco interactivo de venta al por menor, otros todavía creen que es un video juego. Con el aumento de las tecnologías de la información, “Multimedia” puede parecer todo esto.

No hay una definición correcta o equivocada, Multimedia es una continuidad de aplicaciones y tecnologías que permite un amplio rango de experiencias. [4]

Multimedia es todo aquello que utiliza, conjunta y simultáneamente, diversos medios de comunicación en la presentación de la información, como imágenes, animación, vídeos, sonido y texto. Básicamente, multimedia es la cualidad de un sistema o documento que utiliza más de un medio de comunicación al mismo tiempo [1].

Aunque este concepto es tan antiguo como la comunicación humana, ya que al expresarnos en una charla normal hablamos (sonido), escribimos (texto), observamos a nuestro interlocutor (video) y accionamos con gestos y movimientos de las manos (animación), apenas ahora, con el auge de las aplicaciones multimedia para computadora, este vocablo entró a formar parte del lenguaje cotidiano [1].

Hoy en día el significado más habitual del concepto multimedia incluye dos características esenciales: por un lado, la integración de diferentes medios o lenguajes en un mismo documento y, por otro, la interactividad. La multimedia es una combinación de informaciones de naturaleza diversa coordinada por un equipo

computerizado y con la que el usuario puede interactuar, creando un entorno de comunicación activo y participativo.

En las Multimedia Educativas, los alumnos pueden interactuar con el sistema y recibir información simultánea sobre el tema estudiado de manera mas ilustrativa y amena, combinando con los diferentes sentidos para la mayor interpretación y entendimiento del contenido, ampliando las posibilidades de retención de la información y entendimiento del entorno a estudiar.

Cuando un programa de computadora, un documento o una presentación combina adecuadamente los medios, se mejora notablemente la atención, la comprensión y el aprendizaje, ya que se acercará algo más a la manera habitual en que los seres humanos nos comunicamos, cuando empleamos varios sentidos para comprender un mismo objeto o concepto.[1]

Las aplicaciones Multimedia suelen necesitar más memoria y capacidad de proceso que la misma información representada exclusivamente en forma de texto. Por ejemplo: una computadora que ejecute aplicaciones Multimedia tiene que tener una CPU rápida. Un ordenador Multimedia también necesita memoria adicional para ayudar a la CPU a efectuar cálculos y permitir la representación de imágenes complejas en la pantalla. El ordenador también necesita un disco duro de alta capacidad para almacenar y recuperar información Multimedia, así como una unidad de disco compacto para ejecutar aplicaciones almacenadas en CD-ROM. [3]

### **1.2.3 Soporte para Multimedia**

Cuando utilizamos actualmente el término multimedia nos estamos refiriendo a equipos informáticos con la capacidad, al menos, de reproducir imágenes y sonidos. De tal modo que un *sistema multimedia* es un entorno constituido por *hardware* (ordenadores u otros aparatos con los equipamientos necesarios para reproducir, crear y/o registrar imágenes y sonidos) y *software* (programas o aplicaciones que permiten controlar la reproducción, creación y/o registro de imágenes y sonidos).

El software de carácter multimedia se almacena en soportes de gran capacidad como el CD-ROM o el DVD. En conclusión, un sistema multimedia ha de ser capaz de mostrar, producir y/o almacenar información textual, sonora y audiovisual de un modo integrado. [3]

- **CD**

El disco compacto (conocido popularmente como CD, del inglés *compact disc*) es un soporte digital óptico utilizado para almacenar cualquier tipo de información (audio, video, documentos, etc.). Hoy en día tecnologías como el DVD pueden desplazar o minimizar esta forma de almacenamiento, aunque su uso sigue vigente.

Con la llegada del *CD* se pudo almacenar digitalmente una gran cantidad de datos en un solo soporte, extraíble, de rápido acceso, larga vida útil, de poco peso y fácil de transportar. [6]

La capacidad de estos *CD* va desde los *650 MB* y *74 min*, hasta los *1054 MB* y *120 min*. Se debe tener en cuenta que hoy los *CD* tienen una vida útil limitada debido a la degradación de su capa fotosensible, aunque está situada en una media de 30 años. [6]

- **DVD**

**DVD** acrónimo inglés de *Digital Versatile Disc* (Disco Versátil Digital), es un soporte para el almacenamiento de datos binarios de igual funcionamiento y tamaño que el CD-ROM, aunque con pistas más finas, lo cual aumenta la densidad de la información gravable en la superficie y por tanto le da una mayor capacidad de almacenamiento que el CD-ROM.[1]

Los DVD pueden tener distintas capacidades en dependencia del fabricante y tecnología usada para su creación, estos soportes pueden almacenar desde *4.7* hasta *17 GB*. Estas capacidades se incrementan continuamente ya que la producción y desarrollo de los DVD están en constante aumento.

“Podemos decir sin temor a equivocarnos, que la tecnología DVD es la evolución lógica del desarrollo del CD.” [6]

#### **1.2.4 Hipermedia**

Se puede interactuar con sonidos, animaciones y servicios de Internet relacionados con el tema que se está tratando, lo cual ha dado origen a un nuevo concepto: Hipermedia, resultado de la fusión de los conceptos hipertexto y multimedia.

Actualmente estos términos se confunden e identifican entre sí, de tal forma que al nombrar uno de los conceptos anteriores (hipermedia, hipertexto o multimedia) de forma instintiva y casi automática se piensa en los otros dos.

Un documento hipermedia es siempre una multimedia, pero no al revés. Podemos tener un documento multimedia pero que nos presente la información de forma lineal, secuenciada, sin que tengamos la posibilidad de usar interconexiones para movernos y localizar la información por el documento. [5]

#### **1.2.5 Guión**

El guión no es más que un documento elaborado por uno o varios expertos en las materias a desarrollar, este documento será la guía fundamental tanto para el equipo de diseño como para el de programación, para el logro de los objetivos a través de sus pautas o especificaciones.

- Guión de contenido

El guión de contenido es desarrollado por un pedagogo especialista en la educación de niños pequeños y presenta el contenido que debe tener el producto que se le muestra al niño, asegurándose que cumple con las reglas pedagógicas para niños de esa edad.

- Guión multimedia

El guión multimedia se apoya en el guión de contenido y expresa la misma información pero en forma de multimedia, o sea, combinando textos, animaciones,

imágenes y sonido. Contiene además todas las especificaciones y pautas sobre el producto.

### **1.2.6 Producto**

Los productos son aplicaciones multimedia educativas.

### **1.3 Objeto de estudio y campo de acción.**

La gestión de planificación y control del trabajo multimedia en la Universidad de las Ciencias Informáticas (UCI) comprende tres etapas o niveles jerárquicos: Planificación, Montaje y Revisión, cada uno con sus correspondientes grados de consolidación y detalle de la información.

#### **➤ Planificación**

En este nivel se analiza las facultades que están destinadas para trabajar en el desarrollo de multimedia, se escoge una en dependencia de la cantidad de proyectos que ya tenga asignados. De esta manera se analiza los requerimientos de recursos humanos (cantidad de personas que se necesitan, y se escoge el(los) líder(es) del proyecto.

En este proceso el Líder del proyecto realiza un cronograma donde calcula el tiempo que durará el proyecto y las tareas que se le asignará a cada desarrollador.

Se realiza la gestión de los contenidos de la multimedia y los guiones, la recolección y preparación de materiales así como el estudio del caso específico del tema a tratar. Esta tarea es realizada por parte de los especialistas en contenido.

Dentro de la planificación se analizan los detalles del diseño de la multimedia, es decir, todo lo referente a la estructura visual que tendrá el producto. Se planifica la forma en que se realizará la navegación en la multimedia (en caso de que sea interactiva), la forma en que aparecerá el contenido, los colores que se usarán, el tipo de letra, etc. De esto se encarga la Dirección de Diseño. Por otro lado se realiza la

gestión de las medias, o sea, la preparación de las fotos, los videos y todo lo necesario para el montaje de la multimedia. De todo esto se encarga la Dirección de Comunicación audiovisual.

➤ **Montaje**

En esta etapa se recopila todos los datos y elementos del nivel anterior y se pasa a implementar el producto, donde intervienen los programadores de las facultades seleccionadas.

➤ **Revisión**

En este nivel se realiza una revisión y corrección de la multimedia por parte de los especialistas y diseñadores. Esta etapa puede estar solapada con la de Montaje donde en ocasiones a medida que se lleva a cabo el montaje, se hacen revisiones para ir verificando que todo marche bien.

### **1.3.1 Descripción actual de los procesos del negocio**

En la Dirección de Software Educativo de la UCI se desarrollan productos para niños comprendidos en la de edad de 4 a 7 años. En la elaboración de dichos productos intervienen varios equipos de desarrollo que unen sus esfuerzos para lograr la tarea planteada.

El negocio comienza cuando el Líder de proyecto le presenta la propuesta de realización del proyecto al especialista en la educación de niños (pedagogo), éste a su vez elabora un *guión de contenido* que presenta o describe el enfoque pedagógico de los cuentos infantiles. A partir de este documento el guionista de multimedia confecciona el *guión multimedia* que contiene la descripción de los elementos audiovisuales necesarios para abarcar y desarrollar la propuesta del guión

de contenido. Los guiones mencionados anteriormente le sirven de referencia y base a los diseñadores, especialistas audiovisuales y programadores del producto.

En el proceso intervienen los especialistas audiovisuales y diseñadores de interfaz. Los primeros son los encargados de gestionar, recolectar, y crear los elementos audiovisuales del producto, entiéndase fotos, sonidos, videos, etc. Los diseñadores de interfaz confeccionan los componentes de la interfaz como son: botones, animaciones, imágenes, sonidos digitales, etc. Además son los encargados de elaborar el prototipo de interfaz. Este prototipo sirve de guía gráfica a los programadores para que tengan una primera vista de las pantallas del producto. Una vez terminada la tarea de los diseñadores el equipo de programación se encarga de crear la interacción entre los elementos de multimedia logrando así dar forma al producto final que es presentado al Líder del proyecto.

### **1.3.2 Situación problemática.**

La Dirección de Software Educativo de la UCI persigue incrementar los servicios que se ofrecen a los niños que visiten nuestro país, añadir valor a las guarderías con el uso de las nuevas tecnologías de la información además de promocionar y comercializar el software educativo desarrollado en Cuba, mostrando de esa manera los logros de la Industria Cubana del Software.

Se ha previsto la ubicación de puestos de trabajo en Guarderías, Instalaciones turísticas, aeroportuarias y otras, incluyendo mobiliario, diseño del local y equipos de cómputo especialmente diseñados para niños, con software preinstalado y de fácil manejo para permitir trabajar de forma independiente y autodidacta.

La dificultad fundamental que presenta el proceso de desarrollo de software para niños de corta edad es que se pierde o malgasta mucho tiempo en desarrollar éstos productos que son relativamente sencillos y fáciles de resolver. Las herramientas que se encuentran actualmente en la instalación son demasiado generales y poco específicas para este tipo de trabajo.

El desarrollo de estos productos resulta ser un trabajo muy reiterativo, ya que continuamente se tienen que programar métodos que guardan una gran similitud entre los proyectos de esta rama, la mayoría de éstos se pueden reutilizar de un trabajo a otro. Esto trae consigo que se invierta mucho tiempo en repetir actividades antes descritas y elaboradas. Como consecuencia, el tiempo de confección de un producto es muy grande con respecto a su complejidad. La elaboración de los productos requiere de personal altamente capacitado en técnicas y lenguajes de programación y se malgastan recursos y elementos haciendo más costoso el proceso de confección. El producto final carece de flexibilidad y la creación de otro producto similar requiere la reprogramación íntegra del mismo.

#### ***1.4 Análisis comparativo de otras soluciones existentes con la propuesta.***

En éste epígrafe se abordan algunas de las características fundamentales de diferentes herramientas de desarrollo multimedia y una breve comparación de éstas con la solución propuesta. Donde se explican las razones por las cuales se decidió considerar la propuesta planteada como medio de solución al problema existente en la instalación.

##### **1.4.1 Herramientas de desarrollo de Multimedia.**

Las herramientas de desarrollo multimedia suministran por lo general un entorno integrado [7] con:

- Creación, edición e importación de tipos específicos de datos.
- Secuencia de reproducción u hoja de señalizaciones.
- Lenguaje para responder a las acciones de usuario.

Se organizan en grupos, basándose en la presentación que utilizan para dar secuencia y organizar los elementos de multimedia [8]:

- Herramientas basadas en páginas.
- Herramientas basadas en iconos controlados por eventos.
- Herramientas basadas en tiempo y de presentación.

### **1) Herramientas basadas páginas:**

Los sistemas de desarrollo basado en tarjetas y páginas proporcionan una presentación sencilla y fácil de entender para organizar los elementos de multimedia [8]. En estos sistemas de desarrollo los elementos se organizan como páginas de un libro. El sistema de desarrollo permite ligar éstas páginas en secuencias organizadas pudiendo saltar de una a otra.

Las rutinas de navegación se convierten en una simple instrucción para ir a una página que contiene las imágenes y texto adecuado, los sonidos asociados, animaciones y secuencia de vídeo.

Los sistemas de desarrollo basados en páginas están orientados a objeto: los objetos son botones, campo de texto, objetos gráficos, fondo, páginas y aún el proyecto mismo. Las características de los objetos se definen con propiedades (resaltado, negritas, rojos, escondidos, activo, bloqueado, etc.)[8].

Ejemplo típico: ToolBook.

### **2) Herramientas basadas en iconos:**

Estas herramientas disponen de una biblioteca de elementos multimedia. Están basadas en iconos y eventos, proporcionan un enfoque de programación visual para organizar y presentar multimedia. Los elementos multimedia y las señales de interacción (eventos) se organizan como objetos de un proceso. Normalmente utilizan un diagrama de flujo de actividades con bifurcaciones.

En el desarrollo de una aplicación multimedia es muy importante construir una estructura o diagrama de flujo de eventos, tareas y decisiones, arrastrando los

iconos adecuados desde una biblioteca. Estos iconos pueden incluir selecciones de menú, imágenes gráficas, sonidos y cálculos.

Ejemplos: IconAutor y AuthorwareProfessional [7].

### **3) Herramientas Basadas en Tiempo:**

Las herramientas de desarrollo de multimedia basada en tiempo son las más comunes. Cada una utiliza su propio y único enfoque e interfaces de usuario para administrar eventos en el tiempo. Muchas emplean una línea de tiempo visual para dar secuencia a los eventos de una presentación de multimedia, a menudo desplegando capas con elementos en varios medios o en eventos a lo largo de una escala de incrementos tan precisos que alcanzan una el orden de un treintavo de segundo [8].

Ejemplos: Macromedia Director, Action y MediaBlitz.

#### **1.4.2 Comparación con la propuesta.**

Nadie duda que las herramientas de desarrollo multimedia antes descritas sean muy potentes y sofisticadas. Ellas están concebidas para el trabajo con multimedia en general, o sea, permiten que un usuario cree casi cualquier multimedia que se imagine, por lo que esto trae consigo que sean muy generales y cuenten con un número elevado de herramientas complejas.

El usuario de estas herramientas de desarrollo tiene que dominar el lenguaje de programación que ellas posean y para un operador inexperto le resultará muy difícil su manejo si no ha estudiado previamente su ayuda y consultado documentos o manuales de usuario, los cuales además son extensos he intrincados.

Se necesitan más recursos de hardware y conocimientos previos de programación por parte del usuario para la comprensión y utilización de estas herramientas. La mayoría de ellas se tienen que comprar en el mercado internacional lo que conlleva a

un gasto considerable sin mencionar las limitaciones de patentes y derechos de autor. Necesitan más espacio en memoria y muchas de ellas no se encuentran en idioma español, lo que dificulta su comprensión.

Por otra parte son herramientas que llevan mucho tiempo en el mercado y son muy conocidas, lo que facilita en general la disponibilidad de nuevas versiones y mejoras. Además existe mucha bibliografía actualizada sobre las mismas.

Por otra parte la herramienta propuesta (Nanatool) es más sencilla y se adapta específicamente a la solución del problema planteado. Proporciona una interfaz amigable y fácil acceso a los elementos que se utilizarán en la confección de las multimedia. Se apoya en un sistema de plantillas lo que proporciona mayor agilidad en la elaboración de los productos y la reutilización de las pantallas comunes entre ellos.

No requiere de conocimientos de programación previos. Es completamente gratis y necesita menos recursos de hardware, posee una instalación fácil y ocupa poco espacio en disco. Su reducido tamaño la hace más portable y por supuesto se encuentra en castellano.

### **1.5 Objetivos generales y específicos.**

Debido al problema planteado anteriormente se ha propuesto:

Conocer la manera en que se lleva a cabo el proceso de gestión y producción de software educativo en la Universidad de las Ciencias Informáticas y modelar la herramienta "Nanatool" para la confección de los productos multimedia educativa para niños.

A partir de un análisis del objetivo general se derivan los siguientes *objetivos específicos*:

6. Realizar un estudio detallado del proceso de gestión de software educativo en el centro.

7. Realizar un estudio de las tendencias y tecnologías actuales para presentar una base sólida en la solución de la tarea que nos atañe.
8. Diseñar una Base de Datos capaz de almacenar de manera eficiente la información correspondiente a los productos y elementos multimedia.
9. Diseñar una Base de Datos que contenga la información inherente a las plantillas que sirven de guía a las pantallas del producto final y a los usuarios que las manipulan.
10. Diseñar y modelar una herramienta que permita disminuir el tiempo máximo de elaboración de cada proyecto de multimedia educativa para niños, así como hacer más ágil y amena la realización de los productos ofreciendo un ambiente de trabajo más personalizado y cómodo.

### ***1.6 Descripción de las tendencias y tecnologías actuales sobre las que se apoya la propuesta.***

A continuación se realiza una descripción de las tendencias y tecnologías actuales donde se pueden observar las ventajas que ofrecen para el desarrollo de la aplicación.

- **Programación Orientada a Objetos (POO)**

La programación orientada a objetos es una nueva manera de enfocar la programación.

Los programas estructurados se basan en estructuras de control bien definidas, bloques de código, y subrutinas independientes que soportan recursividad y variables locales.

La POO toma las mejores ideas incorporadas en la programación estructurada y las combina con nuevos y potentes conceptos que permiten organizar los programas de forma más efectiva. La programación orientada a objetos permite descomponer un problema en subgrupos relacionados. Cada subgrupo pasa a ser un objeto autocontenido que contiene sus propias instrucciones y datos que le relacionan con ese objeto. De esta manera, la complejidad se reduce y el programador puede tratar programas más largos.

Todos los lenguajes de POO comparten tres características:

*encapsulación, polimorfismo y herencia.*

- **Encapsulación**

La encapsulación es el mecanismo que agrupa el código y los datos que maneja y los mantiene protegidos frente a cualquier interferencia y mal uso. En un lenguaje orientado a objetos, el código y los datos pueden empaquetarse. Dentro del paquete son necesarios tanto el código como los datos. Cuando el código y los datos están enlazados de esta manera, se ha creado un objeto. En otras palabras, un objeto es el dispositivo que soporta encapsulación.

En un objeto, los datos y el código, o ambos, pueden ser privados para ese objeto o públicos. Los datos o el código privado se les conocen y son accesibles por otra parte del objeto. Es decir, una parte del programa que está fuera del objeto no puede acceder al código o a los datos privados. Cuando el código o los datos son públicos, otras partes del programa pueden acceder a ellos, incluso aunque esté definido dentro de un objeto. Normalmente, las partes públicas de un objeto se utilizan para proporcionar una interfaz controlada a las partes privadas del objeto.

- **Polimorfismo**

Polimorfismo (del griego, cuyo significado es muchas formas) es la cualidad que permite que un nombre se utilice para dos o más propósitos relacionados,

Pero técnicamente diferentes. El propósito del polimorfismo aplicado a la POO es permitir poder usar un nombre para especificar una clase general de acciones.

Dentro de una clase general de acciones, la acción específica a aplicar está determinada por el tipo de dato.

El tipo de datos utilizado para llamar a la función determina que versión específica de la función se está usando

De forma general, el concepto de polimorfismo es la idea de una interfaz de múltiples métodos. Esto significa que es posible diseñar una interfaz genérica para un grupo de actividades relacionadas. Sin embargo, la acción específica ejecutada depende de los datos. La ventaja del polimorfismo es que ayuda a reducir la complejidad permitiendo que la misma interfaz se utilice para especificar una clase general de acción. Es trabajo del compilador seleccionar la acción específica que se aplica a cada situación. El programador no necesita hacer esta selección manualmente. Sólo necesita recordar y utilizar la interfaz general.

- **Herencia**

La Herencia es el proceso mediante el cual un objeto puede adquirir las propiedades de otro. Mas en concreto, un objeto puede heredar un conjunto general de propiedades a las que puede añadir aquellas características que son específicamente suyas. En cualquier caso, la clase hija hereda todas las cualidades asociadas con la clase padre y le añade sus propias características definitorias. Mediante el uso de la herencia, es posible describir un objeto estableciendo la clase general a la que pertenece, junto con aquellas características específicas que le hacen único. La herencia juega un papel muy importante en la POO [12].

- **Aplicaciones Windows y Visual Studio.**

Una parte importante de Visual Studio.NET es la capacidad de crear aplicaciones Windows que se ejecutan localmente en los equipos de los usuarios. Visual Studio .NET permite crear la aplicación con su interfaz de usuario mediante formularios Windows Forms.

Las aplicaciones Windows creadas con clases .NET proporcionan muchos beneficios. Es posible obtener acceso a los servicios del sistema operativo y beneficiarse de las

ventajas que proporciona el entorno de ejecución del usuario. Puede tener acceso a datos utilizando ADO.NET además permite realizar funciones avanzadas de dibujado y pintado en los formularios.

Al igual que con cualquier otra aplicación .NET, se pueden crear aplicaciones Windows, hacer llamadas a métodos y clases .NET, compilar la aplicación y distribuir el programa ejecutable resultante.

La ventaja de utilizar Visual Studio .NET es que proporciona herramientas que hacen que el desarrollo de aplicaciones sea mucho más rápido, sencillo y confiable.

## Microsoft.NET

Microsoft.NET es el conjunto de nuevas tecnologías desarrolladas con el objetivo de obtener una plataforma sencilla y potente. Para crear aplicaciones para la plataforma .NET, tanto servicios Web como aplicaciones tradicionales, Microsoft ha publicado el denominado “kit de desarrollo de software” conocido como .NET Framework SDK, que incluye las herramientas necesarias tanto para su desarrollo como para su distribución y ejecución y Visual Studio.NET, que permite hacer todo lo anterior desde una interfaz visual basada en ventanas [9].

Su infraestructura básica comprendida por: el Framework .NET, Microsoft Visual Studio .NET, .NET Enterprises Servers y Microsoft Windows .NET es lo más visible de .NET, a esto es a lo que uno hace referencia cuando dice .NET, y se trata de todas las tecnologías que conforman el nuevo entorno que le permite a los desarrolladores crear y ejecutar aplicaciones robustas y confiables [10].

### **a) El Framework .NET**

“Se le llama Framework, (“entorno de trabajo”), a las Bibliotecas de Clase Base, (también llamadas BCL) y el Common Language Runtime, (CLR).”[10]

Las bibliotecas de clase son una nueva estructura jerárquica de clases que envuelven diversas funcionalidades como acceso a archivos, hilos de ejecución, acceso a base de datos, etc. Y están disponibles para cualquier lenguaje .NET,

(Visual C++.NET, Visual Basic.NET, Visual C#.NET, ASP.NET, también Visual Fox, Cobol, etc.).

## **b) El CLR, (Common Language Runtime) y algunas de sus características y servicios**

“El CLR es el verdadero núcleo de .NET. Como sugiere el nombre, es un entorno de ejecución en el que las aplicaciones escritas en diferentes lenguajes pueden ejecutarse a la vez y llevarse bien; algo también conocido como interoperabilidad multilenguaje”. [11]

- 1) **Modelo de programación consistente:** A todos los servicios y facilidades ofrecidos por el CLR se accede de la misma forma: a través de un modelo de programación orientado a objetos. Esto es una diferencia importante respecto al modo de acceso a los servicios ofrecidos por los algunos sistemas operativos actuales (por ejemplo, los de la familia Windows), en los que a algunos servicios se les accede a través de llamadas a funciones globales definidas en dlls y a otros a través de objetos (objetos COM en el caso de la familia Windows)
- 2) **Ejecución multiplataforma:** El CLR actúa como una máquina virtual, encargándose de ejecutar las aplicaciones diseñadas para la plataforma .NET. Es decir, cualquier plataforma para la que exista una versión del CLR podrá ejecutar cualquier aplicación .NET. Microsoft ha desarrollado versiones del CLR para la mayoría de las versiones de Windows: Windows 95, Windows 98, Windows ME, Windows NT 4.0, Windows 2000, Windows XP y

Windows CE. Por otro lado Microsoft ha firmado un acuerdo con Corel para portar el CLR a Linux y también hay terceros que están desarrollando de manera independiente versiones de libre distribución del CLR para Linux. Asimismo, dado que la arquitectura del CLR está totalmente abierta, es posible que en el futuro se diseñen versiones del mismo para otros sistemas operativos.

- 3) **Integración de lenguajes:** Desde cualquier lenguaje para el que exista un compilador que genere código para la plataforma .NET es posible utilizar código generado para la misma usando cualquier otro lenguaje tal y como si de código escrito usando el primero se tratase. Microsoft ha desarrollado un compilador de C# que genera código de este tipo. La integración de lenguajes esta en que es posible escribir una clase en C# que herede de otra escrita en Visual Basic.NET que, a su vez, herede de otra escrita en C++ con extensiones gestionadas.
  
- 4) **Gestión de memoria:** El CLR incluye un **recolector de basura** que evita que el programador tenga que tener en cuenta cuándo ha de destruir los objetos que dejen de serle útiles. Este recolector es una aplicación que se activa cuando se quiere crear algún objeto nuevo y se detecta que no queda memoria libre para hacerlo, caso en que el recolector recorre la memoria dinámica asociada a la aplicación, detecta qué objetos hay en ella que no puedan ser accedidos por el código de la aplicación, y los elimina para limpiar la memoria de “objetos basura” y permitir la creación de otros nuevos. Gracias a este recolector se evitan errores de programación muy comunes como intentos de borrado de objetos ya borrados, agotamiento de memoria por olvido de eliminación de objetos inútiles o solicitud de acceso a miembros de objetos ya destruidos.

- 5) **Tratamiento de excepciones:** En el CLR todos los errores que se puedan producir durante la ejecución de una aplicación se propagan de igual manera: mediante excepciones. Esto es muy diferente a como se venía haciendo en los sistemas Windows hasta la aparición de la plataforma .NET.

El CLR permite que excepciones lanzadas desde código para .NET escrito en un cierto lenguaje se puedan capturar en código escrito usando otro lenguaje, e incluye mecanismos de depuración que pueden saltar desde código escrito para .NET en un determinado lenguaje a código escrito en cualquier otro.

**Soporte multihilo:** El CLR es capaz de trabajar con aplicaciones divididas en múltiples hilos de ejecución que pueden ir evolucionando por separado en paralelo o intercalándose, según el número de procesadores de la máquina sobre la que se ejecuten. Las aplicaciones pueden lanzar nuevos hilos, destruirlos, suspenderlos por un tiempo o hasta que les llegue una notificación, enviarles notificaciones, sincronizarlos, etc.

- 6) **Interoperabilidad con código antiguo:** El CLR incorpora los mecanismos necesarios para poder acceder desde código escrito para la plataforma .NET a código escrito previamente a la aparición de la misma y, por tanto, no preparado para ser ejecutando dentro de ella. Estos mecanismos permiten tanto el acceso a objetos COM como el acceso a funciones sueltas de dlls preexistentes. [9]

### c) **Microsoft Intermediate Language, (MSIL).**

“Una aplicación .NET compilada por con un compilador .NET, generará un archivo.exe pero que no contiene código máquina sino instrucciones en MSIL.”[10]

Todos los compiladores que generan código para la plataforma .NET generan código escrito en el lenguaje intermedio conocido como Microsoft Intermediate Language (MSIL). Este lenguaje es el único código que es capaz de interpretar el

CLR, y por tanto cuando se dice que un compilador genera código para la plataforma .NET lo que se está diciendo es que genera MSIL.

La principal ventaja del MSIL es que facilita la ejecución multiplataforma y la integración entre lenguajes al ser independiente de la CPU y proporcionar un formato común para el código máquina generado por todos los compiladores que generen código para .NET. Sin embargo, dado que las CPUs no pueden ejecutar directamente MSIL, antes de ejecutarlo habrá que convertirlo al código nativo de la CPU sobre la que se vaya a ejecutar. De esto se encarga un componente del CLR conocido como compilador JIT (Just-In-Time) o jitter que va convirtiendo dinámicamente el código MSIL a ejecutar en código nativo según sea necesario. [9]

#### **d) Seguridad**

La faceta más importante de cualquier entorno de desarrollo de aplicaciones es cómo maneja la seguridad. Afortunadamente .NET pone muchos conceptos sobre la mesa. De hecho, la seguridad comienza tan pronto como se carga una clase por el CLR, ya que el cargador de clases es parte del esquema de seguridad de .NET. Por ejemplo, cuando se carga una clase en el entorno de ejecución .NET, se comprueban los factores relacionados con la seguridad tales como reglas de accesibilidad y requisitos de autoconsistencia. Además, los chequeos de seguridad aseguran que un trozo de código tenga las credenciales apropiadas para acceder a ciertos recursos. El código de seguridad asegura la determinación de los roles y la información de identidad [11].

### **1.7 ¿Qué es un Proceso de Desarrollo de Software?**

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software.

Un proceso define *quién* está haciendo *qué*, *cuándo*, y *cómo* alcanzar un determinado objetivo. En la ingeniería del software el objetivo es construir un producto de software o

mejorar uno existente. Un proceso efectivo proporciona normas para el desarrollo eficiente de software de calidad. Captura y presenta las mejores prácticas el estado actual de la tecnología permite. En consecuencia, reduce el riesgo y hace el proyecto más predecible.

### **1.7.1 El Proceso Unificado de Desarrollo de software (RUP)**

El Proceso Unificado es más que un simple proceso de desarrollo, es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto [13].

- **El Proceso Unificado está dirigido por casos de uso**

“Un sistema de software ve la luz para dar servicio a sus usuarios. Por tanto, para construir un sistema con éxito debemos conocer lo que sus futuros usuarios necesitan o desean” [13].

Un caso de uso es un fragmento de funcionalidad del sistema que proporciona al usuario un resultado importante. Los casos de uso representan los requisitos funcionales. Todos los casos de uso juntos constituyen el modelo de casos de uso el cual describe la funcionalidad total del sistema.

Los casos de uso explican los requisitos de un sistema y guían su diseño, implementación y prueba; o sea, guían el proceso de desarrollo. Basándose en el modelo de casos de uso, los desarrolladores crean una serie de modelos de diseño he implementación que llevan a cabo los casos de uso. Los desarrolladores revisan cada uno de los sucesivos modelos para que sean conformes al modelo de casos de uso. Los ingenieros de prueba prueban la implementación para garantizar que los componentes del modelo de implementación implementan correctamente los casos de uso. De este modo, los casos de uso no sólo inician el proceso de desarrollo sino que le proporcionan un hilo conductor [13].

- **El Proceso de Unificado está centrado en la arquitectura**

El concepto de arquitectura de software incluye los aspectos estáticos y dinámicos más significativos del sistema. La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se refleja en los casos de uso. Sin embargo, también se ve influido por muchos otros factores, como la plataforma en que tiene que funcionar, el software (arquitectura hardware, sistema operativo y sistema de gestión de Bases de Datos). La arquitectura es una vista del diseño completo con las características más importantes resaltadas, dejando los detalles de lado. Debe haber interacción entre los casos de uso y la arquitectura. Por un lado, los casos de uso deben encajar en la arquitectura cuando se llevan a cabo. Por otro lado la arquitectura debe permitir el desarrollo de todos los casos de uso requeridos. En realidad tanto la arquitectura como los casos de uso deben evolucionar en paralelo [13].

- **El proceso Unificado es iterativo e incremental**

El desarrollo de un producto de software comercial supone un gran esfuerzo que puede durar entre varios meses hasta posiblemente un año o más. Es práctico dividir el trabajo en partes más pequeñas o “miniproyectos”. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. Para una efectividad máxima, las iteraciones deben estar *controladas*; esto es, deben seleccionarse y ejecutarse de una forma planificada. En cada iteración los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con los objetivos el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple con los objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque.

Estos conceptos, los de desarrollo dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones mientras que los casos

de uso definen los objetivos y dirigen el trabajo de cada iteración. La eliminación de una de las tres ideas reducirá drásticamente el valor del Proceso Unificado [13].

El lenguaje de modelado UML, hace una abstracción capturando los aspectos esenciales de un sistema, se utiliza para representar la metodología RUP, ya que permite organizar, encontrar, filtrar, recuperar examinar y corregir la información en los sistemas, usándose para entender , diseñar, configurar, mantener, y controlar la información sobre los sistemas, ayudando a los desarrolladores a explorar varias arquitecturas y soluciones fáciles antes de escribir el código .

### **1.8 Lenguajes y gestores de bases de datos utilizados**

En este epígrafe se describen las características fundamentales del gestor de base de datos y los lenguajes de programación, así como otros software que serán utilizados.

#### **1.8.1 C#**

“C# es el primer lenguaje orientado a componentes en la familia de lenguajes C y C++. Es un lenguaje de programación simple, moderno, orientado a objetos y con un sistema de tipos seguro derivado de C y C++.El lenguaje C# combina la alta productividad de Microsoft Visual Basic y la eficacia bruta de C++” [11].

El lenguaje C# es el único que ha sido diseñado específicamente para ser utilizado en .Net por lo que utilizarlo en esta plataforma es mucho más sencillo e intuitivo que hacerlo con cualquiera de los otros lenguajes. Por esta razón, se suele decir que C# es el *lenguaje nativo de .NET*.

La sintaxis y estructuración de C# es muy similar a la C++, lo que facilita la migración de códigos escritos en este último y facilita su aprendizaje a los desarrolladores habituados a él. Algunas de las características principales del C# se describen a continuación:

- **Sencillez:** Elimina muchos elementos que otros lenguajes incluyen y que son innecesarios en .NET. Por ejemplo:
  - El código escrito en C# es *autocontenido*, lo que significa que no necesita de ficheros adicionales al propio fuente tales como ficheros de cabecera o ficheros IDL.
  - El tamaño de los tipos de datos básicos es fijo e independiente del compilador, sistema operativo o máquina para quienes se compile, lo que facilita la portabilidad del código.
- **Modernidad:** C# incluye una instrucción *foreach* que permite recorrer colecciones con facilidad y es ampliable a tipos definidos por el usuario, contiene además un tipo básico *string* para representar cadenas y la distinción de un tipo *bool* específico para representar valores lógicos..
- **Orientación a objetos:** Como todo lenguaje de programación de propósito general actual, C# es un lenguaje orientado a objetos. Una diferencia de este enfoque con respecto a otros lenguajes es que es más puro en tanto que no admite ni funciones ni variables globales sino que todo el código y datos han de definirse dentro de definiciones de tipos de datos, lo que reduce problemas por conflictos de nombres y facilita la legibilidad del código.
- **Orientación a componentes:** La propia sintaxis de C# incluye elementos propios del diseño de componentes que otros lenguajes tienen que simular mediante construcciones más o menos complejas. Es decir, la sintaxis de C# permite definir cómodamente *propiedades* (similares a campos de acceso controlado), *eventos* (asociación controlada de funciones de respuesta a notificaciones) o *atributos* (información sobre un tipo o sus miembros)

- **Seguridad de tipos:** C# incluye mecanismos que permiten asegurar que los accesos a tipos de datos siempre se realicen correctamente, lo que evita que se produzcan errores difíciles de detectar por acceso a memoria no perteneciente a ningún objeto y es especialmente necesario en un entorno gestionado por un recolector de basura
- **Sistema de tipos unificado:** A diferencia de C++, en C# todos los tipos de datos que se definan siempre derivarán, aunque sea de manera implícita, de una clase base común llamada *System.Object*, por lo que dispondrán de todos los miembros definidos en ésta clase (es decir, serán “objetos”).
- **Eficiente:** En principio, en C# todo el código incluye numerosas restricciones para asegurar su seguridad y no permite el uso de punteros. Sin embargo en C# es posible saltarse dichas restricciones manipulando objetos a través de punteros.
- **Compatible:** Para facilitar la migración de programadores, C# no sólo mantiene una sintaxis muy similar a C, C++ o Java que permite incluir directamente en código escrito en C# fragmentos de código escrito en estos lenguajes, sino que el CLR también ofrece, a través de los llamados *Platform Invocation Services (PInvoke)*, la posibilidad de acceder a código nativo escrito como funciones sueltas no orientadas a objetos tales como las dlls de la API Win32. [9]

### 1.8.2 Director y Lingo

Director le permite crear sofisticadas aplicaciones multimedia en un solo programa y poder realizar distintas versiones de su aplicación que quedarán optimizadas para las distintas plataformas y formatos multimedia [14].

Director no solo es un catalizador para adentrarnos en el mundo de la programación, es un producto único y lo suficientemente fácil e intuitivo en su manejo como para que una persona pueda crear su aplicación desde la primera vez que se sienta a trabajar con

él. Al mismo tiempo, tiene la capacidad de trabajo que permite a los desarrolladores más expertos elaborar aplicaciones multimedia. Además posee un lenguaje de elaboración de guiones llamado Lingo que le proporcionará un control total sobre cada uno de los aspectos del programa.

Este potente lenguaje de programación (Lingo) está orientado a la resolución de aspectos fundamentales de las presentaciones multimedia, tales como la interactividad y el control de diferentes tipos de media: textos, gráficos, sonido, vídeo y otros.

Permite programar las acciones de los usuarios controlando el comportamiento de los actores en el escenario sincronizándolos con las acciones del ratón. Lingo está especialmente indicado para la creación de juegos interactivos, software educativo, presentaciones corporativas, catálogos de empresa y un sinnúmero de posibilidades.

Director utiliza una “metáfora” que se ajusta muy bien a la multimedia: el mundo del cine y la animación. A un archivo de Director se le llama movie (película), la ventana donde se produce la animación o la aplicación interactiva se llama Stage (escenario). Los Sprites o canales que son los actores de la película, tienen distintos Behaviors (comportamiento) que dependen de unos Scripts o guiones. A los diferentes recursos utilizados en el programa se les llama Cast members o miembros del reparto. Las ubicaciones generales y la coreografía de los sprites se pueden observar y controlar con el Store o partitura, que se asemeja a la partitura de un animador. Al igual que ocurre en la animación tradicional, los sprites se dibujan en el Stage (escenario) utilizando distintos Inks o tinteros. [14]

Director brinda una interfaz amena y sugerente que le proporciona al desarrollador de software elementos del mundo real con los cuales relacionar la programación y comportamiento de los elementos de las aplicaciones multimedia.

### **1.8.3 Herramienta de modelado de Datos (ER/Studio)**

Es una herramienta de modelado de datos fácil de usar y multinivel, para el diseño y construcción de bases de datos a nivel físico y lógico. Direcciona las necesidades

diarias de los administradores de bases de datos, desarrolladores y arquitectos de datos que construyen y mantienen estas aplicaciones.

ER/Studio está equipado para crear y manejar diseños de bases de datos funcionales y confiables. Ofrece fuertes capacidades de diseño lógico, sincronización bidireccional de los diseños físicos y lógicos, construcción automática de bases de datos, documentación y fácil creación de reportes.

Entre sus principales funcionalidades están:

- Capacidad fuerte en el diseño lógico.
- Sincronización bidireccional de los diseños lógico y físico.
- Construcción automática de Base de Datos.
- Reingeniería inversa de Base de Datos.
- Documentación basada en HTML.

“ER/Studio se diseña para hacer más fácil de entender el estado actual de los datos de la empresa. Simple y fácil al usuario, ayuda a organizaciones para tomar decisiones en cómo resolver embotellamientos de los datos, elimina redundancia y alcanza en última instancia usos de más alta calidad que entreguen datos más eficientes y exactos a la empresa.”[15]. Por lo antes expuesto consideramos adecuado utilizar ER/Studio para la modelación de nuestras bases de datos.

#### **1.8.4 Gestores de Base de Datos**

Existen muchísimos software especializados en bases de datos ,entre ellos los tres más comunes son SQL-SERVER de Microsoft, ORACLE Server de Oracle y MYSQL de Open Source, en estos casos la base de datos reside en un servidor especializado en algún lugar cercano o lejano en una red chica, mediana o grande.

Otros paquetes o software más pequeños y comunes también reciben el nombre de

DBMS (DATA BASE MANAGEMENT SYSTEM) o sistemas administradores de bases de datos.

Este tipo de software se especializa en la creación, mantenimiento, seguridad y privacidad de una base de datos, DBMS comunes son: access, postgres, fox, clipper, etc.

Usaremos Microsoft Access como nuestro generador de bases de datos ya que se ajusta a las características de nuestro sistema. Como producto final se comercializará en CD y es de mediana complejidad en cuanto a base de datos se refiere no es necesario utilizar un servidor robusto como los antes mencionados que están concebidos para el manejo de altas concurrencias. En este caso específico el Access nos brinda toda la rapidez y funcionalidad que nuestra multimedia necesita, además de ser más portable y sencillo.

### ***1.9 Conclusiones***

Al concluir esta primera parte se ha tenido en cuenta los conceptos acerca de la multimedia y los términos que giran alrededor de la misma. Se hace necesario puntualizar que aunque el producto resultante de la aplicación pueda ser en ocasiones una “hipermedia” para un mejor entendimiento se continuará utilizando el término “multimedia” para describirlo, ya que es el más utilizado por la mayoría de las personas relacionadas con el producto.

Se logra definir los lenguajes de programación y los gestores de base de datos a utilizar, se ha identificado el objeto del estudio y la problemática presente hoy que hizo inminente la necesidad de concebir una herramienta que la resuelva. Este capítulo tiene gran importancia para poder realizar un buen análisis y diseño de la aplicación.

## **Capítulo 2 Descripción de la solución propuesta.**

### ***2.1 Introducción.***

En este Capítulo se da a conocer la modelación del negocio, el mismo incluirá los actores y trabajadores del negocio que se tendrán en consideración, los casos de uso del negocio, el diagrama de actividades correspondiente que resume el flujo de actividades del negocio, los diagramas de clases de objetos, los requerimientos del sistema y los casos de uso del sistema, además se ofrece una descripción del mismo. En el presente capítulo se realiza todo el diseño para tener una mejor comprensión de lo que se quiere obtener con el sistema.

### ***2.2 Reglas del negocio a considerar***

- Existe una estrecha relación en el trabajo realizado por cada equipo de desarrollo por lo que existe una estructura que facilita la comunicación entre los equipos y sus miembros.
- Los guionistas pueden definir elementos del diseño pero solo se tomarán en calidad de sugerencias por los diseñadores del producto.
- Se define una estructura de directorio en la que se almacena los elementos creados por los distintos equipos de desarrollo, garantizando que todos sus miembros tengan acceso a ellos, aunque los permisos difieren en dependencia del tipo de usuario.
- Se cuenta con una estructura sólida que garantiza el control de versiones de los distintos productos.

- Los guiones de contenido y multimedia deben ser entregados en tiempo y siguiendo los estándares para este tipo de documentos.

### ***2.3 Descripción de los procesos del negocio propuestos.***

Tras la implantación de la propuesta de aplicación los procesos del negocio sufrirán notables cambios. En un principio continuará el flujo de actividades normalmente hasta llegar al proceso de programación de los productos donde juega un papel importante la herramienta propuesta, mediante ella se conformarán completamente los productos multimedia. De esta forma quedan abolidos los programadores ya que no es necesaria la programación de los proyectos, la herramienta se encarga de esa actividad. Durante esta etapa la herramienta necesitará a un operador que puede ser un miembro del equipo de desarrollo de diseño. Los programadores pueden pasar a formar parte del equipo de diseñadores y encargarse de operar la herramienta.

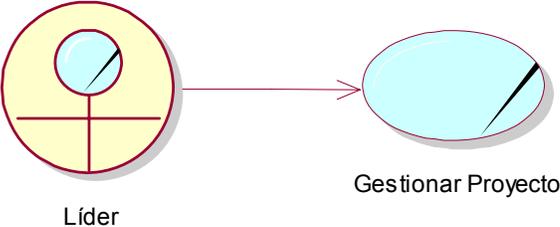
De esta forma se logra reducir el tiempo de confección de los productos porque no se tienen que programar cada uno de ellos sino que sólo se crean en la herramienta las pantallas de los productos a partir de los elementos multimedia que la conforman y la aplicación se encarga de ensamblar dinámicamente el producto final, evitándole a los desarrolladores cualquier contacto con códigos o lenguajes de programación para dicha tarea. Esto resulta muy útil, se puede reducir el número de personal requerido para el proyecto y no es necesario tener amplios conocimientos de programación para la culminación de los proyectos.

### 2.3.1 Descripción de los Actores y Trabajadores del negocio

Trabajadores del negocio	Justificación
<b>Líder</b>	Persona(s) encargada(s) de asignar y controlar los proyectos.
<b>Pedagogo</b>	Persona o equipo de trabajo encargado de elaborar el guión de contenido.
<b>Guionista de Multimedia</b>	Persona o equipo de trabajo encargado de elaborar el guión multimedia que se utilizará durante todo el proceso de confección.
<b>Diseñador de Elementos</b>	Persona o equipo de trabajo encargado de elaborar los elementos audiovisuales que componen al producto multimedia educativa (fotos, videos, etc.), estos elementos son utilizados por los diseñadores de interfaz y programadores para crear el producto final.
<b>Diseñador de Interfaz</b>	Persona o conjunto de personas que se encargan del diseño la interfaz de cada producto así como de la elaboración de componentes que serán usados por los programadores en elaboración del producto como: botones y animaciones.
<b>Programador</b>	Persona o equipo de trabajo encargado de programar la aplicación multimedia, son los responsables de crear la interacción entre los elementos que conforman el producto.

Tabla 1 Descripción de los Trabajadores del sistema

**2.3.2 Diagrama de casos de uso del negocio**



**Figura 1 Diagrama de Casos de Uso del Negocio**

### 2.3.3 Diagrama de Actividades

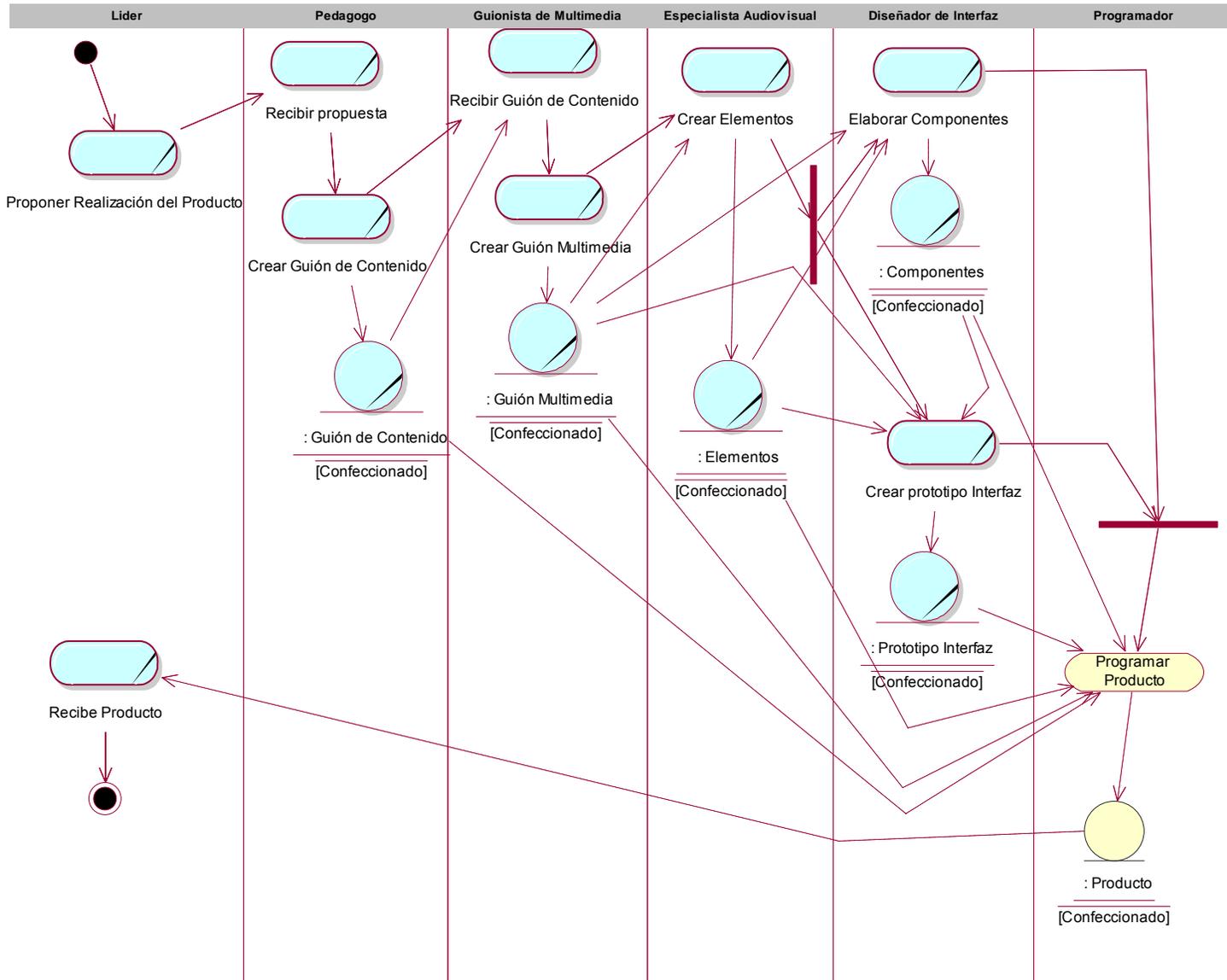


Figura 2 Diagrama de Actividades del Modelo del Negocio

## 2.4 Diagrama de clases del modelo de objetos.

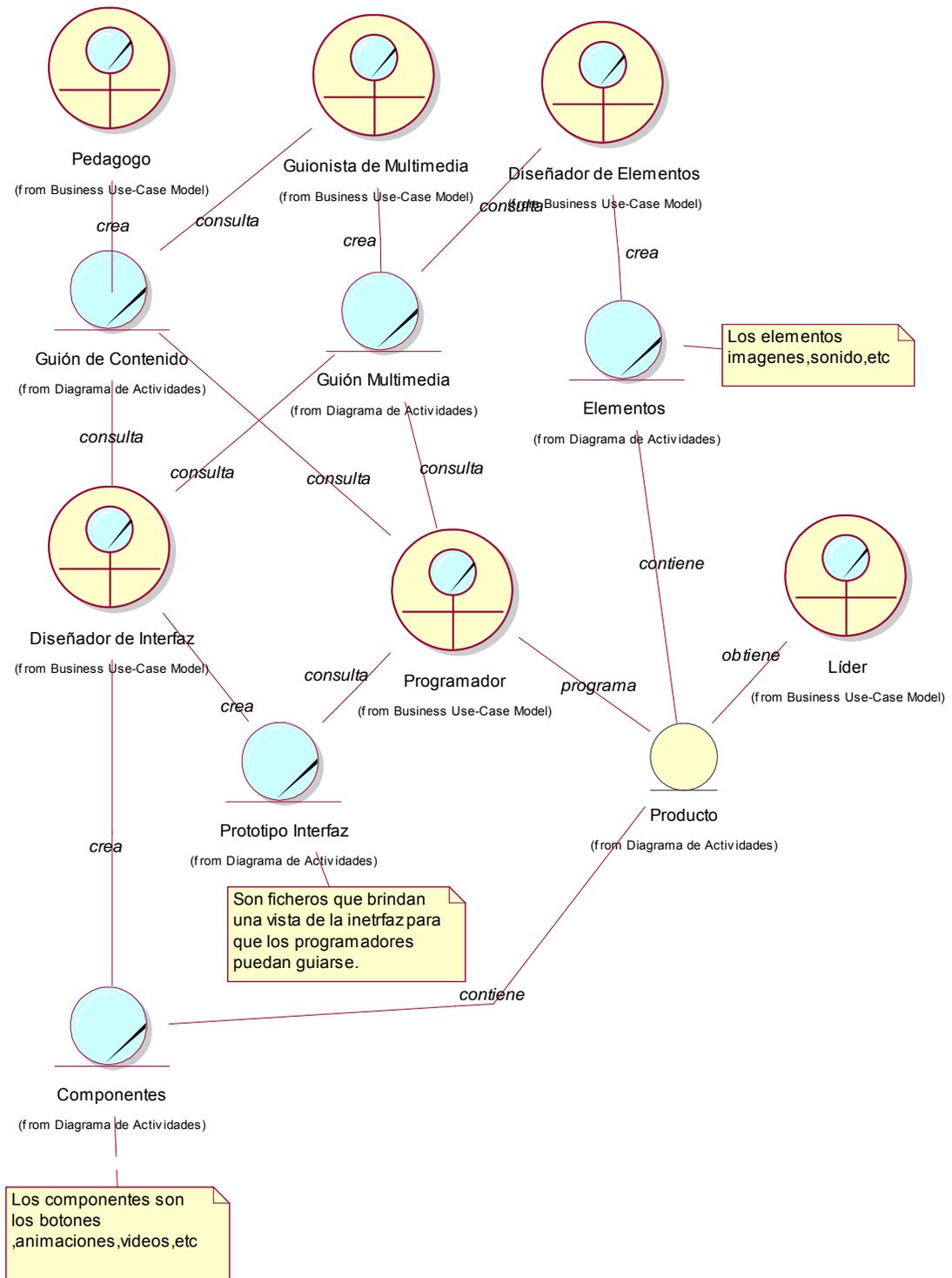


Figura 3 Diagrama de Clases del Modelo de Objetos

## **2.5 Requisitos funcionales.**

- R1 Autenticarse en el sistema, el usuario debe introducir sus nombre de usuario y contraseña.
- R2 Administrar usuarios del sistema, el administrador tiene la facultad de modificar los privilegios de los demás usuarios del sistema.
- R3 Registrar un nuevo usuario. El administrador debe entrar de forma obligatoria los siguientes datos del usuario: Nombre personal y apellido, nombre de usuario, contraseña de autenticación y el rol que desempeña en el sistema.
- R4 Eliminar un usuario registrado.
- R5 Cambiar contraseña de autenticación.
- R6 Crear, modificar y eliminar plantillas.
- R7 Insertar y eliminar elementos en la plantilla.
- R8 Modificar propiedades de los elementos de la plantilla (largo, ancho, coordenadas, etc.)
- R9 Almacenar propiedades de los elementos de la plantilla.
- R10 Visualizar plantillas disponibles en el área de trabajo.
- R11 Guardar plantilla.
- R12 Cargar plantilla.
- R13 Crear, modificar y eliminar pantallas desde el área de trabajo.
- R14 Insertar elementos en la pantalla (imágenes sonido, animaciones, texto, botones, video).
- R15 Visualizar tipos de botones, pueden ser sonido, play, pausa, stop, adelante, atrás, juego1, juego2, etc.
- R16 Sustituir elementos de la plantilla por los propios de la pantalla.
- R17 Definir tipos de pantalla.
- R18 Visualizar listado de pantallas secuenciales por Tema, los Temas pueden ser Lectura y Video, los cuales cuentan con una serie de pantallas que poseen un orden determinado entre sí.
- R19 Crear orden de pantallas secuenciales en el área de trabajo.
- R20 Organizar los elementos de la pantalla por orden de profundidad, este orden es el que define cual elemento está por encima de otro con respecto al fondo.

R21 Visualizar pantallas existentes.

R22 Visualizar tipos de pantallas, pueden ser presentación, guía, lectura, intermedio, juego, video, final, etc.

R23 Cargar pantallas en el área de trabajo.

R24 Cargar pantallas desde la aplicación final (módulo 2).

R25 Cargar elementos de la pantalla desde aplicación final (módulo 2).

R26 Capturar orden de pantallas secuenciales desde aplicación final (módulo 2).

R27 Visualizar pantalla correspondiente desde la aplicación final (modulo 2).

R28 Sustituir elementos típicos de la pantalla.

R29 Comparar elementos.

R30 Imprimir Imagen.

R31 Asignarle propiedades a los elementos de la pantalla.

## ***2.6 Requisitos no funcionales.***

- **Apariencia o interfaz externa**

El sistema debe contener una interfaz amigable al usuario, brindándole todas las facilidades necesarias para la entrada de datos logrando que esta sea lo más cómoda y agradable posible.

En cuanto a la segunda parte del sistema como va dirigida específicamente a niños de pequeña edad, la interfaz debe ser sugerente y atractiva, debe combinar colores llamativos que motiven a los niños.

- **Usabilidad**

El sistema debe permitir el acceso a los usuarios y lograr una interfaz y navegación asequibles y funcionales tanto para usuarios expertos como para los que no tienen conocimientos profundos de informática, de forma que los usuarios que interactúen con la aplicación no presenten dificultades en la asimilación de la herramienta.

- **Rendimiento**

En cuanto al tratamiento de la información el sistema debe garantizar la mayor eficiencia posible logrando que la velocidad de procesamiento sea tal alta como se pueda asegurando un tiempo óptimo de respuesta para evitar que la acción del usuario se torne agobiante y aburrida. Se debe garantizar la disponibilidad y consistencia de la información en todo momento. Además se requiere que el tiempo de recuperación de los datos sea el menor posible.

- **Soporte**

- a) PC con sistema operativo Windows 95, Milenium ,2000, XP o superior
- b) Lector de CD con tasa de transferencia de 24X (3.600KB/s) o superior.
- c) Tarjeta de Sonido de 16 bits
- d) Impresora a color
- e) Bocinas o audífonos.

- **Portabilidad**

El segundo módulo del sistema, que va a ser la parte que se comercializará y con la cual interactúa el usuario final debe almacenarse en un CD. Tanto el producto, como la Base de Datos a el asociada.

- **Seguridad**

En cuanto a la seguridad física depende mucho del usuario y las condiciones que pueda crear para mantener la información segura, en el proceso de creación se asegura la integridad del sistema y la base de datos, ya que la misma se encuentra en la Dirección de Software Educativo de la UCI, en cuyos laboratorios está instalado sistemas de alarma contra incendios, custodios y otros medios de seguridad del centro.

Disponemos de una base de datos que solo la persona que posea o conozca un código de usuario concreto y su contraseña asociada, será capaz de entrar en ella.

Esta persona, que es la propietaria de la base de datos, puede crear grupos de usuario y usuarios a los que puede dar autorizaciones de uso concretas.

Nadie puede llevarse la base de datos **SEGURA** a otro sistema Access y acceder a ella, ni como administrador, ni como ningún otro usuario, excepto sepa el código de usuario y la contraseña.

- **Políticos-Culturales**

El sistema debe estar diseñado en varios idiomas, para que logre adaptarse a las características del lenguaje del país del usuario final.

- **Confiabilidad**

El sistema debe estar protegido contra fallos por lo que se tendrá una copia de la Base de Datos y se le aplicará un plan de mantenimiento a la misma.

- **Software**

- Rational Rose Enterprise Edition para la modelación del sistema
- ER/Studio 5.0 para la modelación de la base de Datos
- Microsoft Access como gestor de Base de Datos
- Para la construcción/modificación/visualización de imágenes:  
Microsoft Paint , Adobe Photoshop 8.0.
- Macromedia Director MX, para la elaboración de la segunda parte del sistema.
- Macromedia Flash MX 2004 para la creación de animaciones y videos.
- C# para la creación del Módulo 1 del sistema donde se realiza toda la entrada de datos así como la conformación y diseño de las pantallas futuras del software.

- **Hardware**

- a) Para la utilización de la aplicación por los usuarios, estos requieren de una computadora con las siguientes características:
- b) Microprocesador:486 a 66Mhz

- c) Memoria RAM: 32Mb
- d) Disco Duro: 2Gb
- e) Periféricos: Teclado, tarjeta de sonido, ratón, audífonos o bocinas, Impresora a color.

- **Restricciones en el diseño y la implementación.**

- 1) Existen algunas restricciones en cuanto al producto final, estas son:
- 2) 800 x 600 pixels de resolución de pantalla
- 3) Debe funcionar desde el CD y desde el HD
- 4) En el producto final cada CD contendrá solamente un cuento en varios idiomas.
- 5) Debe ejecutarse por los elementos reproductores de medias estándares de Windows
- 6) Inhibir la función de la combinación de teclas Alt+Tab

## ***2.7 Descripción del sistema propuesto.***

El presente epígrafe detalla el sistema mediante una descripción general del mismo y la representación del modelo de casos de uso del sistema divididos por paquetes, facilitando así su comprensión y entendimiento. Este epígrafe es de suma importancia para la futura implementación de la herramienta.

### ***2.7.1 Concepción general del sistema.***

La herramienta que se propone desarrollar automatiza la confección de productos multimedia educativa para niños. Este sistema consta de dos módulos principales los cuales mantienen una relación muy estrecha durante todo el proceso de confección ya que el módulo dos depende de la correcta utilización del módulo uno.

- **Modulo1:** *Entrada de Datos*

Este módulo es el encargado de automatizar la gestión de la información correspondiente a las plantillas, pantallas y los elementos de multimedia que las conforman.

Para la implementación de dicho módulo se utilizará la plataforma .Net con el lenguaje C# para explotar las facilidades que este lenguaje brinda, las cuales fueron descritas en capítulos anteriores. Para el control de los elementos que conforman las plantillas y pantallas se utilizan los controles de .Net que nos brindan toda la funcionalidad necesaria para modelar las propiedades de estos elementos. Este módulo de la herramienta le facilita al desarrollador la inserción y manipulación de los elementos para conformar las plantillas que servirán de guía para crear las pantallas del producto final.

El módulo cuenta además con políticas de seguridad otorgando a cada usuario los derechos que le corresponden. Existirán dos tipos de usuario: administradores y diseñadores. Los administradores son los encargados de otorgar los roles a los demás usuarios del sistema, además es el único autorizado a crear plantillas y salvar su contenido en la base de datos correspondiente a las mismas. Por otra parte puede explotar todas las funcionalidades que brinda el sistema descrito. Los diseñadores se encargan de crear las pantallas a partir de las plantillas diseñadas por el administrador, a este usuario se le permite insertar elementos en la pantalla y gestionarlas, pero no se le concede modificar o alterar el contenido de las plantillas en la base de datos correspondiente.

Naturalmente todo usuario para ingresar al sistema debe pasar por un proceso de autenticación, para garantizar la seguridad se utiliza el cifrado de la contraseña del usuario. Luego se le otorgan los roles correspondiente y es libre de realizar las acciones que le conciernen.

- **Módulo 2** *Presentación*

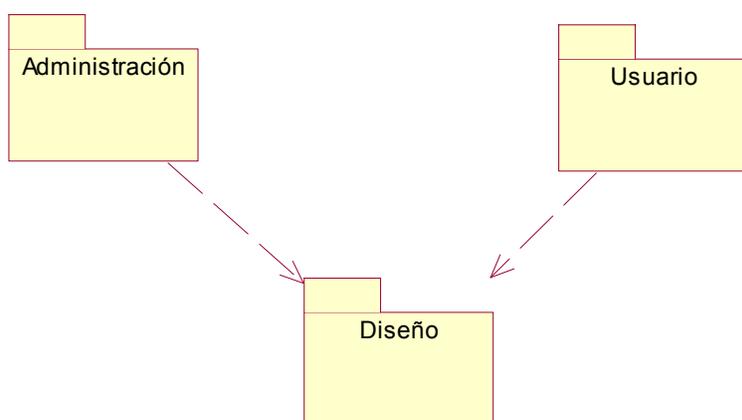
Este módulo es operado exclusivamente por el Usuario y no consta de nivel de seguridad alguno en cuanto usuarios y roles se refiere, los usuarios de este

módulo son niños entre la edad de 4 a 7 años por lo cual presenta una interfaz atractiva y sugerente. Este módulo se especializa en cargar de forma dinámica las pantallas de la base de datos correspondiente he ir ensamblando las mismas para mostrarlas al usuario, la transición y navegabilidad entre las pantallas dan forma al producto final. Para la implementación de este módulo se utiliza Director MX 2004 con su lenguaje de programación Lingo Scripts ya que ofrece poderosas herramientas para el trabajo con multimedia y otras cualidades antes descritas, además se persigue la homogeneidad con los demás productos desarrollados en la entidad.

Se emplea Access como gestor de base de datos para almacenar la información correspondiente a los productos. Para el óptimo funcionamiento de la herramienta se crean dos bases de datos, la primera contiene la información propia de las plantillas y sus elementos y la segunda alberga los datos referentes a las pantallas y sus componentes.

### **2.7.2 Modelo de casos de uso del sistema**

Para modelar los casos de usos del sistema se ha decidido dividir estos en diferentes paquetes atendiendo a las funcionalidades de cada actor:



**Figura 4 Diagrama de Paquetes de Casos de Uso del Sistema**

### 2.7.3 Descripción de los actores del sistema

#### Administrador:

Persona encargada de velar por la correcta manipulación de la herramienta, es quien otorga los permisos a los restantes usuarios del sistema. Es el único autorizado a crear plantillas en la base de datos y goza de todos los beneficios del sistema.

#### Diseñador:

Persona encargada de crear las pantallas que dan forma al producto final, tiene acceso restringido al sistema, no se le permite eliminar ni modificar plantillas de la base de datos.

**Usuario:** Niño de 4 a 7 años de edad interesado en la aplicación, es la persona que interactúa con el producto final y se beneficia de él.

### 2.7.4 Paquete Administración

En este paquete se muestran los casos de uso asociados con el Administrador. El cual es el encargado de registrar a los usuarios y modificar sus roles y propiedades, además de crear las plantillas que servirán de base para la creación de las futuras pantallas.

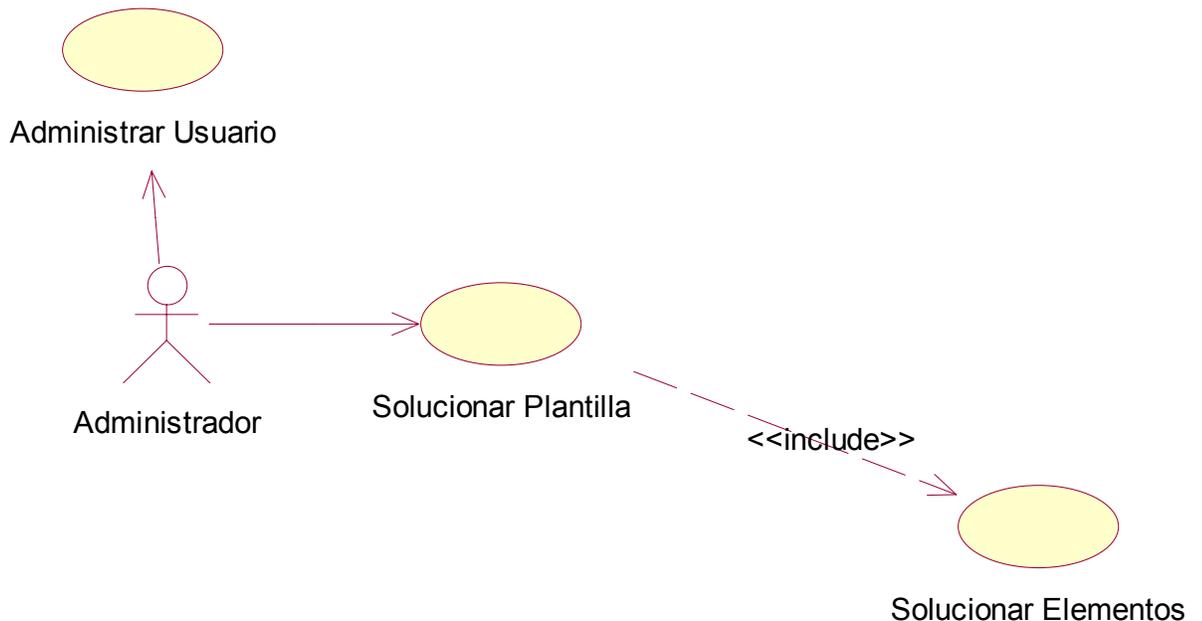


Figura 5 Diagrama de Casos de Uso Paquete Administración

### 2.7.5 Paquete Diseño

En este Paquete se muestran los casos de uso relacionados con el Diseñador, el cual debe autenticarse y así obtener los privilegios sobre el sistema ,es el encargado de crear las pantallas que conforman el producto final.

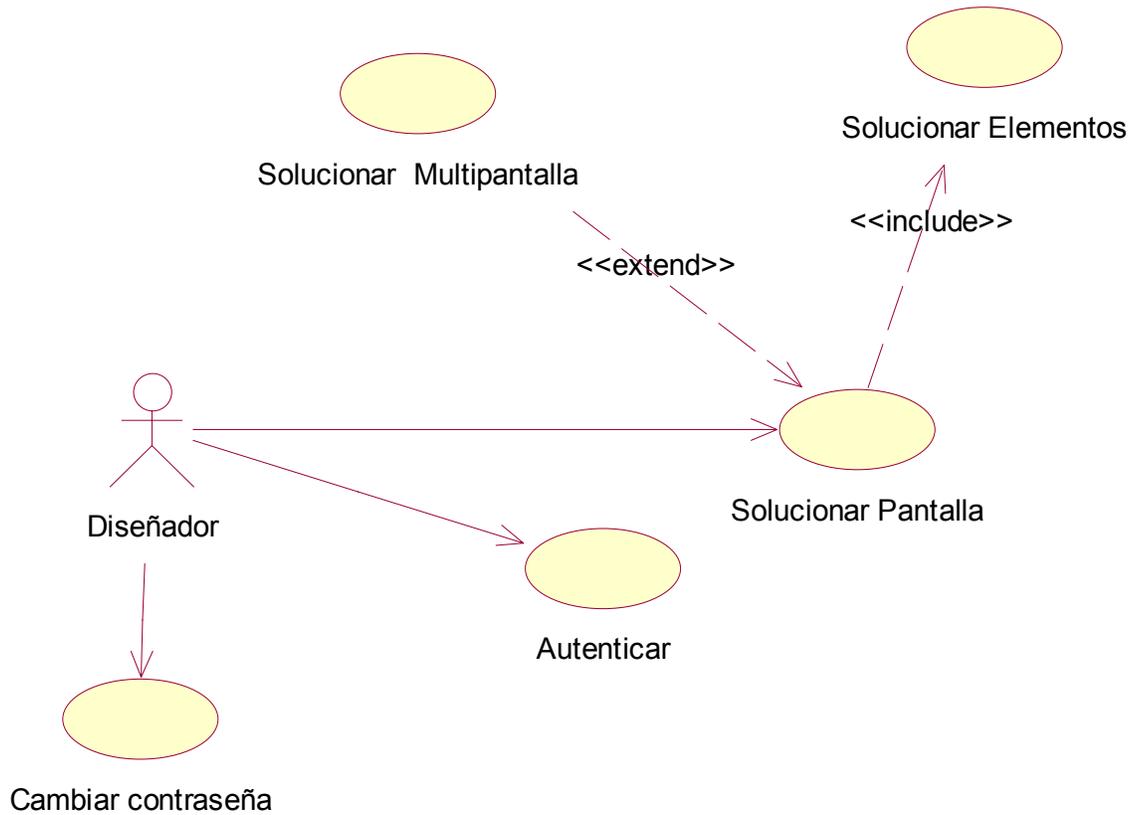


Figura 6 Diagrama de Casos de Uso Paquete Diseño

### 2.7.6 Paquete Usuario

En este paquete se muestran los casos de uso relacionados con el Usuario, el cual interactúa con el producto final, selecciona el idioma en que va a mostrarse la información y permite la impresión de imágenes.

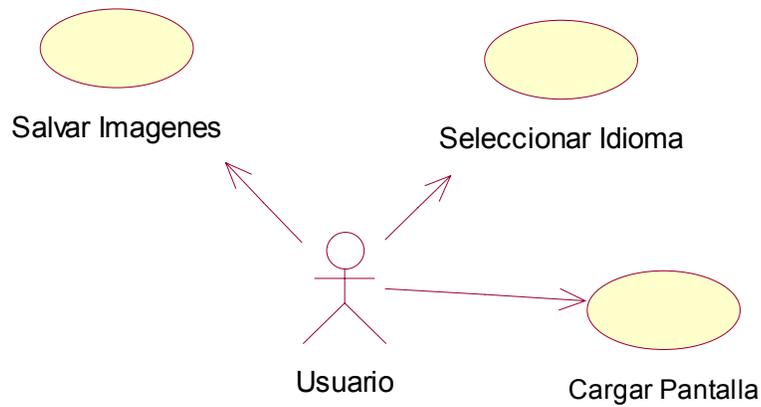


Figura 7 Diagrama de Casos de Uso Paquete Usuario

### 2.7.7 Descripción de los Casos de uso

Este caso de uso representa la entrada de un usuario al sistema.

Nombre del CU	Autenticar
Actores	Diseñador, Administrador
Propósito	Autenticarse en el sistema como usuario del mismo y obtener los privilegios asignados.
Resumen	El Actor solicita autenticarse en el sistema. El sistema muestra formulario donde el actor debe introducir el nombre de usuario y contraseña. El sistema valida los datos y en caso de que la contraseña sea válida para el nombre de usuario le otorga sus privilegios de usuario y termina el caso de uso, en caso contrario muestra mensaje indicando que ocurrió un error en la autenticación y muestra nuevamente el formulario de autenticación.
Referencias	R1
Precondiciones	
Poscondiciones	El usuario ha sido identificado y dado los privilegios según su configuración.
Requerimientos especiales	Por razones de seguridad solo se le permite al actor tres intentos fallidos en cada acción de autenticación, para evitar la posible intromisión de personal no autorizado.

**Tabla 2 Caso de Uso Autenticar**

El siguiente caso de uso contiene las acciones que el administrador realiza sobre los usuarios del sistema.

Nombre del CU	Administrar Usuario
Actor	Administrador
Propósito	Cambiar los privilegios del usuario, insertarlo o eliminarlo del sistema.
Resumen	El administrador solicita crear un nuevo registro de usuario .El sistema muestra formulario donde el actor introduce los datos del nuevo registro y le otorga sus privilegios. El sistema verifica si el usuario existe, de ser así devuelve un mensaje indicando la existencia del usuario y muestra nuevamente el formulario para que el Administrador corrija los datos, en caso contrario

	se inserta el registro y concluye el caso de uso. El administrador solicita modificar los privilegios de un usuario. El sistema muestra listado de los usuarios existentes. El actor selecciona uno y procede a modificar sus características. El sistema actualiza los datos del usuario. En caso de que el Administrador solicite borrar los datos del usuario, el sistema pide confirmación de la acción, si recibe la confirmación procede a eliminar el registro del usuario, en caso contrario finaliza el caso de uso.
Referencias	R2, R3,R4
Precondiciones	El actor ha ingresado al sistema y se ha autenticado como Administrador.
Poscondiciones	El registro de los usuarios queda actualizado.

**Tabla 3 Caso de Uso Administrar Usuario**

El presente caso de uso permite el cambio de contraseña de un usuario.

Nombre del CU	Cambiar Contraseña
Actor	Diseñador
Propósito	Cambiar su contraseña de autenticación.
Resumen	El actor solicita cambiar su contraseña de autenticación. El sistema muestra formulario con el nombre de usuario donde el actor debe introducir la contraseña anterior, la nueva y una confirmación de la nueva. En caso de que la contraseña anterior coincida con el nombre de usuario y la contraseña nueva sea igual a la confirmación de la nueva el sistema realiza los cambios correspondientes y termina el caso de uso, en caso contrario muestra un mensaje indicando que hubo un problema en el proceso y muestra nuevamente el formulario.
Referencias	R5
Precondiciones	El actor ha ingresado al sistema y se ha autenticado.
Poscondiciones	El usuario ha sido identificado y dado los privilegios según su configuración. Se ha cambiado la contraseña del usuario.
Requerimientos especiales	Por razones de seguridad solo se le permite al actor tres intentos fallidos en cada acción de cambio de contraseña, para evitar la posible intromisión de personal no autorizado.

**Tabla 4 Caso de Uso Cambiar Contraseña**

El siguiente caso de uso representa el trabajo con las plantillas, las cual servirá de guía para la futura creación de pantallas.

Nombre del CU	Solucionar Plantilla
Actor	Administrador
Propósito	Crear, Modificar y Eliminar Plantillas.
Resumen	El Administrador solicita realizar una acción sobre una plantilla. En caso de que la acción corresponda a crear una plantilla el sistema muestra formulario donde el actor inserta los datos correspondientes, en caso contrario muestra la lista de plantillas disponibles. Si no existen plantillas creadas anteriormente termina el caso de uso. Si el Administrador solicita eliminar una plantilla determinada el sistema pide confirmación de la acción, en caso de recibir la confirmación elimina los datos de la plantilla, en caso contrario finaliza el caso de uso. Si el actor desea modificar la plantilla el sistema habilita los iconos correspondientes para la inserción de elementos o la modificación de los existentes según sea el caso, guarda los datos de la plantilla y termina el caso de uso.
Referencias	R6, R7, R8, R9, R10, R11, R12; Caso de uso Solucionar Elementos (inlude).
Precondiciones	El actor ha ingresado al sistema y se ha autenticado como Administrador.
Poscondiciones	El usuario ha sido identificado y dado los privilegios según su configuración. Los datos de las plantillas quedan almacenados en l base de datos.
Requerimientos especiales	Garantizar cuestiones de seguridad, solo el administrador puede ejecutar este caso de uso.

**Tabla 5 Caso de Uso Solucionar Plantilla**

Este caso de uso contiene las actividades a desarrollar para manipular los elementos que componen las plantillas y pantallas del sistema.

Nombre del CU	Solucionar Elementos.
Actores	Diseñador y Administrador.
Propósito	Insertar, modificar y eliminar elementos.
Resumen	El Diseñador solicita insertar un elemento pulsando el icono correspondiente al tipo de elemento y luego dando clic en el área de trabajo. El sistema muestra formulario donde el actor introduce el camino asociado a dicho elemento. Si el camino determinado por el usuario no corresponde con el camino predeterminado de la carpeta de trabajo el sistema muestra mensaje informando de la imposibilidad de la acción, le muestra el camino de la carpeta de trabajo y le sugiere que mueva el elemento hacia ese sitio, tras esta acción concluye el caso de uso. En caso contrario el sistema muestra el elemento en el área de trabajo y captura los datos del mismo. Si el actor desea eliminar un elemento, el sistema pide confirmación de la acción, si recibe la confirmación elimina el elemento y actualiza los datos, luego concluye el caso de uso, en caso contrario termina el caso de uso. El Diseñador decide modificar los datos de un elemento e introduce los nuevos datos. El sistema modifica los datos del elemento y termina el caso de uso.
Referencias	R7, R8, R9, R14, R15, R31; Casos de uso: "Solucionar Plantilla" (include), "Solucionar Pantalla" (include).
Precondiciones	El actor ha ingresado al sistema y se ha autenticado. Existe un camino predeterminado para la carpeta de trabajo, la cual almacena los elementos utilizados en la aplicación.
Poscondiciones	Los datos correspondientes a los elementos quedan actualizados.
Requerimientos especiales	Garantizar que la velocidad y el tiempo de respuesta sea lo más óptimo posible.

Tabla 6 Caso de Uso Solucionar Elementos

El presente caso de uso representa el trabajo con las pantallas que conforman el producto final.

Nombre del CU	Solucionar Pantallas.
Actor	Diseñador.
Propósito	Crear, eliminar, modificar, cargar y guardar pantallas.
Resumen	<p>El actor decide realizar una acción sobre la pantalla. Si decide crear una pantalla nueva el sistema muestra la lista de plantillas disponibles. El actor selecciona una de ellas y el sistema despliega sus componentes en el área de trabajo. El diseñador comienza a sustituir los elementos de la plantilla por los propios de la pantalla. El sistema en cada caso muestra los nuevos elementos en el área de trabajo. El diseñador decide insertar elementos adicionales a la pantalla. El sistema inserta los elementos según la acción del actor. Una vez confeccionada la pantalla el diseñador decide guardarla. En caso de que se hayan sustituido todos los elementos de la plantilla por elementos de la pantalla el sistema muestra formulario donde el actor introduce los datos de la pantalla y la guarda en la base de datos, en caso contrario muestra mensaje correspondiente y espera la sustitución de los elementos restantes. El actor decide modificar los datos de una pantalla. Si existen pantallas disponibles el sistema muestra lista de pantallas existentes, en caso contrario muestra mensaje correspondiente y termina el caso de uso. El diseñador selecciona una pantalla, el sistema muestra los elementos de la pantalla en el área de trabajo, el actor modifica las propiedades de los elementos. El sistema guarda los cambios y termina el caso de uso.</p>
Referencias	R12, R13, R14, R14, R15, R6, R17, R18, R19, R20, R21, R22, R31; Casos de uso "Solucionar Elementos" (include) y "Solucionar Multipantallas" (extend).
Precondiciones	<p>Existe al menos un plantilla. El actor ha ingresado al sistema y se ha autenticado.</p>

Poscondiciones	Los datos de las pantallas quedan actualizados en la base de datos.
Requerimientos especiales	Garantizar que la velocidad y el tiempo de respuesta sea lo más óptimo posible.

**Tabla 7 Caso de Uso Solucionar Pantallas**

El siguiente caso de uso describe las actividades desarrolladas en las pantallas que poseen relación secuencial entre si, como es el caso de las pantallas de Lectura y Video, las cuales deben tener un orden determinado entre ellas. Por otra parte este caso de uso se rige por las mismas reglas del caso de uso “Solucionar Pantallas” por lo que se omitieron algunos detalles antes descritos en dicho caso de uso.

Nombre del CU	Solucionar Multipantallas
Actor	Diseñador
Propósito	Trabajo con pantallas secuenciales.
Resumen	El Diseñador solicita realizar una acción sobre una pantalla secuencial. Si desea crear una nueva pantalla secuencial el sistema muestra listado de plantillas donde el actor escoge una y sustituye los elementos por los de la pantalla, introduce los datos de esta última y se dispone a guardar la pantalla en la base de datos. El sistema muestra un listado con todas las pantallas que tengan el mismo tipo que la recién creada. El actor crea el orden de la secuencia entre las pantallas arrastrándolas y colocándolas en la posición que le corresponde dentro de la lista. El sistema captura el orden de las pantallas dentro de la secuencia y salva los datos de la pantalla en la base de datos y actualiza la posición de las demás. Si el actor desea eliminar una pantalla de la secuencia el sistema pide confirmación, en caso de recibirla elimina la pantalla de la lista y actualiza la secuencia de las restantes en la base de datos, en caso contrario termina el caso de uso.
Referencias	R12, R13, R14, R14, R15, R6, R17, R18, R19, R20, R21, R22, R31; caso de uso “Solucionar Pantallas” (extend).
Precondiciones	Existe al menos un plantilla. El actor ha ingresado al sistema y se ha

	autenticado.
Poscondiciones	Los datos de las pantallas quedan actualizados en la Base de Datos.
Requerimientos especiales	Garantizar que la velocidad y el tiempo de respuesta sea lo más óptimo posible.

**Tabla 8 Caso de Uso Solucionar Multipantallas**

Este caso de uso describe las acciones que toma el sistema cuando un usuario selecciona un idioma.

Nombre del CU	Seleccionar Idioma
Actor	Usuario
Propósito	Seleccionar un idioma específico.
Resumen	El caso de uso comienza cuando el usuario selecciona un idioma al pulsar sobre el vínculo correspondiente. El sistema verifica los elementos de la pantalla a cargar que posean el idioma seleccionado. El sistema muestra la pantalla correspondiente con los elementos que poseen el idioma seleccionado y los que no tienen idioma definido.
Referencias	R24, R25, R27
Precondiciones	Existe una base de datos con la información de las pantallas y sus elementos.
Poscondiciones	Queda almacenado el idioma seleccionado por el Usuario.
Requerimientos especiales	Garantizar la disponibilidad de las pantallas. Garantizar la integridad de los datos de los elementos y las pantallas.

**Tabla 9 Caso de Uso Seleccionar Idioma**

El presente caso de uso describe las actividades que desarrolla el sistema tras la solicitud de un usuario de imprimir una imagen.

Nombre del CU	Salvar Imágenes
Actor	Usuario
Propósito	Imprimir una imagen
Resumen	El usuario solicita imprimir una imagen. El sistema captura las imágenes disponibles en la pantalla desde donde se hizo la solicitud. El sistema muestra formulario donde el Usuario

	especifica si desea imprimir la imagen a color o el contorno solamente. El sistema imprime la imagen y termina el caso de uso.
Referencias	R30
Precondiciones	Existe una base de datos con la información de las pantallas y sus elementos.
Poscondiciones	La imagen queda impresa.
Requerimientos especiales	Garantizar la disponibilidad de una impresora conectada a la computadora donde se está ejecutando la aplicación, en su defecto existe una impresora disponible por la red.

**Tabla 10 Caso de Uso Salvar Imágenes**

Nombre del CU	Cargar Pantalla
Actor	Usuario
Propósito	Visualizar el contenido de una pantalla.
Resumen	El usuario solicita transitar de la pantalla actual hacia la siguiente. El sistema compara los elementos de la pantalla actual con los de la pantalla a cargar. Los elementos de la pantalla actual que difieran de los elementos de la pantalla siguiente serán removidos del escenario. El sistema analiza los botones típicos del tipo de pantalla que se desea cargar y los sustituye por sus homólogos de la pantalla siguiente forzándolos a adquirir las propiedades de éstos últimos (coordenadas, orden de profundidad, etc.). Se insertan los elementos de la pantalla siguiente que no guardaban similitud con los de la pantalla actual. El sistema organiza los elementos de la pantalla según su orden de profundidad y los muestra al Usuario.
Referencias	R20, R24, R25, R26, R7, R28, R31
Precondiciones	Existe una base de datos con la información de las pantallas y sus elementos.

**Tabla 11 Caso de Uso Cargar Pantalla**

## ***2.8 Conclusiones.***

Es de suma importancia todo lo referente a los casos de uso del negocio que se han dejado plasmados en este capítulo porque presentan una base para el entendimiento del sistema, gracias al modelo de casos de uso del sistema se logró proyectar una idea general de los requisitos necesarios para la implementación de la herramienta, se identifican sus actores y eventos asociados. Además describe la manera en que el sistema da solución al problema.

## CAPITULO 3 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA

### 3.1 Introducción

En este capítulo se describe la construcción de la solución propuesta mediante la realización de los diagramas de clases del diseño y diagrama de clases persistentes de las cuales se genera el modelo de datos. Se dividen los diagramas en paquetes con sus respectivas relaciones para el mejor entendimiento del sistema. Los aspectos relacionados con los principios de diseño se incluirán en el sitio web que se complementa con este documento.

### 3.2 Diagrama de Clases

Para modelar el diagrama de clases del diseño se ha decidido dividir estos en diferentes paquetes atendiendo a sus funcionalidades dentro del sistema.

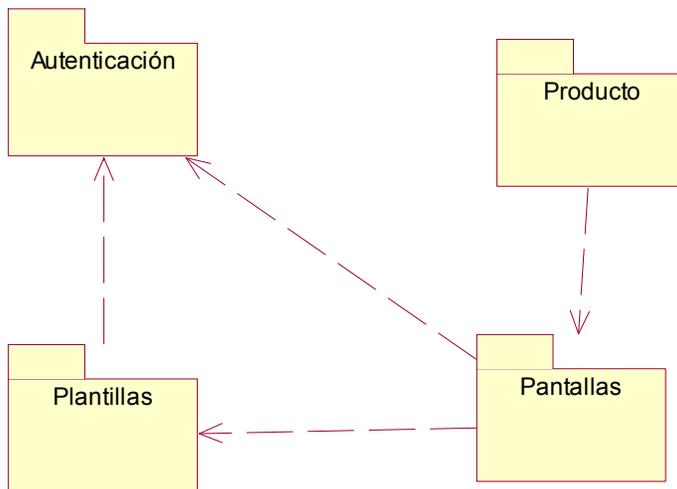


Figura 8 Diagrama de Paquetes del Diagrama de Clases

A continuación mostramos los diferentes diagramas de clases divididos por paquetes. Para su mejor comprensión se han ocultado algunos atributos y operaciones.

### 3.2.1 Paquete Autenticación

En este paquete se modela el proceso de autenticación y control de usuarios por parte del administrador.

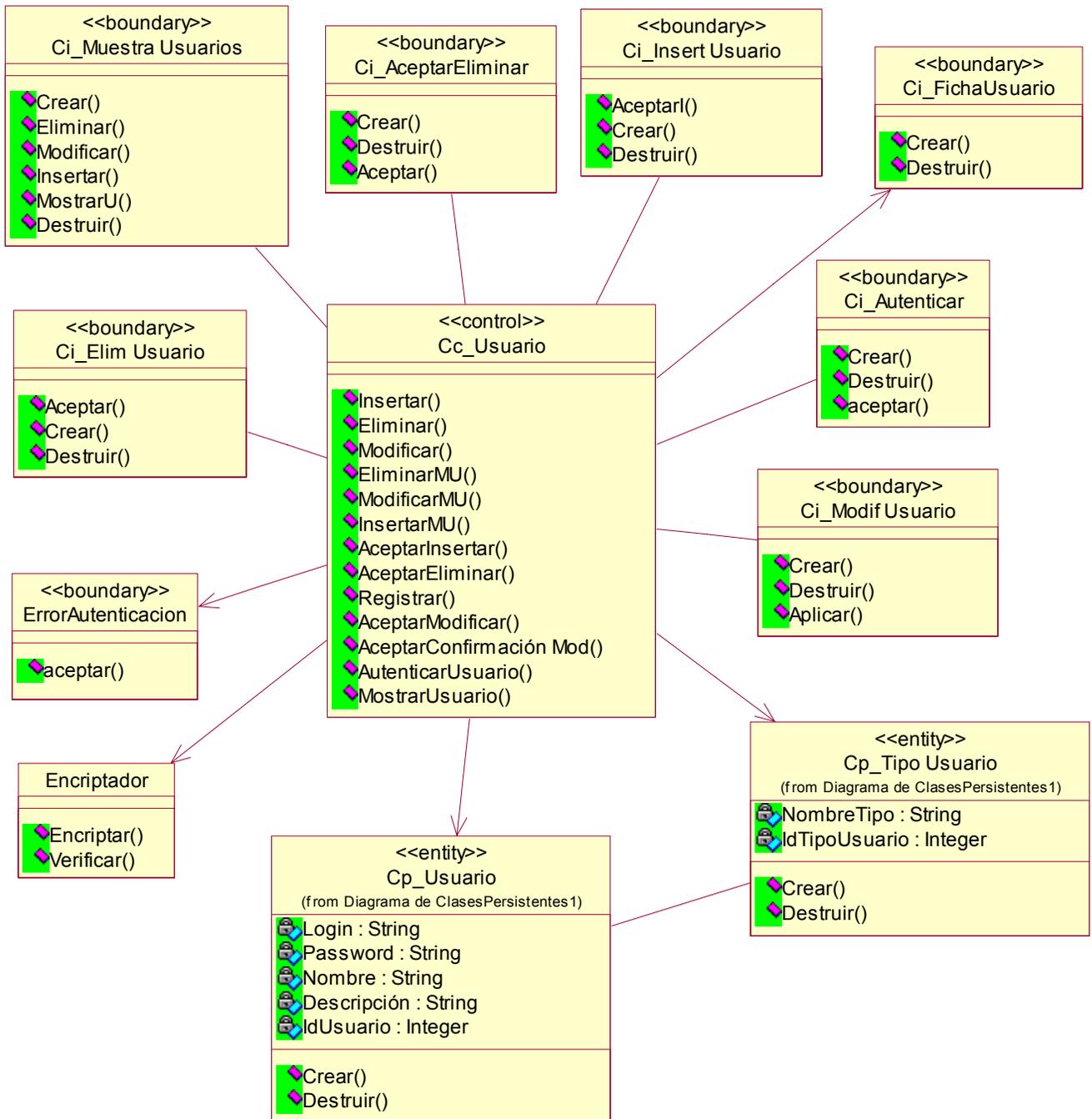


Figura 9 Diagrama de clases del paquete Autenticación

### 3.2.2 Paquete Plantillas

En este paquete se modela la gestión de las Plantillas que servirán de guía para la confección de las pantallas del producto.

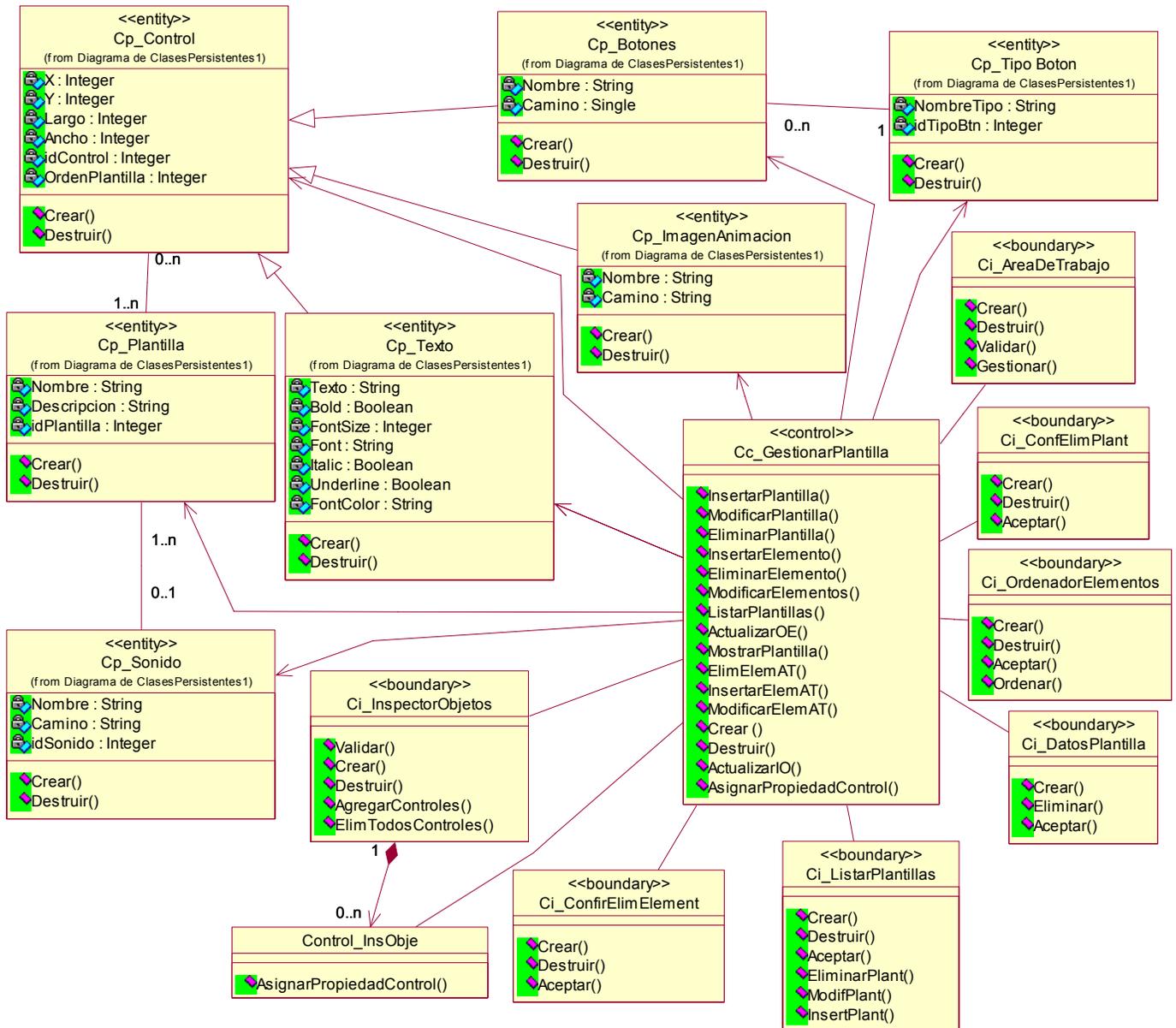


Figura 10 Diagrama de clases del paquete Plantillas

### 3.2.3 Paquete Pantallas

En el presente diagrama se modela la gestión de Pantallas a partir de las Plantillas definidas con anterioridad.

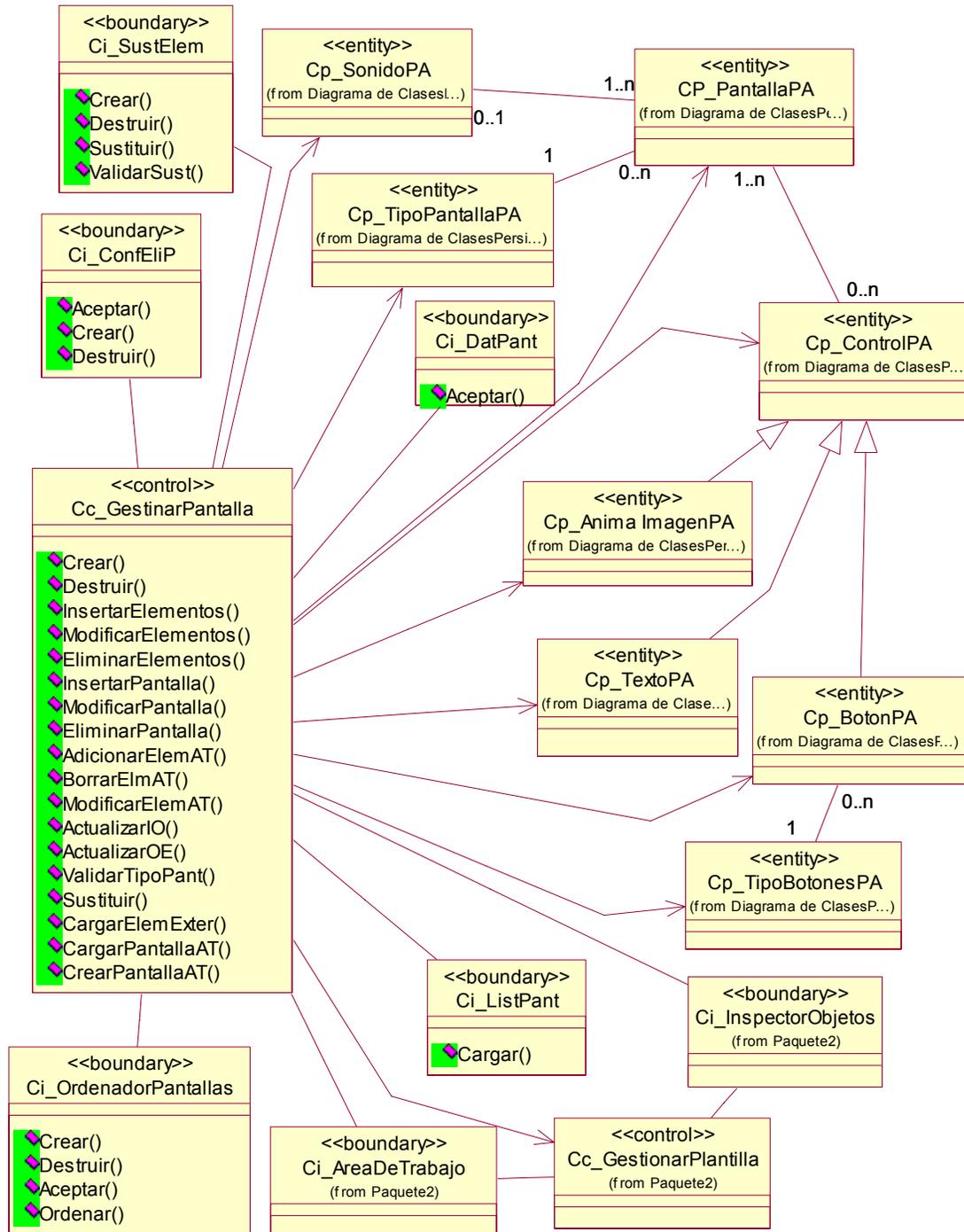


Figura 11 Diagrama de clases del paquete Pantallas

### 3.2.4 Paquete Producto

Este paquete contiene el modelo de clases correspondiente al producto final, el cual carga de la base de datos las pantallas del producto y las muestra al usuario.

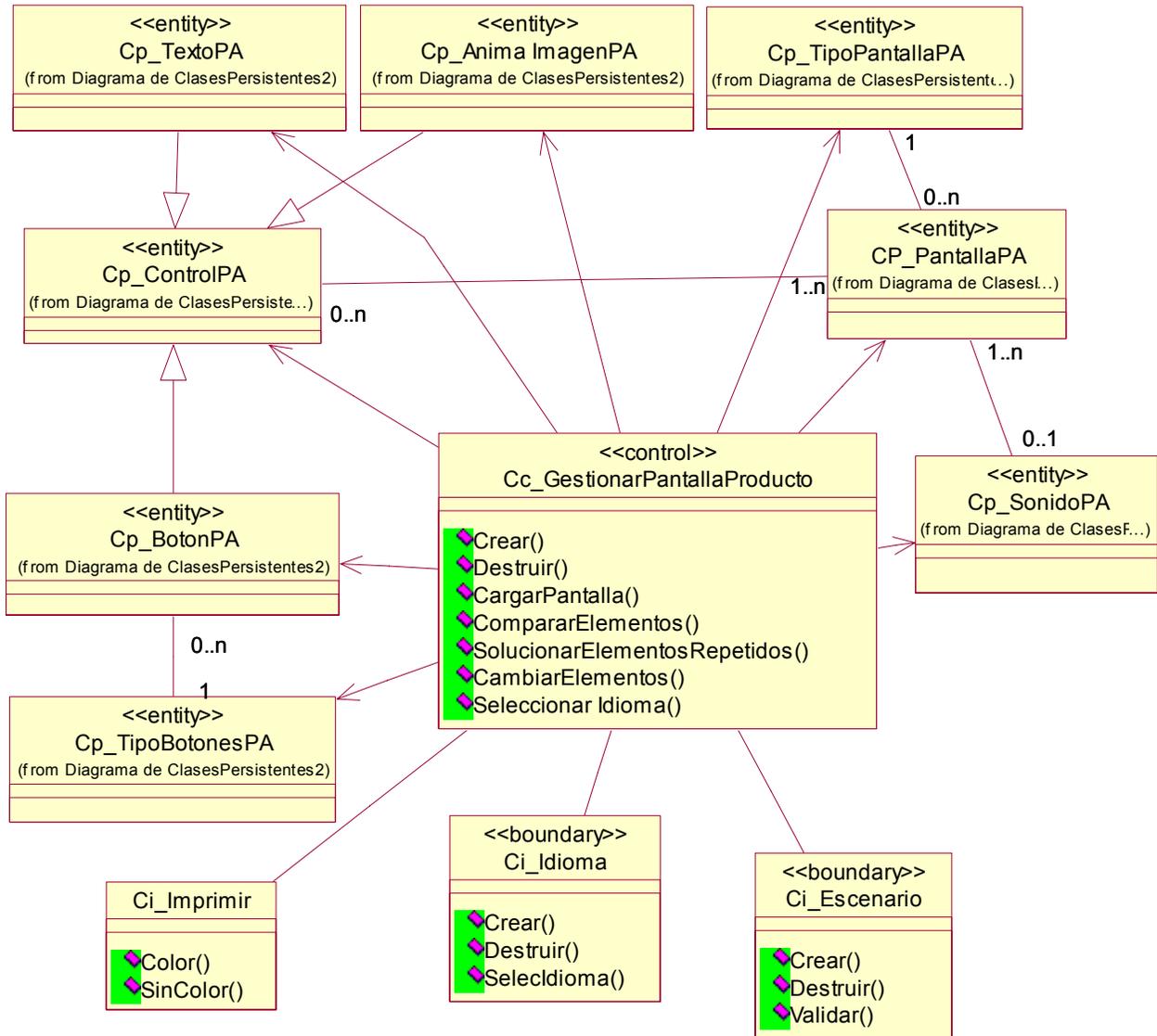


Figura 12 Diagrama de clases del paquete Producto

### 3.3 Diseño de la base de Datos.

En este epígrafe se describe el diseño de las bases de datos a partir de los modelos de datos generados de las clases persistentes correspondientes.

#### 3.3.1 Diagrama de Clases Persistentes Plantillas.

Las clases persistentes son las clases que contienen la información que perdura en el tiempo, como es el caso de la contenida en las bases de datos.

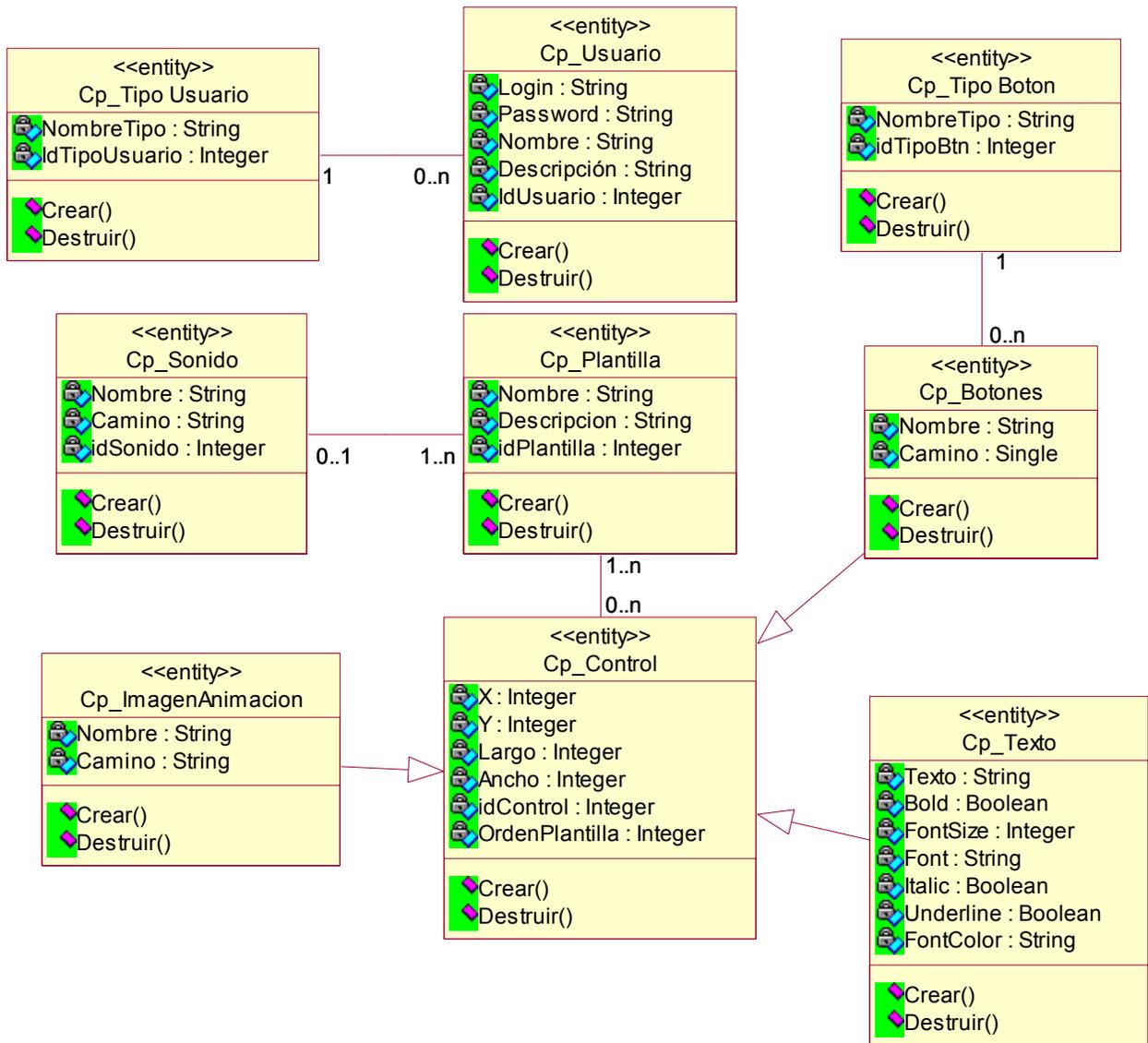


Figura 13 Diagrama de clases persistente Plantillas.

### 3.3.2 Diagrama de Clases Persistentes Pantallas.

A continuación se muestran las clases persistentes correspondientes a la base de datos Pantallas.

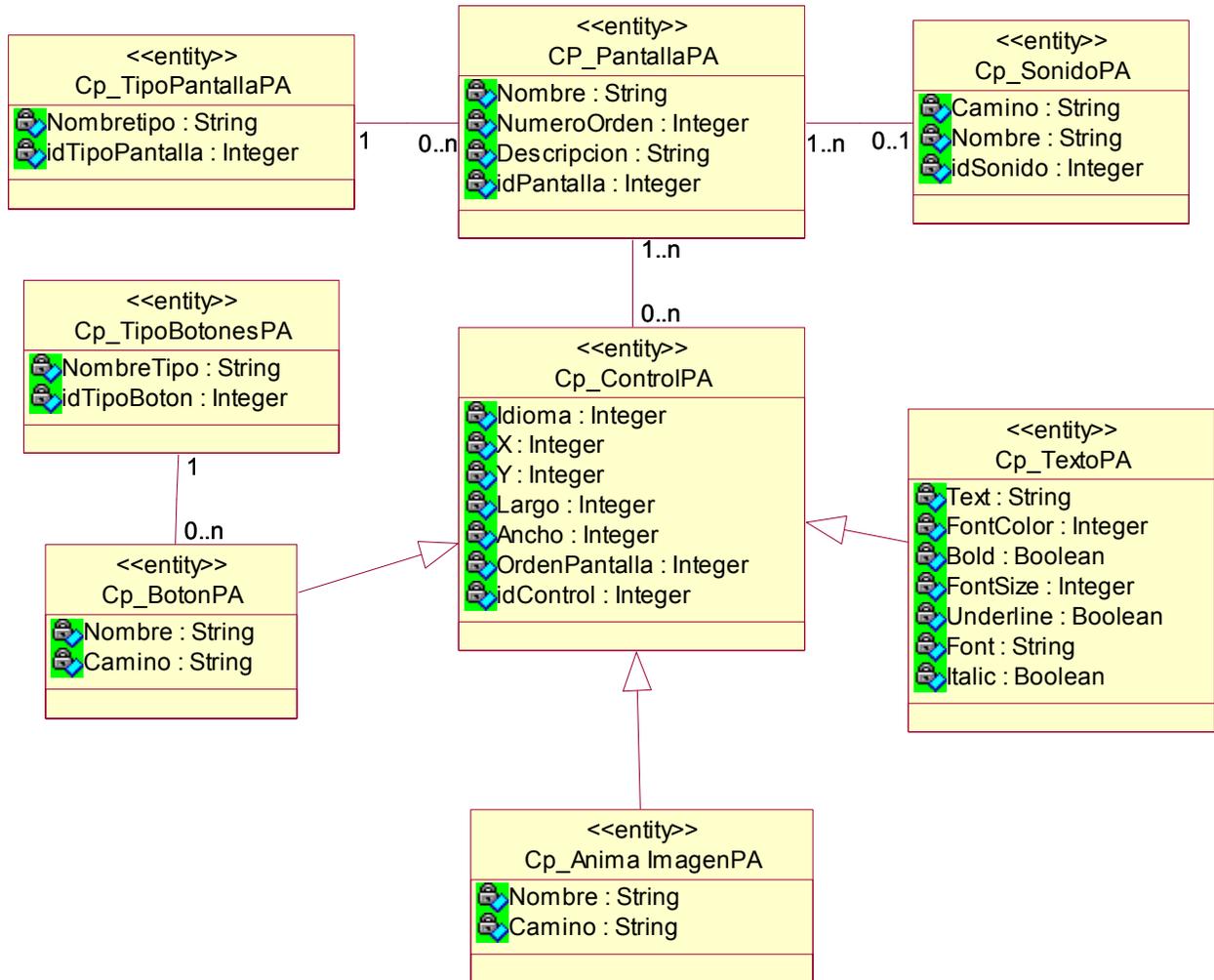


Figura 14 Diagrama de clases persistentes Pantallas.

### 3.3.3 Modelo de Datos Plantillas

Se presenta el modelo de datos correspondiente a la base de datos Plantilla, generado a partir de las clases persistentes.

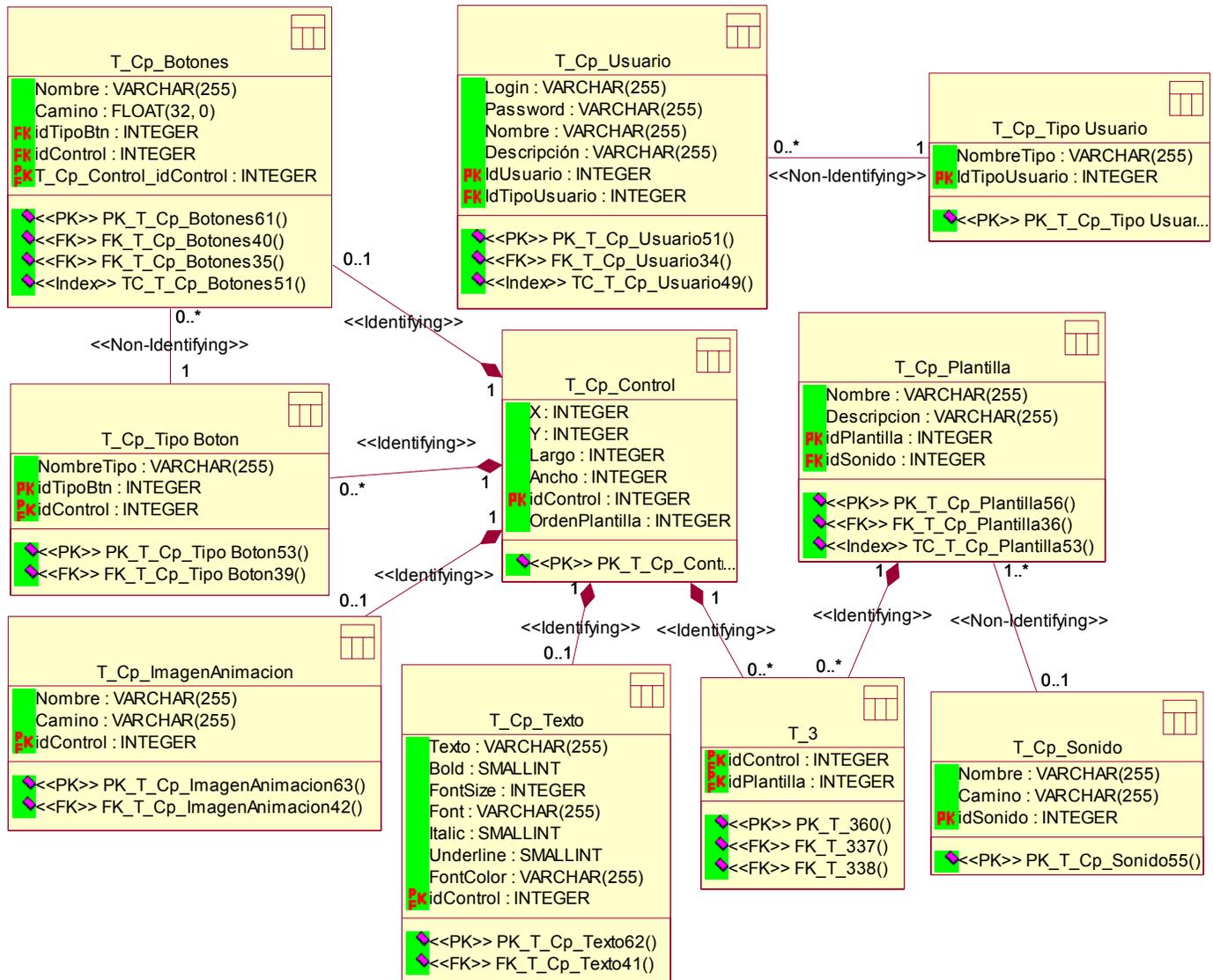


Figura 15 Modelo de datos Plantillas

### 3.3.4 Modelo de Datos Pantalla

Se presenta el modelo de datos correspondiente a la base de datos Pantallas generado a partir del diagrama de clases persistentes Plantillas.

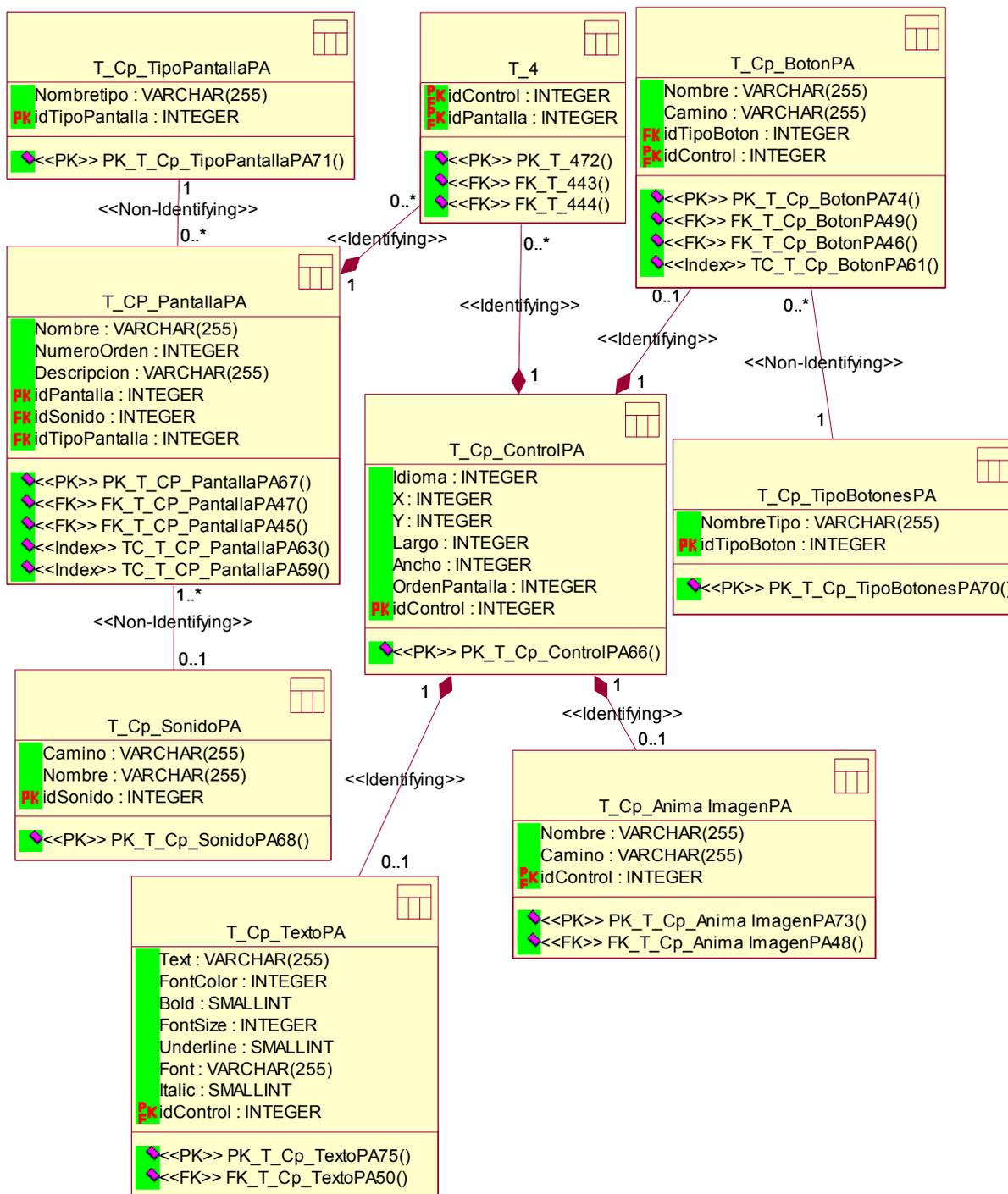


Figura 16 Modelo de datos Pantallas

### **3.4 Estándares de codificación**

1. Para lograr el entendimiento del modelo de la herramienta por el personal encargado de implementarla y demás consultantes en general se ha creado ciertos estándares de codificación:
2. Al asignarle un nombre a una clase interfaz éste comienza con las letras Ci\_ si es una clase controladora sería Cc\_ y el nombre de las clases entidades comienzan con Cp\_, las clases del modelo de datos comienzan con T\_ (recomendado por el sistema).
3. Si el nombre está compuesto por palabras independientes la primera letra de las mismas se escribe con mayúscula, por ejemplo EliminarPlantilla.
4. Los nombres de métodos y funciones comienzan con verbos, por ejemplo EliminarUsuario , Crear, Destruir.
5. Cuando el nombre es demasiado extenso se utilizan abreviaturas por ejemplo: ConfElimUsuario (ConfirmarEliminarUsuario).

### **3.5 Conclusiones**

Hasta este punto se ha reflejado las diferentes partes del proceso de diseño y modelación del sistema. Se ha conseguido reflejar las principales actividades que lo conforman lo cual brinda una información sumamente útil para la posterior implementación de la herramienta.

## Capítulo 4 Estudio de factibilidad

### 4.1 Introducción

En el presente capítulo se realiza un estudio de la factibilidad en la implantación de la herramienta “Nanatool” en la UCI mediante el análisis de los beneficios tangibles e intangibles. Para la realización de esta actividad se calculan los valores estimados del costo, tiempo de duración y cantidad de personas necesarias para su desarrollo.

### 4.2 Planificación

#### Entradas externas

<b>Entrada Externa (EI)</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de Elementos de datos</b>	<b>Clasificación</b>
Eliminar Usuario	1	5	Simple
Insertar Usuario	1	5	Simple
Autenticar	1	2	Simple
Modificar Usuario	1	4	Simple
Ordenar Elementos	2	2	Simple
Área de Trabajo	15	52	Complejo
Ordenar Pantalla	1	1	Simple
Datos Plantilla	1	3	Simple
Datos Pantalla	2	6	Medio
Inspector de Objetos	0	25	Medio
Sustituir elementos	0	1	Simple

Tabla 12 Entradas externas

### Salidas Externas

<b>Nombre</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de Elementos de datos</b>	<b>Clasificación</b>
Listado de Plantilla	6	23	Complejo
Listado Usuarios	1	3	Simple
Imprimir	0	2	Simple
Lista Pantallas	8	28	Complejo

Tabla 13 Salidas Externas

### Peticiones

<b>Nombre</b>	<b>Cantidad de ficheros</b>	<b>Cantidad de elementos de datos</b>	<b>Clasificación</b>
Ficha del Usuario	2	4	Simple
Idioma	0	1	Simple
Escenario	8	29	Complejo

Tabla 14 Peticiones

### Ficheros Internos

<b>Nombre del Fichero interno (ILF)</b>	<b>Cantidad de record</b>	<b>Cantidad de elementos de datos</b>	<b>Clasificación</b>
Usuario	1	5	Simple
Tipo Usuario	1	2	Simple
Tipo Botón	1	2	Simple
Texto	1	7	Simple

Sonido	1	3	Simple
Plantilla	1	3	Simple
Imagen_Animación	1	2	Simple
Control	1	6	Simple
Botones	1	2	Simple
Animación_ImagenPA	1	2	Simple
BotonPA	1	2	Simple
ControlPA	1	7	Simple
PantallaPA	1	4	Simple
SonidoPA	1	3	Simple
TextoPA	1	7	Simple
Tipo Botones	1	2	Simple
Tipo Pantalla	1	2	Simple

Tabla 15 Ficheros Internos

### Puntos de Función Desajustados

Elementos	Simple	X Peso	Medios	X Peso	Complejos	X Peso	Subtotal de puntos de función
Ficheros lógicos internos	17	(*7)	0	(*10)	0	(*15)	119
Petición	2	(*3)	0	(*4)	1	(*6)	12
Salidas Externas	2	(*4)	0	(*5)	2	(*7)	22
Entradas Externas	8	(*3)	2	(*4)	1	(*6)	38
<b>Total (UFP)</b>							<b>191</b>

Tabla 16 Puntos de Función Desajustada

### 4.3 Cálculo de las instrucciones fuentes, esfuerzo, tiempo de desarrollo, cantidad de hombres y costo.

Características	Valor
Puntos de función desajustados	191
Lenguaje	C#
Instrucciones fuentes por puntos de función	69
Instrucciones fuentes	2.768

Tabla 17 Características

#### Factores de Escala

$$SF = \sum_{i=1}^5 SF_i = PREC + FLEX + RESL + TEAM + PMAT = 10.14$$

#### Valores calibrados:

$$A=2.94; B=0.91; C=3.67; D=0.24$$

$$E = B + 0.01 * \sum SF_i = 0.91 + 0.01 * 10.14 = 1.0114$$

$$F = D + 0.2 * (E - B) = 0.24 + 0.2 * (1.0114 - 0.91) = 0.3$$

#### Cálculo de esfuerzo (PM)

$$PM = A * (MSLOC)^E * \Pi E_{mi} = 2.94 * (64.695)^{1.0114} * 1,7557 = 60 \text{ Hombres/Mes.}$$

#### Cálculo del tiempo de desarrollo.

$$TDEV = C * PM^F = 3.67 * (60)^{0.3} = 12.43 \approx 12 \text{ meses (Estimado)}$$

#### Cálculo de la cantidad de hombres.

$$CH = PM / TDEV = 60 / 12 = 5 \text{ hombres}$$

Como el equipo de trabajo está formado realmente por 6 personas, se recalcula el tiempo de desarrollo para la cantidad real de hombres.

$CH^* = 6$  hombres.

$TEDV = PM/CH^* = 60 / 6 = 10$  meses.

**Cálculo del costo.**

Asumiendo como salario promedio mensual (SP) \$225.00

$CHM = CH * SP = 6 * \$225.00 = \$ 1350.00$

$Costo = CHM * PM = \$1350 * 67.16 = \$ 81000.00$

**Cálculos**

<b>Cálculo de:</b>	<b>Valor</b>
Esfuerzo (PM: Hombres - mes)	60 Hombres/Mes
Tiempo de Desarrollo(meses)	10 Meses
Cantidad de Hombres	6
Costo	\$ 81000.00
Salario medio	\$ 225.00
PERS	1.26
RCPX	1.00
RUSE	1.00
PREX	1.12
FCIL	0.87
SCED	1.43
PDIF	1.00

**Tabla 18 Cálculos**

#### ***4.4 Beneficios tangibles e intangibles.***

Una vez implantada la herramienta propuesta es indudable que se obtendrá un considerable ahorro en el tiempo de producción de software multimedia. El desarrollo del proyecto proporciona innumerables beneficios ya que permite que se realicen las actividades de forma más eficiente y rápida y logra reducir la necesidad de contar con personal con amplios conocimientos en lenguajes de programación. Proporciona además un ambiente de trabajo personalizado y cómodo que facilita el trabajo.

#### ***4.5 Análisis de costos y beneficios***

El desarrollo del sistema se ha planificado para una duración de 10 meses, empleando un equipo de desarrollo compuesto por 6 trabajadores que implementará la herramienta en ese tiempo garantizando su completa funcionalidad. Considerando el salario medio de \$ 225.00 en moneda nacional, el proyecto induce un costo de \$ 81000.00 en concepto de salario. Analizando el alto valor práctico que nos proporciona el sistema propuesto y que los desarrolladores del mismo son principalmente alumnos de la Universidad que no perciben salario por dicha tarea, se concluye que es factible la implantación de la herramienta en la Universidad de las Ciencias Informáticas .

#### ***4.6 Conclusiones***

Este capítulo es muy importante para el proceso de desarrollo de software porque es necesario conocer los recursos requeridos, las tareas a ejecutar, las referencias a tener en cuenta y el esfuerzo a emplear si se quiere crear un producto de calidad. Con la culminación de esta parte se tiene conocimiento de los beneficios tangibles e intangibles que proporcionará la herramienta "Nanatool" tras su implantación.

## **Conclusiones**

Con la realización de este trabajo se logró realizar un estudio profundo de la gestión de planificación y control del trabajo multimedia en la Universidad de las Ciencias Informáticas (UCI) pudiendo ubicar sus principales deficiencias y necesidades. Se logró crear una base sólida para la confección de la herramienta propuesta y se diseñó dos bases de datos que resuelven el control y almacenamiento de la información referente a los productos multimedia para niños. Se realizó un análisis general de las tendencias y tecnologías actuales en cuanto a plataformas, lenguajes y metodologías que conllevó a que el diseño del sistema fuese consecuente con su posterior implementación. Todo esto permitió alcanzar el cumplimiento de los objetivos planteados al comienzo del presente trabajo y sobre todo el cumplimiento de la principal tarea propuesta, la modelación y diseño de la herramienta capaz de satisfacer las necesidades del proceso de confección de productos multimedia educativos.

El nivel alcanzado por la educación en Cuba es un mérito reconocido a escala internacional, los software que se desarrollan en la Universidad de las Ciencias Informáticas son representación de la misma. La inclusión de productos educativos aplicando las nuevas tecnologías le incorpora un nuevo valor a la guardería, un reconocimiento a la institución y prestigio al país.

## Recomendaciones

Recomendamos continuar trabajando en la implementación del sistema, pues en esta primera etapa sólo se ha llegado a la modelación y diseño de la herramienta “Nanatool”. En próximas etapas debe llevarse a cabo la implementación íntegra de la misma para lograr brindar un servicio eficiente en la confección de aplicaciones multimedia para niños desarrolladas en la UCI.

Debido al creciente auge del uso de la multimedia en la educación a diario surgen nuevas propuestas y necesidades en esta rama, lo cual nos empuja a la necesidad de ampliar las prestaciones de la herramienta para así trascender los límites del proyecto guardería y poder resolver problemas similares en otros proyectos educativos, por lo tanto se recomienda además dotar a la aplicación de una mayor flexibilidad para que se pueda ajustar a otros tipos de proyectos multimedia.

## Bibliografía

1. .Wikipedia *Multimedia* <http://es.wikipedia.org/wiki/Multimedia> (21/3/2005).
2. Álvaro Gómez, Celia. *Hipertexto Conceptos*  
<http://rayuela.uc3m.es/~calvaro/utiliza.html> (21/3/2005).
3. Estrada Cingualves, Reynaldo . *Revisión Bibliográfica para La Confección De La Multimedia Cgi/Rudg-3,2* <http://www.monografias.com> (1/4/2005).
4. UC. *Introducción a la MULTIMEDIA, Sistemas MULTIMEDIA.*  
<http://www.uco.es/investiga/grupos/eatco/automatica/sMULTIMEDIA/Introduccion%20a%20la%20MULTIMEDIA.zip> (4/4/2005).
5. Díaz Muriel, Dionisio. Valverde Berrocoso, Jesús. López Meneses, Eloy  
*Congreso Internacional de Tecnología, Educación y Desarrollo sostenible*  
<http://www.edutec.es/edutec01/edutec/comunic/TSE03.html> (4/4/2005).
6. Tella Llop, JM. *Multimedia* [http://www.multingles.net/docs/alezito\\_cd.htm](http://www.multingles.net/docs/alezito_cd.htm)  
(6/4/2005).
7. UA. *Informática educativa,*  
[http://www.ual.es/Universidad/Depar/LengComp/programas/MAG/T2-3-4\\_04.pdf](http://www.ual.es/Universidad/Depar/LengComp/programas/MAG/T2-3-4_04.pdf)  
(6/4/2005)
8. Zambrano Rodriguez, Douglas . “*Multimedias*” <http://www.monografias.com>  
(8/5/2005).
9. González Seco, José Antonio .*El lenguaje de programación C#* .  
<http://www.josanguapo.com/> (8/5/2005).
10. Ponello, Damian. *Introducción a Microsoft .NET*  
[http://www.dcp.com.ar/intro\\_net.htm](http://www.dcp.com.ar/intro_net.htm) (9/5/2005).
11. Archer, Tom *A Fondo C #*.Primera edición (en español). Aravaca, Madrid 2001.

12 Schildt, Herbert *C++ Guía de Autoenseñanza*. Aravaca, Madrid 1995.

13 Jacobson, Ivar ; Booch Grande ; Rumbaugh James *El Proceso Unificado de Desarrollo de Software* . Madrid, España 1999.

14 Martin, Robert; Nyquist, John R. "Director 8"

15 Alvarado, Pedro; Laitano, Rommell *Modelo de base de datos con ER/studio*  
<http://www.monografias.com/trabajos14/modelodebase/modelodebase.shtml>  
(15/5/2005).

## Glosario de términos

- ✚ **Guardería:** Lugar o institución donde se agrupan niños pequeños para desarrollar actividades infantiles. Ejemplo: Círculos Infantiles.
  
- ✚ **TIC:** Tecnologías de la Información y las Comunicaciones, también se puede
- ✚ presentar como: NTIC, Nuevas Tecnologías en la Información y las Comunicaciones.
  
- ✚ **Soporte Multimedia :** Medios para almacenamiento de imágenes, sonido, videos, animaciones, texto etc.
  
- ✚ **CPU:** Elemento electrónico del ordenador que proporciona capacidad de cálculo y control.
  
- ✚ **Ícono:** Es una pequeña imagen, normalmente un símbolo, utilizado para representar gráficamente un programa, un fichero, o una función en la pantalla del ordenador.
  
- ✚ **COM :** Tecnología para construir sistemas software a partir de la interacción de objetos (componentes). El modelo de programación COM está basado en la distribución de código de clases en componentes binarios.
  
- ✚ **Interface :** Es el punto de comunicación entre el ordenador y el usuario por donde se transmite la información. Una ventana de autenticación donde se inserta el usuario y la contraseña es un ejemplo de *Interfaz*.

