

**UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS**

**Facultad 5 – Entornos Virtuales e Informática Industrial**



**Herramienta de Autor de Contenido para el  
Sistema EMEDIA**

Trabajo de Diploma para optar por el título de  
Ingeniero Informático

**AUTORES: Isabel Birmania Santell Tabera**

**Yilver Roberto Arcaya Durán**

**TUTOR: Ing. Irina Elena Argota Vega**

**ASESOR: Ing. Belkis Grissel Gonzalez Rodriguez**

**Ciudad de la Habana, 2 de Julio del 2008.  
Año 50 de la Revolución**

## DECLARACIÓN DE AUTORÍA

Declaro que soy el único autor de este trabajo y autorizo al <nombre  
área> de la Universidad de las Ciencias Informáticas a hacer uso del  
mismo en su beneficio.

Para que así conste firmo la presente a los \_\_\_\_ días del mes de  
\_\_\_\_\_ del año \_\_\_\_\_.

"[Insertar nombre(s) de autor(es)]"

"[Insertar nombre(s) de tutor(es)]"

\_\_\_\_\_

\_\_\_\_\_

*"A veces sentimos que lo que hacemos es tan solo una gota en el mar, pero el mar sería menos si le faltara una gota."*

*Agnes Gonsha Bojaxhiu*

DATOS DE CONTACTO

**Tutora:** Ing. Iriana Elena Argota Vega

Profesor Instructor de Matemática UCI- Departamento de Ciencias Básicas

Analista del proyecto SCADA

Dirección: Universidad de las Ciencias Informáticas (UCI) . Edif: 28 Apto: 102

E-mail: [iargota@uci.cu](mailto:iargota@uci.cu)

*Agradecer en pocas palabras todo lo que uno siente hacia tantas personas en un momento como este se convierte en algo bastante difícil, pero apelaré a la sabiduría de mi memoria y les haré llegar mi más sincero agradecimiento, a todos los que de una forma u otra aportaron su pedacito de voluntad para hacer realidad mi sueño.*

*A Fidel y a la Revolución por darme la oportunidad de ser un profesional sencillo.*

*A Isabel mi compañera de tesis, mi hermana que supo confiar en mí y hacer de estos años una verdadera escuela para mí.*

*A mi tutora Irina, por su ayuda excepcional, por ser compañera, amiga y hacerme reír tanto.*

*A Febe y Jaime por siempre estar ahí, y brindarnos su apoyo incondicional.*

*A mis amigas, mis hermanas de todos los tiempos... Grethel, Mayuli, Yade, Nani, Zile, Daimi, Isabel, Yuly, Elizabeth (pantera), Yaly, Odaisis, Brenda, a todas por demostrarme que aún sin lazos sanguíneos, se puede hacer una familia.*

*A las amigas de mi mamá, Mercedes (mi otra mamá), Nani, Haydeé, Alicia. A Frank, a la mamá de Isa por la ayuda. A mis compañeros, y a los tres mosqueteros Campo, Carlos y Prevot...ustedes saben.*

*A toda mi familia por quererme tanto y por la confianza que depositaron en mí.*

*A Dios y la Virgen, por demostrarme que con fe todo se logra.*

*Gracias a todos los que no están, y que han formado parte de esta pequeña vida que me precede.*

*Yil*

*A la Revolución y a Fidel por darme la oportunidad de hacer mis sueños realidad.*

*A mis padres quienes con tanto amor me han guiado por el mejor de los caminos y han depositado toda su confianza en mí.*

*A mi querido hermano, amigo y compañero de tesis Yilver Roberto por estar conmigo en las buenas y en las malas, y por ser mi mejor consejero.*

*A mi tutora Irina por su ayuda excepcional y al mismo tiempo ser amiga.*

*A Yuly por ser mi mejor amiga por brindarme su amistad sincera en estos 5 años.*

*A Eli por sus sabios consejos y por darme su apoyo todo el tiempo.*

*A Haydee, Yaly, Belkis, Yayi, Oda, Brenda, Say, Lisy por siempre estar ahí y acompañarme con su amistad.*

*A Infodanz por ser parte de mi y hacer mis momentos difíciles más gratos.*

*A Heiler por su amor, cariño, paciencia y por apoyarme en todos estos años.*

*A mi primo Jaime y a Febe por brindarme su ayuda incondicional.*

*A toda mi familia por quererme tanto y por la confianza que depositaron en mí.*

*A Carmita, a Oda la mamá de Yil, a Silvia Rita y a mi profe Zenaida por ser madres consejeras.*

*A mis compañeros de grupo, a todos los que me han acompañado en estos 5 años.*

*Ya Dios por estar conmigo siempre, por su fidelidad y por amarme tanto, tanto.*

*A mi querida madre, quien con tanto amor y paciencia infinita ha sabido confiar en mí, siendo mi luz y protagonista de cada momento trascendental de mi vida.*

*A mi papá, por haber sido guía, y profesarme con su amor incondicional los principios que hoy rigen mi personalidad.*

*A mis abuelos, que con todo su amor han mancomunado fuerzas para llegar hasta aquí, y hacer suya esta felicidad.*

*Yilver*

*A mis padres por su amor y dedicación por darme fuerza en los momentos difíciles y porque hoy le debo a ellos todo lo que soy.*

*A mi hermanita querida por darme tanto aliento para seguir adelante.*

*A mi abuela por ser como una madre para mí, por su apoyo y por siempre tenerme presente en sus oraciones.*

*Isabel*

### Resumen

El vertiginoso desarrollo de la Ciencia y la Tecnología ha propiciado que el mundo creativo actual se mantenga en constante cambio, de ahí que la introducción de las Tecnologías de la Información y las Comunicaciones (TIC), cimentadas en la gradación ascendente de productos de software educativos elaborados a nivel internacional, cobren conocimiento en las innumerables modificaciones de la concepción del proceso de enseñanza-aprendizaje. La Universidad de las Ciencias Informáticas (UCI) se ha insertado en este vasto y competitivo mercado, evidenciando como elemento fundamental de su creación, la producción de software educativo como vía de integración del proceso de enseñanza-aprendizaje con las TIC.

Una de las grandes limitantes en el desarrollo de software educativo, lo ha constituido las ventajas o beneficios en el uso de las Herramientas de Autor existentes, principalmente propietarias; las cuales no satisfacen todas las necesidades de la producción nacional, y aún más en la UCI.

*“Las herramientas de autor son herramientas de desarrollo de software que posibilitan a diseñadores instructivos, educadores, maestros y aprendices diseñar un curso multimedia interactivo, y ambientes de aprendizaje en hipermedia sin el conocimiento de lenguajes de programación.” (Dabahg, 2001)*

Con el objetivo de trabajar sobre la solución de esta limitante. Se ha hecho el análisis de una selección de Herramientas de Autor existentes en el mercado y soportadas en software libre, para presentar una propuesta de diseño de una Herramienta de Autor de Contenido en la realización de guiones de contenido; que le permita al Experto (especialista en contenidos) la organización de los procesos educativos de este tipo de aplicaciones.

### Palabras Clave

Herramientas de Autor (HC), Guión de Contenido, Elementos de Contenido, Objetos de Aprendizaje, Elementos Contenedores, Plantillas.



---

**ÍNDICE**

<b>INTRODUCCIÓN:</b> .....	<b>1</b>
<b>CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA</b> .....	<b>6</b>
INTRODUCCIÓN.....	6
1.1 PROCESO DE ENSEÑANZA APRENDIZAJE.....	6
1.2 LA INFORMÁTICA Y EL SOFTWARE AL SERVICIO DE LA EDUCACIÓN. ....	6
1.3 CONCEPTUALIZACIÓN DE HERRAMIENTAS DE AUTOR. ....	7
1.4 CARACTERÍSTICAS HERRAMIENTAS DE AUTOR. ....	9
1.5 EL GUIÓN. ....	11
1.6 OBJETOS DE APRENDIZAJE .....	12
1.7 LOS CONTENIDOS. ....	13
1.8 HERRAMIENTAS DE AUTOR EXISTENTES EN EL MERCADO. ....	14
1.9 METODOLOGÍAS DE SOFTWARE.....	16
1.9.1 Metodologías ágiles o ligeras.....	17
1.9.2 Metodologías pesadas .....	17
1.9.2.1 Metodología Orientada a Objetos: RUP.....	17
1.10 SISTEMA OPERATIVO GNU/LINUX. ....	19
1.10.1 Distribución de Linux: Debian GNU/Linux.....	19
1.11 LENGUAJE DE MODELADO UML .....	20
1.12 PLATAFORMA DE DESARROLLO DE SOFTWARE .....	22
1.13 BIBLIOTECAS GRÁFICAS.....	24
1.13.1 GTK. ....	24
1.13.2 Librería Gráfica: QT .....	24
1.13.3 QT- Designer.....	25
<b>CAPÍTULO 2: CARACTERÍSTICAS DEL SISTEMA</b> .....	<b>26</b>
INTRODUCCIÓN.....	26
2.1 ESTADO ACTUAL DEL NEGOCIO .....	26
2.1.1 Sistema Emedia.....	26
2.1.2 Herramienta Autor de Contenido (HAC).....	27
2.2 MODELO DE DOMINIO .....	28
2.3 ESPECIFICACIÓN DE LOS REQUISITOS .....	30
2.3.1 Requisitos Funcionales .....	31
2.3.2 Requisitos No Funcionales .....	32
2.4 INTERFAZ DE COMUNICACIÓN CON EL USUARIO .....	34
2.5 MODELOS DE CASOS DE USO DEL SISTEMA. ....	35
2.5.1 Diagrama de Casos de Uso del Sistema .....	35
2.5.2 Descripción de los Casos de Usos .....	36
1. Caso de uso crear un nuevo guión de contenido. ....	36
2. Caso de uso Gestionar Guión de Contenido.....	38
3. Caso de Uso Gestionar Taxonomía del Guión de Contenido.....	40
4. Caso de uso Gestionar Plantilla de OA.....	43
5. Caso de uso Gestionar Plantilla de Niveles Taxonómicos (Elemento Contenedor).....	44
6. Caso de uso Crear Paquete Modelo Estudiante. ....	46
7. Caso de Uso Gestionar Modelo Pedagógico. ....	47
<b>CAPÍTULO 3: SOLUCIÓN PROPUESTA</b> .....	<b>51</b>
INTRODUCCIÓN.....	51
3.1 VISTA GESTIÓN MODELO.....	51
3.2 DISEÑO ARQUITECTÓNICO .....	51
3.3 PATRÓN MODELO-VISTA-CONTROLADOR.....	52
3.4 PATRONES DE DISEÑO .....	53
3.4.1 Patrones GOF.....	53
3.4.1.1 Fábrica Pura.....	53
3.5 DIAGRAMAS DE CLASES DEL ANÁLISIS.....	54

3.6 DIAGRAMAS DE CLASES DEL DISEÑO .....	58
3.6.1 <i>Descripción de clases del Diseño</i> .....	58
3.6.2 <i>Diagramas de clases del Diseño</i> .....	73
<b>CONCLUSIONES GENERALES .....</b>	<b>61</b>
<b>RECOMENDACIONES .....</b>	<b>62</b>
<b>REFERENCIAS BIBLIOGRÁFICAS .....</b>	<b>63</b>
<b>ANEXOS .....</b>	<b>65</b>
<b>GLOSARIO .....</b>	<b>83</b>

### **Introducción:**

*“Los seres vivos siempre han necesitado comunicarse, y los humanos han desarrollado y perfeccionado las formas de hacerlo. Inventaron el lenguaje, aprendieron a escribirlo, lograron la forma de reproducir lo escrito, aprendieron a fijar las imágenes y a darles movimiento, crearon música para celebrar sus fiestas y funerales”* (Isasa, 2001); y todos estos procesos se han condicionado y desarrollado aún más con la aparición e inserción en el mundo actual, de las Tecnologías de la Información y las Comunicaciones (TIC).

El avance en la apropiación de los conocimientos por el individuo de la sociedad actual; donde la tecnología juega un papel esencial, ha generado necesidades y exigencias que abogan por un cambio substancial en las características de los sistemas educacionales actuales. Estas modificaciones deben posibilitar, ver más de lo que está reflejado en un libro de texto, en la palabras de un profesor o simplemente en lo que se ve o escucha a rededor.

Existen diversas formas de aplicación de las nuevas tecnologías, una de las más extendidas y utilizadas por su adecuación a las teorías de la enseñanza son los sistemas conocidos como Software Educativo. *“(...) es un “programa de computadora para educación”; “creado con la finalidad específica de ser utilizados como medio didáctico, es decir para facilitar los procesos de enseñanza aprendizaje.”* (Ceja, 2000).

El aumento de la demanda y la diversidad en la formación, ha propiciado una mayor solicitud y evolución en la producción de cursos. Para dar respuesta a esta realidad se han desarrollado aplicaciones informáticas más fáciles de utilizar por el usuario y los profesores. El proceso de producción de un curso es una tarea difícil que requiere mucho esfuerzo por parte de los especialistas en contenido, o sea maestros e instructores, de ahí que estuviera mucho tiempo fuera del alcance de la mayoría de estos. Esta complejidad pudo ser reducida apoyándose en herramientas informáticas apropiadas, que automatizaron una parte o todo el proceso de producción de cursos, surgiendo las Herramientas de Autor. (MSc. Jose L. MOntero, 2005)

*“las Herramientas de Autor son aplicaciones de software orientadas a la creación de materiales educativos con las que el usuario (maestros e instructores), sin necesidad de conocer el lenguaje de programación, pueda crear módulos que integren todos los componentes de un curso.”* (Herramientas, 2000)

El insuficiente número de software educativo libre pasa por múltiples factores, pero en esencia el problema es complejo por la gran demanda del área en tipos de productos, áreas del conocimiento a abordar, niveles escolares y enfoques didácticos. Además este tipo de programas necesita de los esfuerzos de un equipo multidisciplinario difícil de encontrar espontáneamente. La complejidad informática en ocasiones es poco atractiva para los programadores expertos, y no existe en el mundo herramientas de autor libres que ponga en las manos de los sectores interesados las soluciones tecnológicas. (Rodríguez, 2007)

En la actualidad existen decenas de “**sistemas de autor**”, que generalmente utilizan estructuras especializadas en la solución de las tareas clásicas que permiten a los desarrolladores el diseño de la multimedia interactiva. El uso de estas herramientas hoy día se ha visto afectado por las leyes de propiedad, que impiden una correcta utilización de estos sistemas en la creación de materiales educativos. En una encuesta realizada a la Comunidad de Investigación de Educación a Distancia se reflejó que **Adobe Flash** (hasta 2005 **Macromedia Flash**) es el dueño de las Herramientas de Autor. (Pulichino, 2005 )

Debido a las características propias de las Herramientas de Autor para la creación de cursos educativos; la comunidad de desarrolladores de aplicaciones educativas en la UCI, ha insertado su utilización en *el desarrollo de aplicaciones multimedia educativas* (Jimmy, 2001); ya sea para un mejor desarrollo del proceso de formación de sus estudiantes, como para las diferentes enseñanzas y otras instituciones tanto nacionales como extranjeras. Estas herramientas permiten la producción tanto en software libre como propietario.

Precisamente el presente trabajo encamina su solución en la utilización de software libre por las necesidades de producción en la Universidad. El déficit de Herramientas que posibiliten la integración de funcionalidades para la creación de los cursos en un solo paquete; herramientas que permitan su distribución en CD-ROM, que prescindan de un servidor para montar la aplicación; y que permitan la creación de cursos educativos, sin responder a las leyes de propiedad en su utilización. Constituyen necesidades potenciales actuales, que impiden una adecuada producción de los materiales educativos en el centro.

Durante el proceso de desarrollo de software para el aprendizaje en la Universidad, se han detectado un conjunto de irregularidades que atentan contra la calidad de las aplicaciones resultantes:

1. La utilización de herramientas de propósito general, no orientadas al conjunto de especificidades que se requieren para lograr mayor calidad en el desarrollo de aplicaciones.
2. Es necesario aumentar la participación activa de los tutores, profesores en el Proceso de Desarrollo de Software (PDS). Los Expertos en Contenidos con frecuencia, no dominan el PDS, y los desarrolladores no suelen poseer conocimientos suficientes en los diversos contenidos. Dicha característica trae consigo una brecha en la comunicación entre ambas entidades, con afectación en el desarrollo de herramientas que faciliten el aprendizaje. Por lo que se hace necesario vincularlos, para que puedan interactuar en el proceso de desarrollo de los contenidos educativos.
3. Las Herramientas asociadas a la gestión del software educativo se encuentran dispersas y por lo tanto, no integran sus procesos fundamentales como la planificación, el diseño, desarrollo y despliegue. (Jimmy, 2001)

Surge así, la necesidad de crear un Sistema de Gestión del Aprendizaje (SGA) que le permita al profesor la aprehensión de nuevos conocimientos para interactuar con las nuevas tecnologías y contribuya a prescindir de las actuales limitaciones para su uso. “Dándole además la posibilidad de que desarrollen, implementen sus propias ideas, que se sientan cómodos utilizando los nuevos recursos y medios de enseñanza.” Permitiendo que sirva de herramienta efectiva con el uso de los recursos tecnológicos en el proceso de concreción de sus intenciones educativas, y lograr una mayor interacción con los usuarios finales.

Como respuesta que solucione la necesidad planteada anteriormente, surge la idea de crear el proyecto Emedia con el objetivo de desarrollar y gestionar herramientas para el aprendizaje, de propósito general.

El sistema Emedia está compuesto por componentes independientes:

1. *Autor de Objetos de Aprendizaje*: Se utiliza en la creación de los objetos de aprendizaje. “Un objeto de aprendizaje es una entidad informativa digital desarrollada para la generación de conocimiento. “Un recurso digital que puede ser rehusado para ayudar en el aprendizaje.” (Wiley, 2000)
2. *Proyector*: Se ocupa de la visualización de los Objetos de Aprendizaje.
3. *Repositorio*: Posibilita el almacenamiento, acceso y gestión a los objetos de aprendizaje desarrollados.
4. *SGA*: Es la herramienta encargada del despliegue del sistema Emedia.
5. *Intérprete*: Le aporta a la aplicación mayor interactividad y versatilidad. Lo que significa una ampliación de la flexibilidad, puesto que el intérprete del sistema Emedia debe brindar la opción de interpretar código de alto nivel. Permitiendo la

manipulación de los Objetos de Aprendizaje.

Además de los componentes antes presentados, se hace inminente la búsqueda que encamine a una vía factible para la creación de guiones de contenido; con el objetivo de organizar los procesos educativos que guiarán las ideas de los Especialistas en Contenido o Expertos, en la concreción de sus intenciones para la realización de materiales educativos. Por consiguiente, se hace imprescindible trazar una línea de desarrollo de una herramienta que permita la realización de dichos recursos, centrado en el diseño de la misma como un primer avance de la aplicación, y que pueda contribuir a un posterior seguimiento.

A raíz de las condiciones descritas hasta el momento se identificó como **problema científico** de esta investigación: La inexistencia en el sistema Emedia de una Herramienta de Autor para la creación de guiones de contenido.

El problema descrito genera como **objeto de estudio** de la presente investigación: las Herramientas de Autor. El conjunto de principios que serán planteados como solución de esta investigación tendrán su incidencia final en: la Herramienta de Autor de Contenido para el sistema Emedia, lo cual determina el **campo de acción** de la misma.

De acuerdo con el problema científico planteado la **Idea a Defender** definida es la siguiente:

Si se diseña una Herramienta de Autor de Contenido para el sistema Emedia que permita la producción de Guiones de Contenido, entonces será posible facilitar al especialista en contenidos (Experto) la conversión de su experiencia y conocimientos en información fácil de usar; además de posibilitarle la creación de sus cursos a un costo y esfuerzo menor.

El **objetivo general** que se desea alcanzar para darle cumplimiento al presente trabajo es: Diseñar una Herramienta de Autor de Contenido para la creación de guiones de contenido. Para darle cumplimiento al objetivo trazado se determinó que las **tareas a realizar en esta investigación** estarían dirigidas a:

1. Revisar bibliografías referentes a términos que se tratarán en la investigación.
2. Analizar las tecnologías actuales en el mundo de la informática que apoyan la producción de software educativo.
3. Estudiar las necesidades educativas actuales.
4. Realizar el Modelo de Dominio.
5. Identificar los Casos de Uso.
6. Realizar el Modelo de Casos de Uso.
7. Describir detalladamente los Casos de Uso Arquitectónicamente Significativos.
8. Diseñar el Prototipo de Interfaz de Usuario.

9. Realizar diagrama de clases de Análisis
10. Realizar diagramas de clases e interacción del Diseño de los Casos de Uso de mayor prioridad.

El presente trabajo de diploma está estructurado de la siguiente manera:

Un **primer capítulo** que fundamenta todo lo concerniente al objeto de estudio, otras soluciones existentes; así como a los conceptos asociados al problema en cuestión. Además que hace referencia a las tendencias y tecnologías existentes en la actualidad, que presentan el contenido referido a las fundamentaciones teóricas que sirven de base para el desarrollo de la futura solución a proponer.

Un **segundo capítulo** donde se describe el negocio mediante un Modelo de Dominio, el cual sirve de base para determinar que es lo que se va a desarrollar. Descripción de la aplicación que se propone mediante un Modelo de Casos de Usos del Sistema; así como de todos los requisitos funcionales y no funcionales que detallan las capacidades operacionales de esta, y sus propiedades determinadas por un lenguaje entendible.

Un **tercer capítulo** como *solución propuesta*, presenta la solución concreta del análisis informático del problema. Se modelan los Diagramas de Clases del Análisis y Diseño, los Diagramas de Interacción del Diseño, y se especifican los patrones de diseño utilizados.

## Capítulo 1: Fundamentación Teórica

### Introducción

*“las herramientas de autor han evolucionado en la última década basada en innovaciones tecnológicas y pedagógicas desde publicaciones limitadas, sistemas educativos controlados por programas, (...), hasta las publicaciones ilimitadas y ambientes centrados en estudiante” (Dabahg, 2001).*

La frase anterior encamina el contenido del presente capítulo; en el cual se abordarán conceptos como: Herramientas de Autor, Software Educativo, Guiones de Contenido y Procesos de Enseñanza-Aprendizaje, que servirán de fundamentación teórica a la investigación presentada. Además se hace un análisis del estado del arte de los métodos, técnicas y herramientas empleadas para la captura de los requisitos, así como de las tecnologías que se emplean en el desarrollo de los prototipos.

### 1.1 Proceso de Enseñanza Aprendizaje.

*“El proceso de enseñanza-aprendizaje es un proceso esencialmente interactivo y comunicativo, de intercambio de información, compartiendo experiencias, conocimientos y vivencias, que logran una influencia mutua en las relaciones interpersonales.” (López, 2004)*

*“El proceso de enseñanza-aprendizaje constituye la vía mediatizadora esencial para la apropiación de conocimientos, habilidades, hábitos, normas de relación, de comportamientos, de valores legados por la humanidad, que se expresan en el contenido de enseñanza, en estrecho vínculo con el resto de las actividades docentes y extra docentes que realizan los estudiantes .” (M.Sc Febe Angel Ciudad Ricardo, 2007)*

Los medios de enseñanza apoyan este proceso evidenciando diferentes razones que explican su papel. *“La computadora y los software educativos, como medios de enseñanza resultan un eficiente auxiliar del profesor en la preparación e impartición de las clases, ya que contribuyen a una mayor ganancia metodológica y a una racionalización de las actividades del profesor y los alumnos.” (RUMBAUGH, et al., 2000.)*

### 1.2 La Informática y el software al servicio de la educación.

*“En las últimas tres décadas se ha desarrollado un vertiginoso avance de la incorporación de la informática al entorno educativo, dando surgimiento al área de la Informática Educativa y con ella a conceptos como Software Educativo, que se hace muy común en nuestros días.” (M.Sc Febe Angel Ciudad Ricardo, 2007).*



Lograr entornos más amigables que faciliten el desarrollo de software educativo y, además, adaptar los lenguajes a las necesidades y características del software educativo y del profesor, son aspectos que responden a su evolución (*dígase software educativo*). Las primeras ideas sobre software educativo aparecen en la década de los 60' con programas hechos con herramientas de programación.

Cataldi las divide:

1. Lenguajes para el aprendizaje (Logo).
2. Lenguajes y herramientas que sirven para la generación del producto de software educativo.
3. Las **herramientas de autor**.

Así ha ido creciendo en estos años esta nueva clase de software, cuyo objetivo es facilitar la creación, publicación y la gestión de los materiales educativos digitales a utilizar en la formación. El nivel de habilidades y conocimientos que el creador (autor) de los materiales educativos requiere, puede variar grandemente; abarcando tanto aquellas destinadas a la creación de materiales multimedia, como los editores de HTML, software de comunicación y trabajos colaborativos. (Tunis., 2005)

En este contexto surgen las herramientas de autor, como programas destinados a la creación de nuevos materiales educativos.

### 1.3 Conceptualización de Herramientas de Autor.

Según De Benito *“la puesta en marcha de experiencias educativas basadas en Internet implica la utilización de diferentes tipos de herramientas, como las de creación de recursos multimedia (programas para el tratamiento de imágenes, grabación y edición de sonido, video, generación de animaciones, etc.); de elaboración de contenidos de aprendizaje (editores de páginas Web, lenguajes de autor o archivos en formato pdf); de creación de pruebas y exámenes, de creación de ejercicios de autoevaluación; de comunicación, de gestión académica, para el trabajo colaborativo, gestión de alumnos, etc.”* (B. De Benito Crosetti, 2002)

Existen varias definiciones de Herramientas de Autor, y posiciones diversas a su denominación:

*“Las herramientas de autor son herramientas de desarrollo de software que posibilitan a diseñadores instructivos, educadores, maestros y aprendices diseñar un curso multimedia interactivo, y ambientes de aprendizaje en hipertexto sin el conocimiento de lenguajes de programación. Las herramientas de autor tienen como objetivo aplicaciones educativas que contengan generalmente, ya sea implícitamente o explícitamente, un modelo particular de la tarea en la que el usuario final debe estar ocupado, así como un modelo del proceso editorial del mismo.”* (Dabahg, 2001)

*“Las herramientas de autor son aplicaciones que tienen la intención de reducir el esfuerzo necesario para producir software, cargando con la responsabilidad en los aspectos mecánicos o la tarea, guiando al autor, y ofreciéndole elementos predefinidos que puede relacionar conjuntamente para satisfacer una necesidad particular (Educativa)”. (... ) (T. Murray, 2003)*

En general desde la perspectiva educativa, la finalidad de este tipo de herramientas es clara; el profesor crea el programa que le parezca más adecuado en función de sus objetivos, del plan de estudio y de los estudiantes que él atiende.

*“Cada **sistema de autor** provee herramientas o indicios que ayudan al autor en este (usualmente en forma descendente) proceso de estudiar en sus partes y elaborar el contenido al nivel necesario de detalle, para un modelo particular de elementos instructivos y sus relaciones”. (T. Murray, 2003)*

La selección de una Herramienta de Autor dependerá de dos factores: las características particulares de la aplicación a desarrollar y la formación y experiencias del propio desarrollador.

Según Bell *“son para ayudar a los diseñadores a hacer frente al incremento de la complejidad en la creación de aplicaciones interactivas, los investigadores y desarrolladores comerciales han producido una variedad de sistemas de autor que buscan reducir la labor asociada con la construcción de software para entrenar y educar, y para centrar el tipo de experiencia requerida en el contenido y elementos educacionales.” (Bell, 1998)*

Paul Bangs sostiene: *“que todos nosotros tenemos un concepto de la palabra “autor”, como verbo o como sustantivo. En un contexto literario general lo tomamos como un proceso creativo, que depende (usualmente) solamente de la mente y la pluma (o un procesador de texto) del escritor.” “Sin embargo en el mundo de las computadoras, el término “authoring” (redacción de un hipertexto y su preparación) es tomado como una dimensión nueva y diferente.” (Bangs, 2004)*

El **“grado de flexibilidad”** de una Herramienta de Autor, es la capacidad que esta posea para resolver problemas, más allá de los estereotipos preconcebidos por su creadores. Según De Leeuwe, *“son herramientas para crear con facilidad contenidos de aprendizaje.” (Leeuwe, 2004)*

Para Hall, *“los especialistas y/o programadores pueden usar las herramientas de autor para convertir el diseño basado en papel en el interno de un programa de aprendizaje. Esto le permite al autor ensamblar texto, gráficos, medios y animación, hilvanados dentro de un programa interactivo de instrucción.” (Hall, 2004)*

Según Barchino *“no son más que aplicaciones informáticas que crean y empaquetan elementos multimedia con la propia información del curso.” (R. Barchino Plata, 2004)*

Estos **metamedios**, como también se les suele llamar a las herramientas de autor; proveen generalmente módulos desde los cuáles se pueden organizar actividades, o se puede interconectar pequeños componentes y adecuar a los objetivos, conocimientos y habilidades que desee desarrollar el autor.

En la presente investigación se ha arribado a la conclusión, de que las Herramientas de Autor o *metamedios*: son aplicaciones informáticas, que les permiten a los expertos el diseño de la multimedia interactiva y el aprendizaje de entornos hipertexto; sin ningún tipo de conocimiento de lenguajes de programación. Disminuyendo así, el esfuerzo a realizar por parte de los educadores; ofreciéndoles guías, elementos y ayuda para crear materiales educativos en el proceso de formación.

### 1.4 Características Herramientas de Autor.

Las Herramientas de Autor se clasifican en cinco categorías (McGreal, Gram y Marks (1998)):

- Herramientas para la creación de recursos multimedia (*Media Creation Tools*).
- Herramientas para la edición de páginas Web (*Web Publishing Tools*).
- Herramientas para la comunicación a través de Internet (*Internet Based Conferencing Tools*).
- Herramientas de autor (*Internet Enabling Authoring Tools*).
- Herramientas integradas para la distribución de entornos de aprendizaje (*Integrated Distributed Learning Environment*).

Las Herramientas de Autor, según Landon (1998) “*diferencian dos tipos de aplicaciones: Aplicaciones Integradas (Integrated Application) y Aplicaciones Componentes (Component Applications). Las Aplicaciones Integradas son herramientas que integran diferentes aplicaciones para cada uno de los diferentes niveles de usuario: administrador, profesor y alumno, se corresponde con el último tipo de herramientas propuestas por los autores anteriores (Plataformas de Teleformación, LCMS (Learning Content Management Systems), Universidades Virtuales). Las Aplicaciones Componentes se refieren a herramientas más específicas, las cuales solamente desarrollan aspectos concretos como la comunicación (herramientas que proporcionan foros de discusión, etc.), ejercicios, desarrollo de materiales, etc.*” (B. De Benito Crosetti, 2002)

La relación de dependencia de la herramienta de autor con el modelo pedagógico del curso a crear y el modelo del proceso de producción en la herramienta de autor, constituyen objetivos que defienden estas herramientas en el proceso de desarrollo de software con finalidades educativas.

La selección correcta de las herramientas adecuadas a las necesidades específicas de una organización o usuario requiere de un análisis detallado de las funcionalidades que ofrecen las mismas. Para ello se trae a colación algunos criterios para su adecuada utilización:

1. *Facilidad de uso versus libertad creativa:* Este criterio permite establecer un continuo en uno de cuyos extremos se encuentran herramientas basadas en plantillas prediseñadas por el programador que no requieren por parte del usuario prácticamente ninguna preparación técnica. En este caso, lo que se gana en cuanto a facilidad de uso se pierde en flexibilidad y creatividad.
2. *Programación automática:* Esta funcionalidad libera a los desarrolladores de cursos de su dependencia con respecto a los programadores. De este modo no es necesario que los usuarios de la herramienta, dígame los profesores que desarrollen sus cursos y materiales e-learning, conozcan programación o dependan de expertos en esta área para poder elaborar sus contenidos.
3. *Interoperabilidad y standards:* La primera de estas características se refiere a la capacidad de la herramienta para trabajar con otro software y/o sistemas de e-learning. Esto se logra siempre y cuando la herramienta cumpla con los estándares internacionalmente reconocidos. Los estándares más reconocidos en el mercado son hoy por hoy: Aviation Industry Computer-based Training Committe (AICC), Sharable Content Object Reference Model (SCORM), IMS Global Learning Consortium y Microsoft LRN.
4. *Adaptación a distintos estilos de aprendizaje:* Algunas herramientas posibilitan esta opción mediante la creación de variables. Las variables permiten al instructor crear bifurcaciones en el proceso lineal del curso y orientar a los alumnos en estas específicas direcciones en función de criterios pedagógicos, de estilos específicos de aprendizaje detectados en los alumnos, etc.
5. El diseño de cursos con variables es más complejo que el diseño lineal y único clásico. Sin embargo abre la puerta a la creatividad y a la innovación pedagógica y didáctica.
6. *Ficheros multimedia:* La mayoría de las herramientas soportan ficheros JPGs, WAVs y GIFs. Sólo algunas soportan video y audio. Cerciorarse de lo que se necesita en el ámbito multimedia y de lo que ofrecerá la herramienta.
7. *Extensibilidad:* En algunas ocasiones los usuarios y/o las organizaciones se encuentran con la necesidad de adaptar la herramienta a sus propias necesidades emergentes. Para cubrir esta necesidad la mejor solución es obtener una herramienta de código abierto que permita a los programadores

adaptar la configuración inicial de la herramienta a las necesidades específicas de los usuarios.

Otras de las formas o vías de caracterización de las herramientas de autor está dada por la forma de aprender a usarla, que se reviste de dos pasos por lo general: conocer el uso o la mecánica de la herramienta en particular, y luego determinar cómo utilizarla en beneficio del proceso que uno ejecuta.

Las ventajas de la adopción de estos programas como instrumento de trabajo por parte de un profesor son múltiples y evidentes. Además del ahorro de tiempo de desarrollo y la menor formación técnica necesaria, hay que destacar otras razones como:

- La necesaria actualización de los métodos pedagógicos para adaptarnos, tanto individual como institucionalmente, a las necesidades educativas de la Sociedad de la Información.
- La mayor valoración profesional del profesor con destrezas tecnológicas frente al “analfabeto informático”.
- Las mejoras en la calidad de la enseñanza, ya que permiten un mayor grado de efectividad en la comunicación de contenidos, versatilidad horaria y independencia física del aula o centro formativo, liberación de las funciones más rutinarias relacionadas con la transmisión de información.
- Mayores posibilidades de trabajo en colaboración y posibles nuevas fuentes de ingresos (otras formas de edición, nuevos productos o aplicaciones). (Zazpe, 1998)

### 1.5 El Guión.

*“(...) el guión (...) abarca desde los aspectos estructurales y funcionales hasta los formales y estéticos, con un nivel de detalle que en principio permite desarrollar la ejecución del proyecto sin ambigüedades. Se discute mucho hasta dónde deben especificarse los detalles, pero mientras más preciso sea el guión mucho más rápidamente y a un menor costo se podrá llevar a cabo la elaboración de la obra.”* (Barrera Yanes, 1998)

*“La función primordial del guión es dar objetividad al proyecto de la obra multimedia de forma tal que pueda independizarse el proceso de ejecución del proyecto de concepción y diseño. (...) En segundo lugar, el guión es imprescindible para lograr una comunicación clara y precisa entre los integrantes del equipo de trabajo de manera que aunque realicen tareas independientes todos conozcan cómo tienen que hacer su labor y cómo encaja cada uno dentro de la obra.”* (Barrera Yanes, 1998)

El guión constituye un elemento importante en el desarrollo de una aplicación multimedia para la formación de los estudiantes. En una aplicación multimedia la metodología de desarrollo "obliga" a pensar que la organización de los contenidos, conjuntamente con el resto de los materiales multimedia debe tener una lógica a la hora de su presentación, y en cierto modo unas líneas maestras que sirvan de hilo conductor a la aplicación. Desde este punto de vista no tiene por que haber diferencias con otros medios audiovisuales tradicionales: cine, TV, publicidad, radio, etc. Y por supuesto con otras aplicaciones informáticas vinculadas como son los juegos, presentaciones, etc. En el caso concreto de una aplicación multimedia se puede distinguir un guión estructurado a tres niveles distintos:

- *Guión de contenidos*: incluirá todo lo relativo a la organización y estructuración de los contenidos, y que entre otros incluirá aspectos que tienen que ver con la modularización (diferentes niveles de fragmentación e interpretación) e interrelación (diferentes formas de vinculación entre los diferentes módulos) de los mismos. Como punto de partida podemos considerarlo como una primera aproximación a una representación hipertextual de los contenidos.
- *Guión de la aplicación*: representa el esquema de la organización funcional de la aplicación y abarca aspectos tales como organización de la navegación, vinculación física entre elementos, etc. Estos aspectos tienen que ver con la forma de como los contenidos van a ser transmitidos: mediante un simple "pasa páginas" o vinculados a través de una historia que incluya los aspectos típicos de cualquier dramatización: planteamiento, nudo y desenlace.
- *Guión multimedia*: una aplicación multimedia para la formación suele incluir varios elementos multimedia: videos, locuciones, animaciones con sonido, etc. En todos estos casos será preciso que exista un guión específico, que sirva de base al producto final.

### 1.6 Objetos de Aprendizaje

Formalmente no hay una única definición del concepto de objeto de aprendizaje y las definiciones son muy amplias. El Comité de Estandarización de Tecnología Educativa, plantea que los objetos de aprendizaje son "una entidad, digital o no digital, que puede ser utilizada, reutilizada y referenciada durante el aprendizaje apoyado con tecnología"; "cualquier recurso digital que puede ser reutilizado para apoyar el aprendizaje". (IEEE, 2001).

Se dan como ejemplos de objetos de aprendizaje los contenidos multimedia, el contenido instruccional, los objetivos de aprendizaje, software instruccional, personas, organizaciones o eventos referenciados durante el aprendizaje basado en tecnología. Dada la amplitud y variedad de las definiciones, así como la diversidad de recursos que pueden considerarse como OA, es difícil llegar a término estricto, es por ello que para la realización de este trabajo, se considerará como OA, a cualquier recurso con una intención formativa, compuesto de uno o varios elementos digitales, descrito con metadatos, que pueda ser utilizado y reutilizado.

### 1.7 Los Contenidos.

El desarrollo de los contenidos de un curso de formación viene determinado por una serie de aspectos que se pasa a comentar a continuación:

Metodologías de la formación que se aplican en los entornos multimedia de formación.

*Discursivas*: son aquellas que presentan una gran influencia del soporte tradicional de la formación: *el libro*. Suelen ser sencillos en su diseño y debido a su estructura funcional suelen denominarse "pasa páginas", pues su cualidad principal en la navegación es muy equivalente al manejo tradicional del libro.

*Exploratorias*: son aquellas cuya cualidad principal es ofrecer al usuario la capacidad de investigar sobre los contenidos sin una pauta fija, permitiendo una navegación y un aprendizaje, mediante ensayo y error.

*Simulaciones de entorno*: bastante utilizadas en la actualidad pretenden generar entornos virtuales que simulen los lugares de ocurrencia del proceso formativo. Mediante este procedimiento se recurre al planteamiento de situaciones en dichos entornos y a la evaluación de la toma de decisiones por parte del sujeto que aprende.

*Tipos de alumnos*: Este aspecto es bastante crucial, puesto que la manera en cómo ha de ser transmitido el conocimiento y evaluado el rendimiento de los estudiantes va a depender, entre otros, de los siguientes aspectos:

- Edad
- Nivel de estudios
- Entorno sociocultural
- Empleados o futuros trabajadores
- Proceso de aprendizaje individual o en grupo

*Elaboración de contenidos*: En este apartado cabe hacer diferentes consideraciones:

1. Los tipos de expertos, fundamentalmente especialistas en áreas y temas concretos así como técnicos en formación, que pueden participar en la elaboración de los contenidos. Básicamente pueden provenir de:



- *Las propias empresas cliente:* Suele ser el procedimiento más adecuado, puesto que son los auténticos expertos que conocen además la idiosincrasia de la empresa. Los problemas radican en que su dedicación a un proyecto de estas características, les limita continuar con las funciones que desempeñen desde el momento de incorporarse al proyecto.
2. *Contratación externa:* Si son temas de alta especialización y no se puede contar con los expertos de la empresa cliente, será preciso recurrir a consultores externos. Los problemas suelen radicar en el desconocimiento de las características de la empresa y en los altos costes que conlleva su contratación.
  3. *Las propias empresas desarrolladoras del producto:* Existen empresas que desarrollan productos multimedia para la formación que cuentan con expertos tanto en los aspectos tecnológicos como en los de formación, de esta manera la principal ventaja estriba en un trabajo con grupos homogéneos y que conocen ambos aspectos del desarrollo. En este caso los expertos en formación actúan de interlocutores con los formadores de la empresa y actuando como ingenieros de la adquisición del conocimiento.
  4. *Adquisición del conocimiento, fundamentalmente centrado en los dos tipos de conocimiento, declarativo y procedimental:*
    - El conocimiento declarativo: es la adquisición de una base de conocimiento adecuadamente organizada y estructurada, relativa a un dominio de intervención determinada.
    - El conocimiento procedimental: es la adquisición de habilidades de toma de decisión y de resolución de problemas pertenecientes a dicho dominio.

### **1.8 Herramientas de Autor existentes en el mercado.**

Las Herramientas de Autor más básicas son aquellas que solamente permiten un conjunto limitado de acciones para que el usuario interactúe con el sistema, por ejemplo, crear mapas sensibles o ir de una diapositiva a la siguiente, ejemplo de estas herramientas puede ser OpenOffice.org, Impress o Mapas Web, sensibles a conjuntos de páginas html con javascript. Las herramientas más avanzadas incluyen lenguajes de programación como en el caso de Squeak o la creación de ambientes con flash.

Algunas Herramientas de Autor incluso permiten tomar elementos digitalizados, o learning objects, de un curso para reutilizarlos en otro. Posibilitan adaptar la



información, su presentación y secuenciación, a las necesidades del experto. Haciendo permisibles los siguientes aspectos:

1. Relacionar la información y sus fuentes.
2. Incluir elementos de contenido que evalúen los conocimientos alcanzados.
3. Activar objetos de aprendizaje.

En la actualidad existen variadas y diversas Herramientas de Autor, que se utilizan de disímiles formas. En la presente investigación se señalaron las siguientes:

### **Authorware**

Herramienta dedicada a la creación de contenido e-learning altamente interactivo. Permite importar presentaciones de Microsoft PowerPoint, reproduce contenido en Mac OS X, utiliza estándares de la industria en relación con JavaScript e integra fácilmente contenido con Learning Management Systems (LMS). Es una herramienta muy potente pero su principal desventaja es que requiere un proceso intenso de aprendizaje.

### **Autore**

Esta herramienta multiplataforma permite generar contenidos e-learning con una interfaz extraordinariamente sencilla e intuitiva. Su uso no requiere ningún tipo de conocimientos en programación. Los contenidos que genera son materiales multimedia estructurados de acuerdo al modelo de un Objeto de Aprendizaje (Objetivo o idea, Desarrollo y Evaluación). Autore cuenta con un potente editor de páginas y es capaz de importar textos, imágenes, animaciones, documentos de audio y video, ecuaciones MML. Además, es posible elaborar ejercicios de autoevaluación. Los metadatos se pueden introducir durante el proceso de autoría. Autore ha sido diseñada e implementada por el Campus Virtual de la UPV-EHU y es uno de los elementos que conforman su LCMS.

### **Clic**

Clic es un software de libre distribución que permite crear diversos tipos de actividades educativas multimedia en sistema operativo Windows. Permite crear distintos tipos de actividades que pueden contener texto, gráficos, sonidos y otros recursos multimedia.

También es posible encadenar grupos de actividades en paquetes para que se ejecuten secuencialmente. Orientado al profesorado de niveles pre-universitarios.

### **EasyProf**

Herramienta de autoría diseñada para que autores y formadores sin conocimientos de informática puedan crear contenidos educativos multimedia. Genera todo el contenido en HTML usando SCORM v1.2. Permite acceso a demos de todas sus utilidades y posibilidades de presentación de contenidos.

La existencia de estas herramientas de autor, posibilita crear contenidos educativos mediante aplicaciones informáticas para usuarios no expertos en informática. En la presente investigación la herramienta a desarrollar, parte de la misma idea; sólo que está encaminada a la creación de guiones de contenido, para permitir al educador la producción de los cursos.

### **1.9 Metodologías de software**

Las metodologías para el desarrollo de software describen el conjunto de fases, etapas, actividades y tareas asociadas a la producción de software (aplicación) de calidad. La finalidad del uso de metodologías es lograr el desarrollo de un software de calidad.

Rumbaugh dio la siguiente definición:

*“Una metodología de ingeniería de software es un proceso para la producción organizada del software, empleando para ello una colección de técnicas predefinidas y convencionales en las notaciones. Una metodología se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada paso. Los pasos de la producción del software se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo”* (RUMBAUGH, y otros, 2000.)

*Piattini define a la metodología de desarrollo de software como “un conjunto de procedimientos, herramientas y un soporte documental que ayuda a los desarrolladores a realizar nuevo software”.* (Piattini, 1996)

Clasificar las metodologías es una tarea bien difícil por la gran cantidad de propuestas y diferencias en el grado de detalle, información disponible y alcance que poseen. A grandes rasgos, considerando su filosofía de desarrollo se pueden agrupar en: metodologías ágiles o ligeras y metodologías pesadas.

### 1.9.1 Metodologías ágiles o ligeras

*“Estas metodologías están más orientadas a la generación de código con ciclos muy cortos de desarrollo, se dirigen a equipos de desarrollo pequeños, hacen especial hincapié en aspectos humanos asociados al trabajo en equipo e involucran activamente al cliente en el proceso. Entre las metodologías ágiles tenemos a: Extreme Programming (XP), SCRUM, Crystal Clear, Feature -Driven Development (FDD), Dynamic Systems Development Method (DSDM), Adaptive Software Development (ASD).”* (Collazo, 2007)

### 1.9.2 Metodologías pesadas

Las metodologías pesadas son aquellas que están guiadas por una fuerte planificación durante todo el proceso de desarrollo, en la cual se establecen estrictamente las actividades involucradas, los roles definidos, los artefactos que se deben producir, las herramientas y notaciones que serán utilizadas así como el modelado y documentación detallada.

Ejemplo de estas metodologías tenemos a: SW-CMM (Software Capability Maturity Model), RUP (Rational Unified Process), entre otras.

Precisamente en el actual trabajo se decidió tomar RUP como Metodología de Desarrollo de Aplicaciones, debido a:

#### 1.9.2.1 Metodología Orientada a Objetos: RUP

La metodología RUP, fue desarrollada en 1998 por Grady Booch, Ivar Jacobson y James Rumbaugh, prominentes metodólogos en la industria de la tecnología y sistemas de información.

Una Metodología de Desarrollo de Software Orientada a Objeto (OO) consta de los siguientes elementos:

- Conceptos y diagramas (Modelo).
- Etapas y definición de entrega en cada una de ellas.
- Actividades y recomendaciones. (ISW 2006)

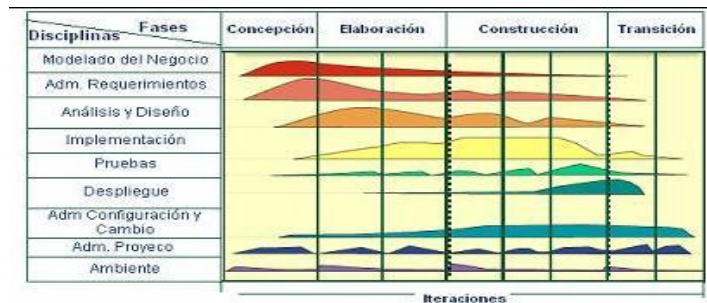
El Proceso Unificado de Rational (RUP) *“... es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema software. Sin embargo, el Proceso Unificado es más que un simple proceso; es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas de software, para diferentes áreas de aplicación, diferentes tipos de organización, diferentes niveles de aptitud y diferentes tamaños de proyecto”* (Robertson, 2003)

“El Proceso Unificado es una propuesta de proceso para el desarrollo de software orientado a objeto que utiliza el Unified Model Language (UML) para describir todo el proceso. Está basado en componentes, y además es uno de los procesos más generales de los existentes actualmente, y en realidad está pensado para adaptarse a cualquier proyecto, y no tan solo de software. Sus características principales son:

- Guiado/Manejado por casos de uso.
- Centrado en arquitectura.
- Iterativo e Incremental.
- Desarrollo basado en componentes.
- Utilización de un único lenguaje de modelación.
- Proceso Integrado.” (López, 2004)

Para desarrollar la propuesta que presenta este trabajo, se ha decidido utilizar esta Metodología, principalmente porque este proceso utiliza UML, de ahí que se ha convertido en el estándar internacional para definir, organizar y visualizar la arquitectura de una aplicación orientada a objetos.

El RUP divide el proceso de desarrollo en ciclos, teniendo un producto final al final de cada ciclo. **Ver Figura 1.**



**Figura 1. Metodología de desarrollo de software: RUP.**

Cada ciclo se divide en fases que finalizan con un hito donde se debe tomar una decisión importante:

- **Conceptualización (Concepción o Inicio):** Se describe el negocio y se delimita el proyecto describiendo sus alcances con la identificación de los casos de uso del sistema.
- **Elaboración:** Se define la arquitectura del sistema y se obtiene una aplicación ejecutable que responde a los casos de uso que la comprometen. A pesar de que se desarrolla a profundidad una parte del sistema, las decisiones sobre la arquitectura se hacen sobre la base de la comprensión del sistema completo y los

requisitos (funcionales y no funcionales) identificados de acuerdo al alcance definido.

- **Construcción:** Se obtiene un producto listo para su utilización que está documentado y tiene un manual de usuario. Se obtiene uno o varios entregables del producto que han pasado las pruebas. Se ponen estos entregables a consideración de un subconjunto de usuarios.
- **Transición:** El entregable ya está listo para su instalación en las condiciones reales. Puede implicar reparación de errores.

La integración en una sola aplicación de la gestión de proyectos de desarrollo de software y el trabajo colaborativo de diferentes roles en la construcción, que ofrece la herramienta Rational, no existe actualmente en el mercado, incluso en herramientas de Software Libre. En contraposición a lo anterior, hoy día si se cuenta con herramientas basadas en RUP que permiten la generación automática de la estructura del código para múltiples lenguajes de programación, y posibilitan los controles de calidad y del proyecto.

### 1.10 Sistema Operativo GNU/Linux.

Este proyecto será desarrollado a partir de las herramientas que brinda el software libre. Buscando la independencia tecnológica que este proporciona al posibilitar la libertad de uso y distribución de los programas sin incurrir en litigios de licenciamiento o asuntos legales.

#### 1.10.1 Distribución de Linux: Debian GNU/Linux

El Proyecto Debian es una asociación de personas que han hecho causa común para crear un sistema operativo (SO) libre, bajo la licencia General Public Licence (GPL). Basado en el conocido y distribuido núcleo de Linux la distribución es llamada Debian. Una gran parte de las herramientas básicas que completan el sistema operativo, vienen del proyecto GNU; de ahí el nombre: GNU/Linux. Debian es la única distribución importante de GNU/Linux mantenida solamente por voluntarios, es decir, sin un enfoque comercial, esto tiene ventajas y desventajas.

En primer lugar, las personas que se dedican a Debian tienen una alta motivación en participar en la misma, y se actualiza la distribución diariamente, apareciendo paquetes nuevos de software constantemente. Al mismo tiempo, existe un compromiso de calidad, no se desea distribuir software con errores.

En segundo lugar, dada su actitud abierta a la participación de todos, en el mismo espíritu original de Linux, constantemente hay personas que se unen a Debian para

participar aportando su granito de arena, no solamente haciendo paquetes de programas, sino colaborando en el Servidor de Web, traduciendo documentación de Debian, documentando fallos, o ayudando a los usuarios a través de las listas de correo que mantiene la comunidad.

Como desventajas tiene un mayor componente técnico que otras distribuciones. También, es posible que ciertos paquetes no estén tan actualizados como debieran, quizás porque sus desarrolladores han dejado de actualizarlos y nadie se ha hecho cargo. Sin embargo esto es algo que todos los desarrolladores tratan de evitar y, aunque cada desarrollador mantiene sus paquetes, no es raro que otro desarrollador (incluso un usuario) envíe una nueva versión del paquete para arreglar un problema o actualizarlo. (DEBIAN, 2007)

Se decidió usar la distribución Debian porque es una distro (distribución de GNU/Linux) de desarrollo muy estable, por lo que los paquetes que se desarrollen en él y quieran ejecutarse utilizando cualquier distribución siempre serán estables. A diferencia de otras distribuciones tiene un magnífico soporte de estabilidad en las aplicaciones (no requieren ser compiladas en la máquina que las esté usando). Los módulos del LDAP (Lightweight Directory Access Protocol) se pueden ejecutar sin problemas permitiendo que los usuarios usen sus sesiones en cualquier máquina dentro del área de trabajo, ahorrando recursos de hardware.

### 1.11 Lenguaje de modelado UML

Entre los lenguajes de modelado que define OMG (Object Management Group) el más conocido y usado es UML (Unified Modelling Language). UML es un lenguaje gráfico para especificar, construir y documentar los artefactos que modelan un sistema. UML fue diseñado para ser un lenguaje de modelado de propósito general, por lo que puede utilizarse para especificar la mayoría de los sistemas basados en objetos o en componentes, y para modelar aplicaciones de muy diversos dominios de aplicación y plataformas de objetos.

Se estableció comparación entre UML 1.5 y UML 2.0; este último ofrece a los proveedores de herramientas CASE (Computer-Aided Software Engineering) la producción automática de programas basado en especificaciones del software.

El UML 1.5 está constituido por 7 diagramas básicos y dos diagramas (**Ver Figura 2**) que constituyen variaciones de dos de los anteriores:

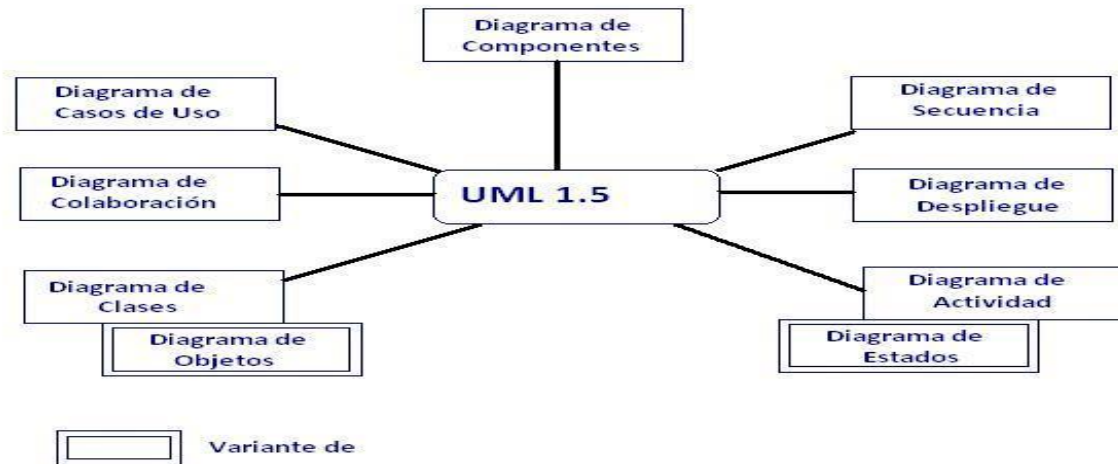


Figura 2. Diagramas de UML 1.5

En UML 2.0 se definen una serie de diagramas adicionales a los establecidos en UML 1.5. El conjunto de diagramas se encuentra organizado en torno a dos categorías: diagramas estructurales y diagramas dinámicos o de comportamiento. (Ver Figura 3)

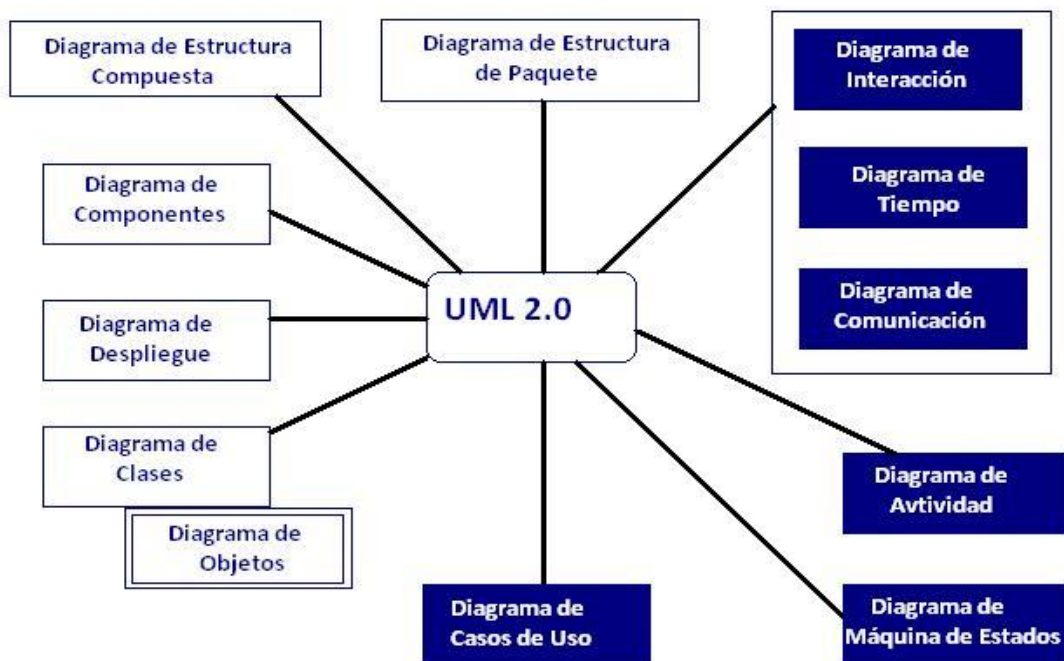


Figura 3. Diagramas de UML 2.0

UML 2.0 es la mayor revisión que se le ha hecho a UML desde la versión 1.0. El modelo conceptual en el 2.0 ha sido reestructurado completamente y nuevos diagramas han sido incorporados. Los diagramas tradicionales también han sido mejorados, por ejemplo, los diagramas de secuencia y de despliegue; en el primero es posible hacer referencias a otros diagramas de secuencia, además de incluir flujos



alternos, opcionales, lazos, entre otros y en el segundo, sobre los diferentes nodos de la infraestructura de red se colocan, a modo de artefactos (elementos físicos simples), los elementos componentes del software. (ANACHE, 2004)

En UML 2.0 los diagramas aparecen dentro de un marco (frame) que posee una etiqueta para indicar el tipo de diagrama, lo que permite al usuario una referencia rápida al diagrama invocado.

Esta versión proporciona a los analistas, arquitectos y desarrolladores; herramientas cada vez más potentes que les permiten aprovechar mejor los modelos y como consecuencia generar una mayor cantidad de código reduciendo significativamente el ciclo de desarrollo de sus aplicaciones.

### 1.12 Plataforma de desarrollo de Software

Las **Plataformas de Desarrollo de Software** son una solución completa para desarrollar software y sistemas basados en software. Como parte de la necesidad de implantar una plataforma homogénea de diseño, modelado y desarrollo de software se investigaron tecnologías de licenciamiento libre (Software Libre) y propietario: Umbrello, ArgoUML, BoUML y RSA.

**ArgoUML** es una herramienta muy completa, es Open Source, y libre bajo la licencia BSD, a pesar de no estar completamente terminada. Sin embargo el modelo de usuario no está completamente implementado. Este modelo consiste en modificar la herramienta para hacerla más usable en función de la información que se maneje del usuario. Las imágenes que genera de los diagramas, no son lo suficientemente óptimas estéticamente. Además es un poco lenta y se bloquea en repetidas ocasiones, lo que no permite un viable desarrollo en su uso para la realización de diagramas.

**Umbrello** es una herramienta soportada en software libre, open source bajo la licencia GPL. Soporta los 9 diagramas principales de diseño, así como un gran número de lenguajes de programación, entre ellos, python. Su interfaz es extremadamente simple y minimalista. Los diagramas se agrupan en vistas: vista lógica, vista de componentes, vista de casos de uso, de entidad-relación, y las clases que se usan son comunes a todos. Genera código automáticamente y gestiona la documentación. Su instalación es cómoda para el usuario y ocupa poco espacio. Se encuentra en los repositorios oficiales de Ubuntu, con buena valoración por parte de la comunidad. Sin embargo es demasiado lenta y se bloquea constantemente. La pantalla de diseño del



diagrama está poco automatizada. Se basa en Drag & Drop, pero es demasiado simple y las imágenes que genera son demasiado sencillas.

**BoUML** es una herramienta freeware (con las fuentes disponibles) que ofrece versiones compiladas para los principales sistemas operativos gracias al framework Qt. Es muy ligera y fácil de instalar permite visualizar la documentación del proyecto en html e importar el diseño a xmi y mdl, soporta los principales lenguajes orientados a objetos: C++, Java y PHP. Permite incluir y exportar código automáticamente a partir del diseño y generar documentación en varios formatos (HTML, XMI,). Es una herramienta muy fácil de usar. *“Permite trabajar con UML.2, es rápida y apenas consume memoria, generar código para Java, C++ e IDL (y controla bastante la generación), hace reingeniería inversa (a partir del código sacar el modelo (Permite trabajar con UML 2, y además aunque no es Java, también es multiplataforma: Linux,*

*MacOS y Windows.* “ (García, 2007). Estas últimas funcionalidades no han sido soportadas aún por la herramienta de modelado ArgoUML.

### **IBM Rational Software Architect**

Es un entorno de modelación y desarrollo que aprovecha el Lenguaje Unificado de Modelado versión 2.0 (UML v2.0) para diseñar arquitecturas de software soportadas por C++, J2EE y aplicaciones Web. Aprovecha al máximo la plataforma abierta Eclipse 3.2. Ofrece una gama completa de herramientas de desarrollo de software para una variedad de tecnologías de implementación.

Las muchas características de modelado y edición visual muy poderosas de la herramienta están diseñadas para mejorar la productividad, elevar el control arquitectónico y facilitar la experiencia del diseño al código para Java 2 Enterprise Edition, C++, la Arquitectura Orientada a Servicios (SOA) y los servicios Web. RSA ayuda a migrar con más facilidad de una herramienta de desarrollo a otra. Permite agregar características a medida que cambien las necesidades, lo que hace del producto una selección consumible para las necesidades de diseño y de construcción. Además es único en su habilidad de dar soporte de modelado a otros dominios tales como WSDL y XSD (XML Schema Definition), permitiéndole crear diagramas que combinan los elementos de UML con estos últimos. (RUMBAUGH, y otros, 2000)

En la presente investigación se arribó a la conclusión de que la herramienta a utilizar para la modelación de los diagramas es: RSA, por las características que responden positivamente al modelado de la aplicación. Dentro de las facilidades que este le proporciona a la herramienta clasifican:

Está basado en ECLIPSE.

Trabaja en el Lenguaje de Modelado 2.0 lo que permite la realización de un gran número de diagramas con sus funciones que posibilitan su explicación con mucha más claridad.

Permite además realizar la ingeniería inversa de códigos con lenguaje C++ a diagramas UML.

### **1.13 Bibliotecas gráficas.**

Los sistemas libres están adentrándose con creces en el ambiente de las interfaces gráficas de usuario, dejando atrás el mito del sistema operativo en modo texto, para ello emplean diferentes paquetes de desarrollo y además variadas tecnologías para la representación de los elementos gráficos.

Este epígrafe resalta las características principales de las tecnologías libres empleadas para construir software de código abierto.

#### **1.13.1 GTK.**

GTK (GIMP Toolkit) es un grupo importante de bibliotecas o rutinas para desarrollar interfaces gráficas de usuario (GUI) para principalmente los entornos gráficos GNOME, XFCE y ROX de sistemas Linux. GTK es la abreviatura de GIMP toolkit (conjunto de rutinas para GIMP). Es software libre (bajo la licencia LGPL), multiplataforma y parte importante del proyecto GNU. Inicialmente fue creado para desarrollar el programa de manejo de imágenes GIMP, sin embargo actualmente es muy usada por muchos otros programas en los sistemas GNU/Linux. Cabe mencionar que Qt es una alternativa a GTK que también es muy utilizada (en el entorno KDE, por ejemplo). GTK se ha diseñado para permitir programar con lenguajes como C, C++, C#, Java, Perl, PHP o Python. Actualmente su última versión es GTK+ 2, con una cantidad importante de mejoras respecto a la primera versión, aunque sin embargo, no es compatible con ella.

#### **1.13.2 Librería Gráfica: QT**

Para el desarrollo de los prototipos de interfaces gráficas de usuario y de la posterior aplicación, se hace necesario utilizar alguna librería que permita implementarlo, con este objetivo se seleccionó la librería QT.

Qt es una biblioteca de programación para desarrollar aplicaciones con interfaces gráficas de usuario. Permite desarrollar programas profesionales rápidamente. La biblioteca Qt está disponible para Linux y también para varias versiones de Unix, e incluso para Windows y Macintosh. Por lo tanto, permite escribir programas que pueden convertirse fácilmente a esas plataformas.

Qt comenzó a distribuirse comercialmente en 1996 y desde entonces ha sido la base para numerosas aplicaciones incluyendo la popular interfaz gráfica para Linux llamada KDE, disponible en todas las grandes distribuciones de Linux.

En el mercado se pueden encontrar las siguientes distribuciones de Qt:

1. Qt Enterprise Edition y Qt Profesional Edition, disponibles para el desarrollo de software con fines comerciales. Incluye servicio de soporte técnico y están en disponibles ampliaciones.
2. Qt Free Edition Es la versión para Unix/X11 para el desarrollo de software gratuito y de código abierto. Se puede obtener gratis sujeto a los términos de la Q Public License and the GNU General Public License. Para plataformas Windows también esta disponible la versión Qt non comercial.
3. Qt Educational Edition es una versión de la Qt Profesional Edition con licencia únicamente para fines educacionales.
4. Qt/Embedded Free Edition.

### 1.13.3 QT- Designer

QT Designer es una herramienta para el desarrollo de formularios y presentaciones gráficas para las aplicaciones con las librerías QT. Permite acelerar el desarrollo de interfaces de alto rendimiento, a la vez que proporciona una forma fácil de diseñar interfaces gráficas de usuario avanzadas generando el código fuente para las mismas, lo que permite al desarrollador ajustarlo a sus necesidades.

Este generador de interfaces fue creado inicialmente por la empresa TROLLTECH para trabajar en varias distribuciones Linux. No obstante, actualmente puede instalarse en otras plataformas como Windows y Mac OsX.

El QT Designer utiliza como base la librería gráfica de QT, que ha sido transportada a diversas plataformas, lo que permite que el código generado por el QT Designer pueda ser utilizado en diversas plataformas. Además, el QT funciona sólo o asociándose a algunos entornos de desarrollo integrado como Visual Studio .Net o Eclipse. Esta

herramienta provee características muy poderosas como la previsualización de la interfaz, soporte para widgets y un editor de propiedades bastante poderoso.

Una vez concluido el análisis de las bibliotecas hemos decidido utilizar la biblioteca Qt. La selección se debe a que posee un framework más completo y con mucha más utilidad a la hora de un desarrollo rápido, usando C++ como lenguaje de programación. Es muy respetado, una librería de muchas funcionalidades de alto rendimiento y prestaciones, documentación impecable y GPL. GTK es una librería muy extendida pero con más bugs que Qt y documentación confusa, desactualizada y escasa. Cuando analizamos mas a fondo nos damos cuenta de que los programas que usan Qt son más ligeros y dan la sensación de mayor configurabilidad que sus contrapartes en GTK. Se puede concluir que las aplicaciones Qt son más rápidas, ligeras y optimizadas.

### Capítulo 2: Características del Sistema

#### Introducción

*El proceso de aprender como desarrollar aplicaciones educativas solo acaba de comenzar. Aplicaciones educativas para la Web o CD-ROM (HC) son justamente uno de los resultados de este vertiginoso desarrollo, que permite convertir los conocimientos en información fácil de usar y brindar asistencia a los desarrolladores de software educativo. (Hennicker, 2000)*

En este Capítulo se hace referencia al estado actual del negocio, y para poder comprender el contexto en el que se emplaza el sistema se necesita definir conceptos que se puedan agrupar en un Modelo de Dominio como alternativa al Modelo de Negocio, en sistemas altamente centrados en tecnologías informáticas; debido a la poca estructuración de los procesos que describen el negocio. Además se enumeran los Requisitos Funcionales y No Funcionales, determinando la concepción general del sistema y detallando la propuesta que presenta la investigación.

#### 2.1 Estado actual del Negocio

##### 2.1.1 Sistema Emedia

La concepción del sistema Emedia está encaminada a ser un software instalado en un servidor que se utiliza para crear, gestionar y distribuir cursos destinados a la preparación individual de cada estudiante.

Sirve además de contenedor de cursos y Objetos de Aprendizaje (OA), e incorpora otras herramientas para facilitar la comunicación y el trabajo colaborativo entre profesores y estudiantes, permitiendo el seguimiento y la evaluación del alumno.

Este está compuesto por 5 componentes independientes, que la integran como un SGA. Con estas herramientas se intenta lograr que el sistema para realizar su producto final, no necesite la utilización de herramientas foráneas; sino que acceda a las herramientas integradas en su paquete. Esto brinda una facilidad de uso plena, debido a que la instalación de las funcionalidades para la producción del curso, sólo se hace una vez. El sistema Emedia usa como alternativa el Software Libre y las oportunidades que GNU/Linux le ofrece para su instalación, reduciendo los costos de informatización.

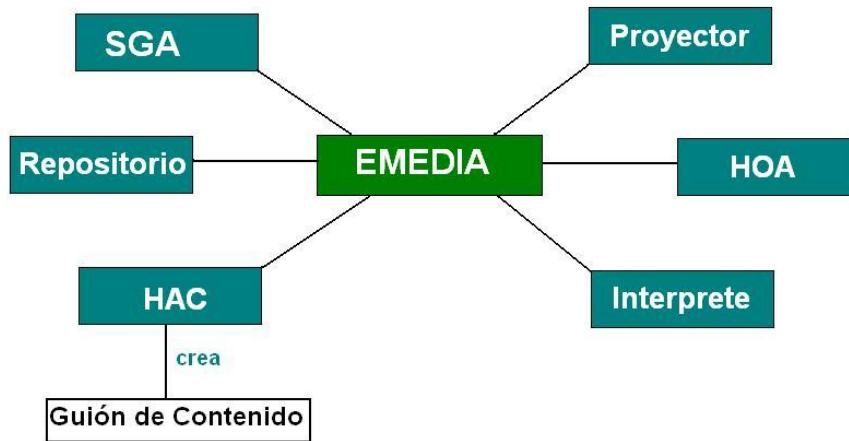


Fig 4. Concepción Sistema Emedia.

### 2.1.2 Herramienta Autor de Contenido (HAC)

Como solución a los objetivos propuestos en la presente investigación se propone desarrollar una Herramienta de Autor de Contenido, que funcionará como un componente de diseño independiente para la creación de guiones de contenido, que se integrará al sistema Emedia.

**Autor de Contenido:** La Herramienta de Autor de Contenido permitirá la creación de Guiones de Contenido. Los guiones poseen Elementos de Contenido, los cuales están compuestos por Elementos Contenedores y Objetos de Aprendizaje (estos a su vez poseen componentes).

La arquitectura propuesta persigue que la herramienta que se elabore pueda dar respuesta a los requerimientos de portabilidad, comprensibilidad, extensibilidad y eficiencia. Uno de los requerimientos que más estresan la solución es la cualidad de que el sistema pueda desplegarse en ambiente consola tipo Escritorio y tipo Web. Se impone una dualidad de conceptos en los que se aboga por el desempeño que se logra con las aplicaciones nativas en modo de consolas gráficas, y por otro lado el empeño en la facilidad, ubicuidad y portabilidad que brindan las tecnologías Web.

En esta primera etapa del desarrollo de la presente Herramienta de Autor de Contenido, no se aborda la capacidad de la herramienta de ser visualizada sobre la Web, sino que funcione como una herramienta escritorio. Es necesario garantizar que la futura integración de estos componentes tribute en correspondencia con la arquitectura definida para la Herramienta principal.

La HAC que se propone brinda funcionalidades de crear, exportar, importar, editar, eliminar tanto los guiones como los elementos de contenido agregados a estos. De

forma tal que el profesor (Experto en contenidos) sea capaz de convertir su experiencia en información fácil de usar por los estudiantes. Y se realicen las actividades de automatización en la creación de los cursos educativos. La principal idea de estos sistemas, es lograr que el profesor pueda diseñar un curso, sin ningún conocimiento de lenguajes de programación.

La funcionalidad de Crear Guiones permite crear un guión de contenido, y guardarlo en la plantilla correspondiente.

La funcionalidad de Exportar Guión permite exportarlo con el formato deseado por parte del Experto.

La funcionalidad de Adicionar Plantilla de Estudiante permite al Experto crear la plantilla correspondiente del estudiante, y añadirlo al paquete Modelo Estudiante.

En resumen con la presente aplicación el profesor podrá crear su guión de contenido para el curso, mostrándose la taxonomía del guión de contenido como un árbol de jerarquías.

### 2.2 Modelo de Dominio

Debido al bajo nivel de estructuración que presenta el negocio sobre el que se está investigando y que está altamente centrado en tecnologías informáticas, se propone un Modelo de Dominio, ya que de manera visual permite mostrar los principales conceptos que se manejan en el dominio del sistema en desarrollo. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un claro y pleno conocimiento del funcionamiento del objeto de estudio. Este modelo contribuirá a identificar algunas clases que se identificarán en el sistema.

Según el Método para la modelación de entornos organizacionales con flujos de información complejos y difusos (**DoMet**), se consta con las siguientes actividades para hacer más entendible el entorno de trabajo:

1. *Determinación de la estructura organizacional:* consiste en la determinación de jerarquía por niveles de la organización en estudio. **Organigrama.**
2. *Determinación de los eventos principales:* al no poder delimitarse los procesos por no existir fronteras entre los mismos, se hace necesario listar los eventos principales que ocurren en la organización. **Descripción de Eventos Principales.**
3. *Determinación de informaciones:* objetos palpables físicos, conceptos que se manejen en la organización o informaciones propiamente dichas. **Glosario de Conceptos.**

4. *Determinación de quien o quienes participan:* quienes interactúan con los eventos principales descritos. **Registro de Roles**
5. *Definición de clases:* resultado de todas las actividades anteriores. **Clases del dominio.**
6. *Relacionar clases del dominio.* (M.Sc Febe Angel Ciudad Ricardo, 2007)

En la presente investigación se consta de las siguientes actividades:

1. *Organigrama:* Entidad Organizacional Educativa. En este caso se hace referencia a la entidad en la cual se realizarán las actividades de producción de los cursos destinados a la docencia.
2. *Eventos principales:*
  - *Impartir clases:* Es el momento en el cual guiado por la estructuración del curso a través de las formas de organización de la docencia, el profesor presenta nuevos contenidos a los estudiantes, apoyándose en los documentos y los medios de enseñanza.
  - *Evaluación de lo aprendido:* Es el evento mediante el cual el profesor se retroalimenta del estado del aprendizaje del estudiante. Este se realiza a través de diferentes formas orales o escritas, determinadas por el plan de estudios de la especialidad. Tiene una evaluación determinada en la escala de 2, 3, 4 ó 5 puntos.
3. *Conceptos:*
  - *Forma de organización de la Docencia:* Es la manera en la que en la enseñanza universal se organizan los procesos de enseñanza – aprendizaje y que delimitan la manera en la cual de forma consecutiva los estudiantes desarrollarán este proceso con la guía y tutoría de los profesores. Estará determinada por las Clases, Clases Prácticas, Laboratorios, Seminarios, Talleres.
  - *Documentos:* Son aquellos materiales que utilizan los estudiantes para el aprendizaje. Estos pueden ser: Libros, Revistas, Artículos, Guías.
  - *Medios de Enseñanza:* Son aquellos medios de los cuales se apoya el profesor para impartir los contenidos establecidos por el Plan de Estudio (Guión del contenido del curso). Estos serán Maquetas, Esquemas, Power Point, etc.



- *Cursos*: Conjunto de recursos y materiales de aprendizaje con un objetivo específico.
- *Guión de Contenido*: Incluirá todo lo relativo a la organización y estructuración de los contenidos. Es un documento que precisa objetivos, estrategia pedagógica, dinámica comunicativa, que brinda una secuencia lógica de los contenidos y las ideas principales de diseño, para dar respuesta a una necesidad educativa específica y que debe ser consultada en la elaboración de un software educativo.
- *Plantillas*: Patrón que sirve como modelo para organizar una información determinada.
- *Entidad Organizacional Educativa*: Referente a toda organización destinada a la producción de cursos educativos.

4. *Quien o quienes participan*: profesor (Experto en Contenidos) y estudiante.

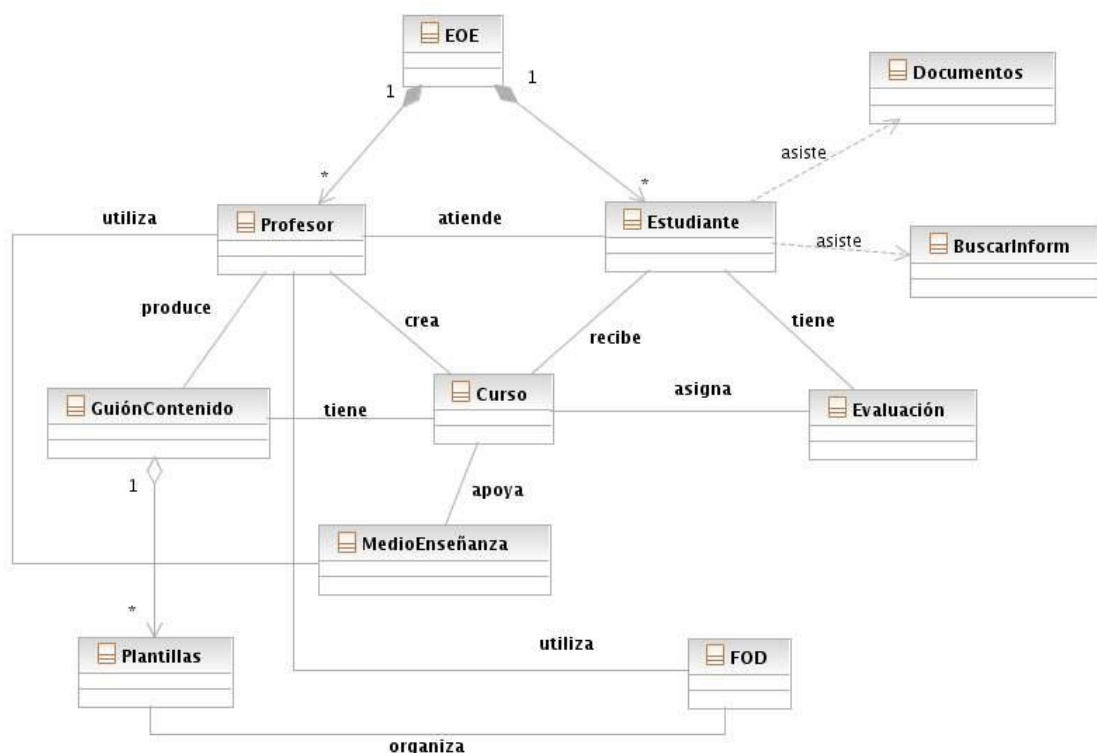


Fig 5. Modelo de Dominio

### 2.3 Especificación de los Requisitos

Las especificaciones de requisitos constituyen la forma de detallar todas las capacidades operacionales del sistema y sus propiedades expresándolas en un

lenguaje entendible para las partes implicadas en el desarrollo. En el marco del presente trabajo las especificaciones de requisitos están constituidas por los requisitos tanto funcionales como no funcionales y los casos de uso en que se agrupan dichos requisitos.

### **2.3.1 Requisitos Funcionales**

Los requisitos son capacidades y condiciones que debe cumplir el sistema. Para dar cumplimiento a los Objetivos planteados en el presente trabajo, se enumerarán mediante requisitos funcionales, las funciones que el sistema debe ser capaz de cumplir:

#### **RF AC\_1 Crear un nuevo Guión de Contenido.**

RF AC\_1.1 Definir atributos del Guión de Contenido.

#### **RF AC\_2 Gestionar Guión de Contenido.**

RF AC\_2.1 Exportar un Guión de Contenido.

RF AC\_2.2 Editar atributos del Guión de Contenido (Editar)

RF AC\_2.3 Mostrar propiedades.

#### **RF AC\_3 Gestión de la Taxonomía de un Guión de Contenido.**

RF AC\_3.1 Adicionar un elemento de contenido de la taxonomía.

RF AC\_3.1.1 Definir atributos del elemento de contenido.

RF AC\_3.1.2 Numerar nivel taxonómico en profundidad.

RF AC\_3.1.3 Nombrar cada nivel taxonómico.

RF AC\_3.1.4 Definir tipo de elemento de contenido

RF AC\_3.2 Editar un elemento de la taxonomía.

RF AC\_3.3 Eliminar un elemento de la taxonomía.

RF AC\_3.4 Gestionar la Navegación entre los elementos de la taxonomía.

RF AC\_3.5 Importar un elemento de contenido a la taxonomía.

RF AC\_3.6 Comprobar la navegación de elemento de la taxonomía.

#### **RF AC\_4 Gestionar Plantilla de Objetos de Aprendizaje (OA).**

RF AC\_4.1 Adicionar Plantillas de OA.

RF AC\_4.2 Editar Plantillas de OA.

RF AC\_4.3 Eliminar Plantillas de OA.

### **RF AC\_5 Gestionar Plantilla de Niveles Taxonómicos (Elemento Contenedor).**

RF AC\_5.1 Agregar un elemento a la Plantilla de Niveles Taxonómicos (Elemento Contenedor).

RF AC\_5.2 Editar un elemento de la Plantilla de Niveles Taxonómicos (Elemento Contenedor).

RF AC\_5.3 Eliminar un elemento de la Plantilla de Niveles Taxonómicos (Elemento Contenedor).

### **RF AC\_6 Crear Paquete Modelo de Estudiante**

#### **RF AC\_7 Gestionar Modelo Pedagógico.**

RF AC\_7.1 Crear plantilla de estudiante.

RF AC\_7.2 Editar la plantilla del estudiante.

RF AC\_7.3 Eliminar la plantilla del estudiante.

RF AC\_7.4 Definir evaluación del estudiante.

RF AC\_7.5 Definir salida del estudiante.

RF AC\_7.5.1 Definir nivel.

RF AC\_7.5.2 Definir Especialidad.

### **2.3.2 Requisitos No Funcionales**

Los Requisitos no funcionales son propiedades o cualidades que el producto debe cumplir. Debe pensarse en estas propiedades como las características que hacen al producto atractivo, usable, rápido y confiable.

#### **Usabilidad**

Capacidad para múltiples idiomas. La interfaz de la aplicación debe ser lo más atractiva y clara posible para el usuario final. El sistema debe proveer al usuario ayuda e información.

### **Portabilidad**

Necesidad de que el sistema sea multiplataforma. El sistema debe funcionar en sistemas de la familia GNU/Linux.

### **Confiabilidad**

Garantía de un tratamiento adecuado de las excepciones y validación de las entradas del usuario.

### **Requisitos de Implementación**

Emplear QT-Designer para crear aplicaciones clientes de cualquier tipo, usando como biblioteca las librerías de QT, para desarrollar Interfaces Gráficas de Usuarios (GUI).

### **Requisitos de Hardware**

Para el desarrollo: Pentium 600 MHz o superior con 128 MB de memoria RAM y 20 GB de disco duro. Para la explotación: las PC Clientes deben ser Pentium de 133 MHz o superior con 128 MB de memoria RAM mínima, 256 MB de memoria RAM recomendada.

### **Requerimiento asociados al Licenciamiento**

Se debe garantizar que el sistema sea software libre y, por tanto, cualquier componente software que se utilice también deberá ser libre.

### **Requerimiento de Apariencia o interfaz externa**

Se debe garantizar que los colores de la interfaz de la aplicación sean claros. La interfaz debe ser de fácil comprensión en su funcionamiento permitiendo la utilización del sistema sin mucho entrenamiento.

### **Requerimiento de Interfaces de Usuario**

La aplicación debe proveer al usuario una gran versatilidad en la presentación de la información en las pantallas, en cuanto a disponibilidad de los recursos y organización de los mismos.

Permitir mostrar opciones estándares (copiar, cortar, pegar...), pudiendo acceder mediante el clic derecho sobre un componente o por la combinación de teclas calientes.

### 2.4 Interfaz de Comunicación con el Usuario

Los prototipos de interfaz de usuario nos ayudan a comprender y especificar las interacciones entre los actores humanos y sistema durante la captura de los requisitos. No sólo nos ayuda a desarrollar una interfaz gráfica mejor, sino también a comprender mejor los casos de usos. A la hora de especificar la interfaz gráfica de usuario también pueden utilizarse a otros artefactos, como los modelos de interfaz gráfica y los esquemas de pantalla. (RUMBAUGH, y otros, 2000.)

Este posee contadas ventajas que determina su sobrada importancia:

- Los prototipos son fácilmente modificables y prescindibles.
- El prototipo mejora la relación entre desarrolladores y usuarios.
- Los usuarios obtienen mayor satisfacción con los sistemas desarrollados por prototipado.
- Un prototipo puede proporcionar la prueba conceptual necesaria para la consolidación en el proceso.
- Un prototipo puede servir como base para especificaciones operacionales.

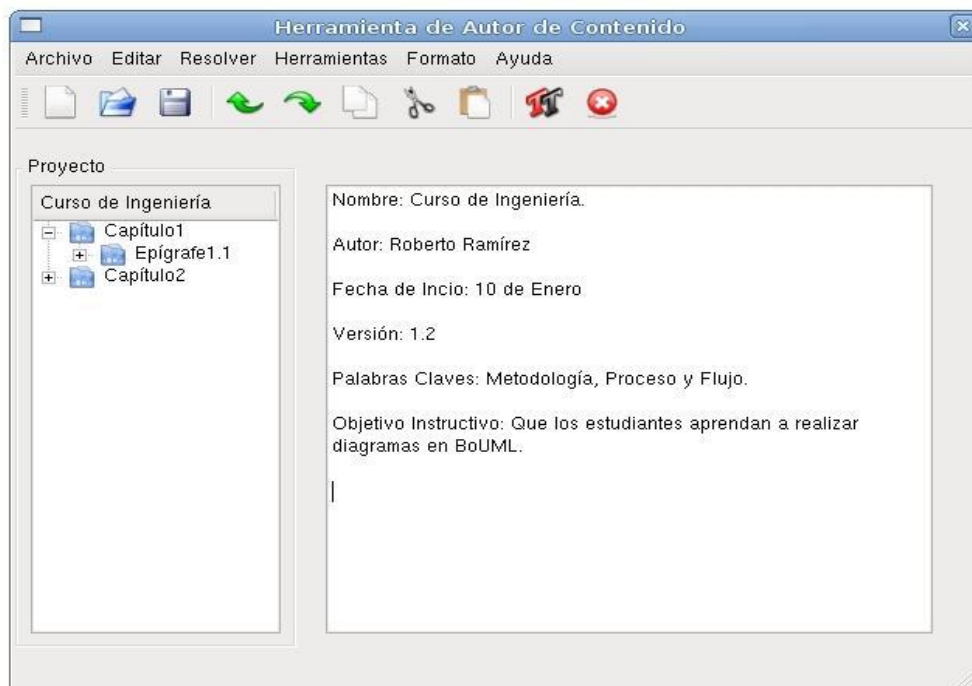


Fig 6. Ejemplo Interfaz Comunicación Usuario.

### 2.5 Modelos de Casos de Uso del Sistema.

Una vez capturados los requisitos del sistema, se representarán mediante un Diagrama de Casos de Uso. Para ello primero se debe definir claramente cuáles son los actores o el actor que van a interactuar con el sistema, y los Casos de Uso que representarán todas las funcionalidades del sistema. Un Caso de Uso “es una descripción de la secuencia de interacciones que se producen entre un actor y el sistema, cuando el actor usa el sistema para llevar a cabo una tarea específica” (RUMBAUGH, y otros, 2000). Un actor representa un conjunto coherente de roles que juegan los usuarios de los Casos de Uso cuando interactúan con éstos. Los actores pueden ser personas (roles que desempeñan las personas), aparatos eléctricos o mecánicos, y otros sistemas de cómputo. En este caso con el sistema interactúa un solo actor: Experto (Especialista en Contenidos).

Actor del Sistema	Justificación
Experto (Especialista en Contenidos).	Es quien interactúa con el sistema. Es el encargado de montar el curso, apoyado por las funcionalidades brindadas por la aplicación.

**Fig 7. Tabla** Actor del sistema.

#### 2.5.1 Diagrama de Casos de Uso del Sistema

Los casos de uso son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación. El diagrama de casos de uso del sistema realizado se define a continuación.

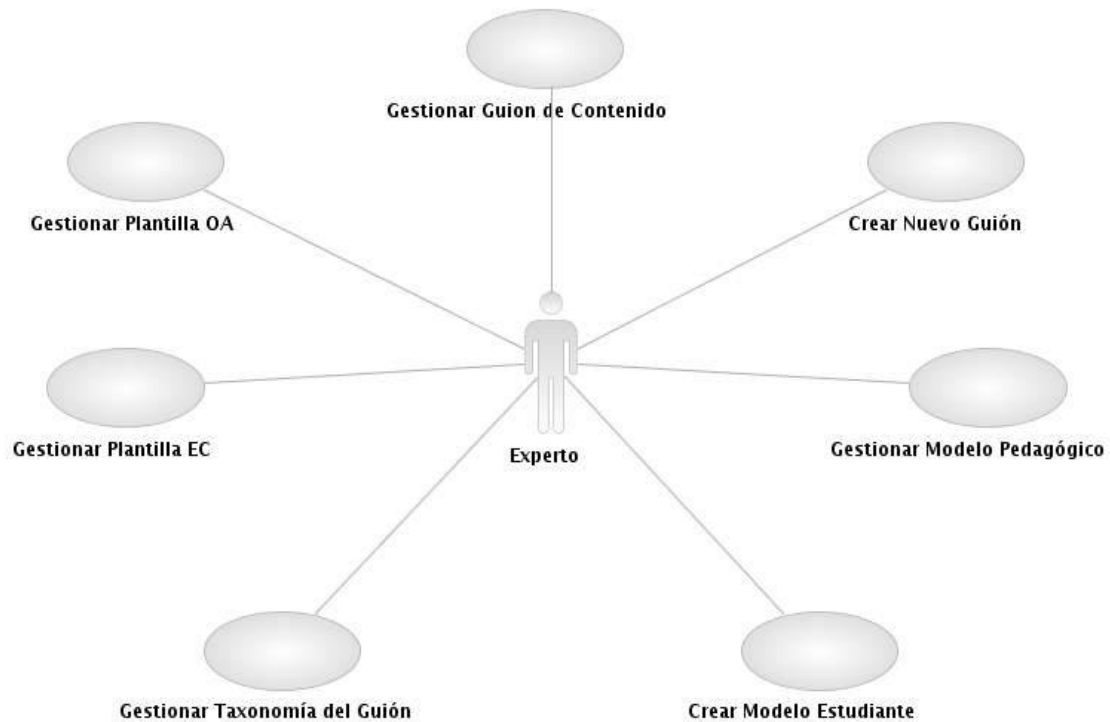


Fig 8. Diagrama de Casos de Uso del Sistema

### 2.5.2 Descripción de los Casos de Usos

El objetivo principal de detallar cada caso de uso es describir su flujo de sucesos en detalle, incluyendo cómo comienza, termina e interactúa con el actor (RUMBAUGH, y otros, 2000.).

Con el propósito de lograr una mejor comprensión de cada fragmento de funcionalidad del sistema se realizó una descripción textual de cada caso de uso.

A continuación se muestra una especificación de alto nivel de cada caso de uso, para dar una idea general de las características de cada uno:

#### 1. Caso de uso crear un nuevo guión de contenido.

<b>Caso de Uso:</b>	<b>Crear un nuevo Guión de Contenido</b>
<b>Actor(es):</b>	Experto (inicia)
<b>Propósito:</b>	Permite la creación de un nuevo Guión de Contenido, mediante plantillas.
<b>Resumen:</b>	El caso de uso se inicia cuando decide crear su guión, accediendo a la opción Nuevo Guión. El Experto llena los datos a llenar del guión y selecciona la plantilla sobre la que trabajará. El caso de uso termina cuando se adiciona el guión al árbol de taxonomía.

## *Capítulo 2: Características del Sistema*

<b>Referencias:</b>	RF_AC 1
<b>Precondiciones:</b>	Para poder seleccionar crear un nuevo guión de contenido a partir de plantillas, es necesario que estas existan previamente.
<b>Post-condiciones</b>	El guión se adicionó al árbol de taxonomía y ha sido creado.
<b>Flujo Normal de Eventos (Ver Anexo 1)</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
1. El experto selecciona la opción de Archivo/Nuevo Guión	1.1. El sistema muestra una ventana de nuevo guión, mostrando los datos generales (nombre, autor, fecha de inicio, fecha de terminación) a llenar por parte del Experto.
2. El Experto llena los datos y da clic en Siguiente.	2.1. El sistema carga en la misma ventana las opciones de selección las plantillas (plantilla vacía, plantilla predefinida, plantilla existente) que el Experto debe escoger para montar el guión.
3. El Experto escoge la plantilla Vacía da clic en Aceptar.	3.1 El sistema actualiza la interfaz de usuario con la plantilla Vacía de Nuevo Guión.
4. El Experto da clic sobre la opción Finalizar.	4.1 El sistema crea la plantilla donde se guardará el guión, cierra la ventana y muestra el guión creado en el árbol de taxonomía, y en el área de trabajo la plantilla con los datos.
<b>Flujos Alternos de los Eventos</b>	
Acción 3	3.1 El Experto escoge la Plantilla Predefinida y el sistema muestra una ventana con las Plantillas Predefinidas.  3.2 El Experto escoge la plantilla con la que desea trabajar, y el sistema actualiza la interfaz de usuario con la Plantilla Predefinida de Nuevo Guión.
Acción 3	3.1 El Experto escoge la Plantilla Existente y el sistema muestra una ventana de Abrir Plantilla.  3.2 El Experto localiza la plantilla y la selecciona y el sistema actualiza la interfaz de Nuevo Guión con la Plantilla existente seleccionada.



## Capítulo 2: Características del Sistema

Acción 4	Si el experto selecciona la opción Anterior, el sistema vuelve al estado anterior en el caso de uso.
Acción 4	Si el experto selecciona la opción Cancelar, el sistema vuelve al estado inicial del caso de uso.
<b>Prioridad</b>	Crítico

### 2. Caso de uso Gestionar Guión de Contenido

Caso de Uso:	<b>Gestionar Guión de Contenido</b>
Actor(es):	Experto (inicia)
Propósito:	Permitir al Experto Exportar o Editar el Guión de Contenido.
Resumen:	El caso de uso se inicia cuando el Experto decide gestionar su guión, accediendo a las opciones que lo permiten. El Experto Exporta, Edita y lee Propiedades, el caso de uso termina cuando el Experto Exporta, Editar o lee Propiedades del guión.
Referencias:	RF_AC 2
Precondiciones:	Debe estar creado el guión de contenido.
Post-condiciones	El guión ha sido exportado y modificado.
Prioridad	Crítico.

#### Flujo Normal de Eventos (Ver Anexo 2)

Acción del Actor	Respuesta del Sistema
1. El Experto selecciona la opción Exportar o Editar el Guión de contenido o Propiedades.	<p>1.1. El sistema realiza una de estas acciones:</p> <p>a) Si selecciona Exportar el guión ir a la sección “Exportar Guión”</p> <p>b) Si selecciona Editar los atributos del guión ir a la sección “Editar Guión”</p> <p>c) Si selecciona Propiedades ir a la sección “Mostrar Propiedades”</p>
<b>Sección 1: Exportar Guión</b>	
<b>Flujo Normal de Eventos</b>	
Acción del Actor:	Respuesta del Sistema

## Capítulo 2: Características del Sistema

	2.1 El sistema muestra una ventana donde el Experto guardará la plantilla del Guión y el formato en que lo desea (doc, .pdf, project, plant, .emm. <b>ac</b> ). ac extensión o formato en como guardará la aplicación.
3. El Experto selecciona la dirección donde guardará la plantilla del Guión y el formato y da clic en Aceptar	3.1 El sistema guarda (se refiere a la dirección donde se exportará el guión) la plantilla en la dirección seleccionada, con el formato seleccionado.  3.2 El sistema carga en la misma ventana el árbol de taxonomía, para ir exportando cada elemento del árbol dentro de la plantilla del guión.
4. El Experto selecciona el elemento de contenido del nivel de taxonomía.	4.1 El sistema marca el elemento taxonómico como seleccionado.
7. El Experto da clic en Aceptar.	7.1 El sistema exporta la plantilla del elemento de contenido, dentro de la plantilla del guión, en forma de árbol. Y muestra un cartel que dice : Desea exportar otro elemento
8. El Experto oprime SI.	8.1 El sistema realiza el mismo procedimiento del flujo 3.2, y así sucesivamente hasta exportar los demás elemento del curso.
<b>Sección 2: Editar Guión</b>	
<b>Flujo Normal de Eventos</b>	
	9.1 El sistema carga una ventana donde muestra los datos del Guión de Contenido a Editar.
10. El Experto edita los datos y guarda los cambios, dando clic en Aceptar.	10.1 El sistema actualiza los cambios.
<b>Sección 3: Mostrar Propiedades</b>	
<b>Flujo Normal de Eventos</b>	
	11.1 El sistema carga una ventana donde muestra los datos del Guión de Contenido para sólo lectura.
12. El Experto lee los datos y da clic en Aceptar	12.1 El sistema cierra la ventana.
<b>Flujos Alternos de los Eventos</b>	

## Capítulo 2: Características del Sistema

Acción 3,7,10,12	Si el Experto selecciona la opción Cancelar, el sistema vuelve al estado inicial del caso de uso.
Acción 8	El Experto selecciona NO y no exporta las plantillas de los demás elementos.

### 3. Caso de Uso Gestionar Taxonomía del Guión de Contenido

Caso de Uso:	<b>Gestionar taxonomía del Guión de Contenido.</b>
Actor(es):	Experto (inicia)
Propósito:	Permitir al Experto crear, eliminar, editar, comprobar navegación, definir navegación.
Resumen:	El caso de uso se inicia cuando el Experto decide gestionar la taxonomía del guión. El Experto crea, elimina, edita, comprueba la navegación, define navegación. El Caso de uso termina cuando el Experto Crea, Importa, Elimina, Edita, Define navegación que tendrá el estudiante por el curso en dependencia de sus evaluaciones, y Comprueba la navegación que ha tenido el estudiante una vez que ha sido evaluado por el curso.
Referencias:	RF_AC 3
Precondiciones:	Debe estar creado el guión.
Post-condiciones	El elemento de contenido se creó, editó, eliminó, etc a la taxonomía.
Prioridad	Crítico.
<b>Flujo Normal de Eventos (Ver Anexo 3)</b>	
Acción del Actor	Respuesta del Sistema
1. El Experto selecciona Crear, Importar, Eliminar, Editar, Comprobar Nav o Definir Nav, del elemento taxonómico.	<p>1.1. El sistema realiza una de estas acciones:</p> <p>a) Si decide crear el elemento a la taxonomía ir a la sección "Crear Elemento"</p> <p>b) Si decide Editar el elemento de contenido ir a la sección "Editar Elemento"</p> <p>c) Si decide Eliminar el elemento de contenido ir a la sección "Eliminar Elemento"</p> <p>d) Si decide navegación ir a la sección "Definir Navegación entre los elementos de la taxonomía"</p>

	<p>f) Si decide navegación importar un elemento de contenido a la taxonomía “Importar Elemento”</p> <p>g) Si decide la opción comprobar ir a la sección “Comprobar la navegación de elemento de la taxonomía.”</p>
<b>Sección 1: Crear Elemento</b>	
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	2.1 El sistema carga una ventana de Crear elemento, con formularios de datos a llenar y escoger del árbol de taxonomía que se muestra en la ventana el elemento de contenido sobre el que desea crear el elemento. Si es el primer elemento a adicionar, el árbol se mostrará en blanco.
3. El Experto selecciona sobre que elemento desea adicionar el elemento.	3.1 El sistema marca el elemento como seleccionado.
4. El Experto da clic en Aceptar	4.1 El sistema crea la plantilla con el elemento de contenido y lo muestra en el árbol de taxonomía.
<b>Sección 2: Editar Elemento</b>	
<b>Flujo Normal de Eventos</b>	
	5.1 El sistema carga una ventana con formularios a editar por parte del Experto, permitiéndole modificar los datos del elemento.
6. El Experto modifica los datos seleccionados por él. Y da clic en Aceptar.	6.1 El sistema actualiza los cambios.
<b>Sección 3: Eliminar Elemento</b>	
<b>Flujo Normal de Eventos</b>	
	7.1 El sistema carga una ventana con el árbol de taxonomía para que el Experto seleccione el elemento a eliminar.
8. El Experto marca el elemento.	<p>8.1 El sistema muestra el elemento como seleccionado.</p> <p>8.2 El sistema carga en la misma ventana un cartel que dice: “Se eliminarán los elementos hijos del</p>

	elemento de contenido”.
9. El Experto da clic en Eliminar	9.1 El sistema elimina el elemento de contenido de la taxonomía.
<b>Sección 4: Definir Navegación entre elementos de taxonomía</b>	
<b>Flujo Normal de Eventos</b>	
	10.1 El sistema carga una ventana con las Precondiciones y las Acciones para la navegación a definir por parte del Experto, la cual estará definida por las reglas de navegación definidas al crear los elementos de contenido.
11. El Experto selecciona Precondiciones.	11.1 El sistema muestra las Precondiciones con los atributos del estudiante y el valor de lo mismos.
12. El experto define los atributos y da clic en Acciones.	12.1 El sistema carga en la ventana las Acciones, donde le Experto a partir de las Precondiciones, especifica cómo va a ser el camino que recorrerá el estudiante dentro del elemento de contenido.
13. El Experto define las Acciones de navegación, y guarda los cambios, dando clic en Aceptar	13.1 El sistema crea la plantilla con la navegación definida y cierra la ventana.
<b>Sección 5: Importar un elemento de contenido a la taxonomía</b>	
<b>Flujo Normal de Eventos</b>	
	14.1 El sistema muestra una interfaz de búsqueda que permite al Experto localizar en el directorio que se encuentra la plantilla del elemento a importar. Para ello filtra los archivos que tenga la misma extensión de la aplicación.
15. El Experto localiza el archivo y lo selecciona.	15.1 El sistema lo marca como seleccionado.
16. El Experto da clic en Importar.	16.1 El sistema importa el elemento y lo guarda en una plantilla y lo muestra en el árbol de taxonomía.
<b>Sección 6: Comprobar la navegación de elemento de la taxonomía</b>	
<b>Flujo Normal de Eventos</b>	
	17.1 El sistema carga una ventana donde el Experto puede ver la navegación desempeñada en un futuro por el estudiante en el curso, esto es variable depende de las reglas de navegación determinadas por el Experto.

18. El Experto comprueba la navegación y da clic en Aceptar.	18.1 El sistema cierra la ventana y termina así el caso de uso.
<b>Flujos Alternos de los Eventos</b>	
Acción 4,9,13,16,18	Si el Experto selecciona la opción Cancelar, el sistema vuelve al estado inicial del caso de uso.
Acción 6	El Experto da clic en Siguiente y el sistema vuelva al estado anterior del caso de uso.

#### 4. Caso de uso Gestionar Plantilla de OA

Caso de Uso:	<b>Caso de uso Gestionar Plantilla de OA</b>
Actor(es):	Experto (inicia)
Propósito:	Permitir a al Experto Adicionar, Editar, Eliminar.
Resumen:	El caso de uso comienza cuando el Experto decide gestionar la Plantilla del OA, accediendo a las opciones que brinda. El Experto puede Adicionar, Editar o Eliminar una plantilla del OA a la Taxonomía. El caso de uso termina cuando el Experto realiza una de estas acciones.
Referencias:	RF_AC 4
Precondiciones:	Debe existir una plantilla de OA.
Post-condiciones	La plantilla de OA ha sido adicionada, editada o eliminada.
Prioridad	Crítico.
<b>Flujo Normal de Eventos (Ver Anexo 4)</b>	
Acción del Actor	Respuesta del Sistema
1. El Experto selecciona la opción Adicionar, Eliminar o Editar, o da clic derecho sobre el elemento.	<p>1.1 El sistema realiza una de estas acciones:</p> <p>a) Si decide adicionar ir a la sección “Adicionar Plantilla de OA a la taxonomía”</p> <p>b) Si decide editar ir a la sección “Editar Plantilla de OA a la taxonomía”</p> <p>c) Si decide eliminar ir a la sección “ Eliminar Plantilla de OA a la taxonomía”</p>
<b>Sección 1: Adicionar Plantilla de OA</b>	
<b>Flujo Normal de Eventos</b>	

## *Capítulo 2: Características del Sistema*

Acción del Actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz de búsqueda que permite al Experto localizar en el directorio que se encuentran las plantillas de los OA.
3. El Experto localiza la plantilla y la selecciona.	3.1 El sistema marca la plantilla como seleccionada,  3.2 Muestra en la misma ventana un árbol de taxonomía, para que el experto seleccione sobre qué elemento desea adicionar la plantilla.
4. El Experto selecciona sobre qué elemento adicionar la plantilla y da clic en Aceptar.	4.1 El sistema actualiza el área de trabajo mostrando la plantilla seleccionada, y la muestra además adicionada en el árbol de taxonomía.
<b>Sección 2: Editar Plantilla de OA</b>	
<b>Flujo Normal de Eventos</b>	
	5.1 El sistema muestra una ventana que brinda la posibilidad de modificar los datos.
6. El Experto modifica los datos y los guarda. Da clic en Aceptar.	6.1 El sistema actualiza los cambios.
<b>Sección 3: Eliminar Plantilla de OA</b>	
<b>Flujo Normal de Eventos</b>	
7.El Experto selecciona la opción de Eliminar Plantilla OA	7.1 El sistema carga una ventana con el listado de plantillas de OA, para que el Experto seleccione la que desea eliminar.
8. El Experto selecciona la Plantilla.	8.1 El sistema marca la plantilla como seleccionada.
9. El Experto da clic en Aceptar.	9.1 El sistema elimina la plantilla seleccionada.
<b>Flujos Alternos de los Eventos</b>	
Acción 4,6,9	Si el Experto selecciona la opción Cancelar, el sistema vuelve al estado inicial del caso de uso.

### **5. Caso de uso Gestionar Plantilla de Niveles Taxonómicos (Elemento Contenedor).**

Caso de Uso:	<b>Caso de uso Gestionar Plantilla de Elemento Contenedor.</b>
--------------	--

Actor(es):	Experto (inicia)
Propósito:	Permitir a al Experto Adicionar, Editar, Eliminar.
Resumen:	El caso de uso comienza cuando el Experto decide gestionar la Plantilla de Niveles Taxonómicos, accediendo a las opciones que brinda. El Experto puede Adicionar, Editar o Eliminar una plantilla de Niveles Taxonómicos. El caso de uso termina cuando el Experto Adiciona, Edita o Elimina una plantilla de Niveles Taxonómicos.
Referencias:	RF_AC 5
Precondiciones:	Debe existir una plantilla de Elemento Contenedor.
Post-condiciones	La plantilla de Niveles Taxonómicos ha sido adicionada, editada o eliminada.
Prioridad	Crítico.
<b>Flujo Normal de Eventos (Ver Anexo 5)</b>	
Acción del Actor	Respuesta del Sistema
1. El Experto selecciona la opción Adicionar, Eliminar o Editar, o da clic derecho sobre el elemento.	<p>1.1 El sistema realiza una de estas acciones:</p> <p>a) Si decide adicionar ir a la sección “Adicionar Plantilla Niveles Taxonómicos”</p> <p>b) Si decide editar ir a la sección “Editar Plantilla Niveles Taxonómicos”</p> <p>c) Si decide eliminar ir a la sección “Eliminar Plantilla Niveles Taxonómicos”</p>
<b>Sección 1 : Adicionar Plantilla de Niveles Taxonómicos</b>	
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	2.1 El sistema muestra una interfaz de búsqueda que permite al Experto localizar en el directorio que se encuentran las plantillas de los Elementos Contenedores.
3. El Experto localiza y selecciona la plantilla.	<p>3.1 El sistema marca la plantilla como seleccionada.</p> <p>3.2 Muestra en la misma ventana un árbol de taxonomía con los elementos de contenido, sobre los que el Experto deseara adicionar la plantilla.</p>
4. El Experto selecciona sobre qué elemento adicionar la plantilla y da	4.1 El sistema adiciona la plantilla EC, a la plantilla del guión y la muestra como adicionada en el árbol de taxonomía.



clic en Aceptar.	
<b>Sección 2: Editar Plantilla de Niveles Taxonómicos</b>	
<b>Flujo Normal de Eventos</b>	
	5.1 El sistema muestra una ventana que brinda la posibilidad de modificar los datos.
6. El Experto modifica los datos y los guarda. Da clic en Aceptar.	6.1 El sistema actualiza los cambios.
<b>Sección 3: Eliminar Plantilla de Niveles Taxonómicos</b>	
<b>Flujo Normal de Eventos</b>	
7.El Experto selecciona la opción de Eliminar Plantilla EC	7.1 El sistema carga una ventana con el listado de plantillas de EC, para que el Experto seleccione la que desea eliminar.
8. El Experto selecciona la Plantilla.	8.1 El sistema marca la plantilla como seleccionada. 8.2 Muestra un cartel que dice: “Se eliminarán los elementos hijos”.
9. El Experto da clic en Aceptar.	9.1 El sistema elimina la plantilla seleccionada.
<b>Flujos Alternos de los Eventos</b>	
Acción 4,6,9	Si el Experto selecciona la opción Cancelar, el sistema vuelve al estado inicial del caso de uso.

### 6. Caso de uso Crear Paquete Modelo Estudiante.

Caso de Uso:	<b>Caso de uso Crear Paquete Modelo Estudiante.</b>
Actor(es):	Experto (Inicia)
Propósito:	Permitir crear la carpeta del Estudiante.
Resumen:	El caso de uso comienza cuando el Experto decide Crear el paquete Estudiante. Esta se mostrará en el árbol de taxonomía y se guardará en la plantilla Modelo Estudiante. El caso de uso termina cuando el Experto crea el paquete Estudiante.
Referencias:	RF_AC 6
Precondiciones:	Debe haber creado al menos un guión.

## Capítulo 2: Características del Sistema

Post-condiciones	La carpeta de estudiante fue creada.
Prioridad	Crítico.
<b>Flujo Normal de Eventos (Ver Anexo 6)</b>	
Acción del Actor	Respuesta del Sistema
1.El Experto selecciona la opción Crear Modelo Estudiante, o da clic derecho sobre el Guión.	1.1 El sistema carga una ventana con formularios a llenar por el Experto.
2. El Experto llena los datos y da clic en Aceptar.	2.1 El sistema guarda la Carpeta con su nombre en la carpeta Guión, y la muestra en el árbol de taxonomía, al mismo nivel del curso.
<b>Flujos Alternos de los Eventos</b>	
Acción 2	Si el Experto selecciona la opción Cancelar, el sistema vuelve al estado inicial del caso de uso.

### 7. Caso de Uso Gestionar Modelo Pedagógico.

Caso de Uso:	<b>Caso de uso Gestionar Modelo Pedagógico.</b>
Actor(es):	Experto (inicia)
Propósito:	Permitir trabajar con el estudiante.
Resumen:	El caso de uso comienza cuando el Experto decide gestionar su modelo pedagógico, para ello adiciona, elimina, edita la plantilla del estudiante, además de definir la salida del estudiante de un curso y definir su evaluación. El caso de uso termina cuando el Experto realiza una de estas acciones.
Referencias:	RF_AC 7
Precondiciones:	Debe estar creada la carpeta de Estudiante.
Post-condiciones	La plantilla de Estudiante fue adicionada, eliminada, editada. Se definió la salida del estudiante, etc.
Prioridad	Crítico
<b>Flujo Normal de Eventos (Ver Anexo 7)</b>	
Acción del Actor	Respuesta del Sistema
1.El experto selecciona la opción Adicionar, Editar, Eliminar, Definir Salida,	1.1. El sistema carga una ventana con las siguientes opciones:

etc, o da clic derecho sobre la Carpeta Estudiante.	<p>a) Si selecciona Adicionar ir a la sección “Adicionar Plantilla Estudiante”</p> <p>b) Si selecciona Editar ir a la sección “Editar Plantilla Estudiante”</p> <p>c) Si selecciona Eliminar ir a la sección “Eliminar Plantilla Estudiante”</p> <p>d) Si selecciona Evaluación ir a la sección “Evaluación Estudiante”</p> <p>e) Si selecciona Salida ir a la sección “Salida Estudiante”</p>
<b>Sección 1: Adicionar Plantilla Estudiante</b>	
<b>Flujo Normal de Eventos</b>	
Acción del Actor	Respuesta del Sistema
	2.1 El sistema muestra una ventana con las opciones de plantillas (Vacía, Plantilla Predefinida, Plantilla Existente) con las que el Experto trabajará.
3. El Experto selecciona la plantilla Vacía.	3.1 El sistema actualiza la interfaz de usuario mostrando la plantilla seleccionada.
4. El Experto da clic en Aceptar.	4.1 El sistema adiciona la plantilla seleccionada y la muestra en el árbol de taxonomía.
<b>Sección 2: Editar Plantilla Estudiante</b>	
<b>Flujo Normal de Eventos</b>	
	5.1 El sistema carga una ventana de Editar con un listado para que el Experto seleccione la plantilla del estudiante que desea Editar.
6 El Experto selecciona la Plantilla del Estudiante y da clic en Aceptar	6.1 El sistema carga en la misma ventana los datos a editar de la plantilla seleccionada.
7. El Experto modifica los datos y da clic en Aceptar.	7.1 El sistema actualiza los cambios.
<b>Sección 3: Eliminar Plantilla Estudiante</b>	
<b>Flujo Normal de Eventos</b>	
	8.2 El sistema muestra una ventana con las plantillas a Eliminar.
9. El Experto selecciona la plantilla a Eliminar y da	9.1 El sistema elimina la plantilla seleccionada, y actualiza los cambios, eliminándola de la taxonomía.

clic en Aceptar.	
<b>Sección 4: Evaluación Estudiante</b>	
<b>Flujo Normal de Eventos</b>	
	10.1 El sistema carga en el área de trabajo la plantilla Vacía para que el Experto cree la evaluación.
11. El Experto define la Evaluación y da clic en Guardar.	11.1 El sistema muestra una ventana donde muestra un árbol de taxonomía con los elementos de contenido para que el Experto seleccione sobre qué elemento se añadirá la evaluación.
12. El Experto selecciona el elemento y da clic en Aceptar.	12.1 El sistema adiciona la plantilla de evaluación en la Paquete Estudiante del árbol de taxonomía, con los datos del elemento de contenido donde se pondrá la evaluación.
<b>Sección 5: Salida Estudiante</b>	
<b>Flujo Normal de Eventos</b>	
	13.1 El sistema carga una ventana en la parte izquierda con los datos del estudiante que lo identifican en su navegación por el curso en la futura aplicación y en la parte derecha las categorías.
14. El Experto en dependencia de datos y trayectoria del estudiante, y selecciona categoría.	14.1 El sistema marca la categoría como seleccionada.
15. El Experto da clic en Aceptar	16.1 El sistema da salida al estudiante del curso, con la categoría seleccionada.
<b>Flujos Alternos de los Eventos</b>	
Acción 3	3.1 El Experto escoge la Plantilla Predefinida y el sistema muestra una ventana con las Plantillas Predefinidas.  3.2 El sistema actualiza el área de trabajo y adiciona la plantilla a la taxonomía.
Acción 3	3.1 El Experto selecciona la opción Plantilla Existente y el sistema carga una ventana con los directorios a localizar la plantilla.  3.2 EL sistema carga en el área de trabajo la plantilla seleccionada y la agrega a la taxonomía
Acción 4,6,7,9,12,15	Si el Experto selecciona la opción Cancelar, el sistema vuelve al estado inicial del caso de uso.

### Capítulo 3: Solución Propuesta

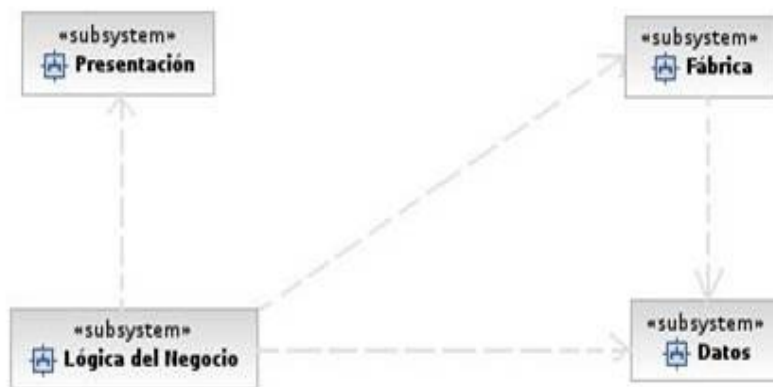
#### Introducción

En este capítulo se modelan los principales artefactos que ayudan a manejar las dificultades que implican la construcción de aplicaciones de este tipo. Para su estudio más detallado se realizarán por cada caso de uso, los correspondientes diagramas de clases del análisis y diseño, además de los diagramas de interacción de este último.

#### 3.1 Vista Gestión Modelo

*“La vista de gestión del modelo modela la organización del modelo en sí mismo. Un modelo es una descripción completa de un sistema, con una determinada precisión, desde un punto de vista. Puede haber varios modelos de un sistema desde distintos puntos de vista; por ejemplo, un modelo de análisis y un modelo de diseño. Un modelo se representa como una case especial de un paquete.”* (James Rumbaugh, 2000)

*A continuación se presenta la vista de gestión del modelo que muestra la descomposición de la totalidad del sistema en paquetes y sus relaciones de dependencia.*



#### 3.2 Diseño Arquitectónico

El diseño es una actividad en la que se toman decisiones importantes, frecuentemente de naturaleza estructural. Comparte con la programación un interés por la abstracción de la información y de las secuencias del procesamiento, pero el nivel de detalle es muy diferente en ambos casos. El diseño constituye representaciones coherentes y bien planificadas de los programas, concentrándose en las interrelaciones de los componentes de mayor nivel y en las operaciones lógicas implicadas en los niveles inferiores. La arquitectura de software es la estructura general de un sistema. Esta comprende los componentes de software, las propiedades de esos componentes

visibles externamente, y las relaciones de ellos. Razones diferentes fundamentan la importancia de la arquitectura de software:

- Las representaciones de la arquitectura de software facilitan la comunicación entre todas las partes interesadas en el desarrollo de un sistema basado en computadoras.
- La arquitectura destaca decisiones tempranas de diseño que tendrán un profundo impacto en todo el trabajo de ingeniería de software, y es tan importante en el resultado final del sistema como una entidad operacional.
- Constituye un modelo comprensible de cómo está estructurado el sistema y de cómo trabajan juntos sus componentes.

### 3.3 Patrón Modelo-Vista-Controlador

La arquitectura del Modelo Vista Controlador (MVC) divide una aplicación interactiva en 3 áreas: procesamiento, salida y entrada. Para esto, utiliza las siguientes abstracciones:

- **Modelo (Model):** Encapsula los datos y las funcionalidades. El modelo es independiente de cualquier representación de salida y/o comportamiento de entrada.
- **Vista (View):** Muestra la información del usuario. Pueden existir múltiples vistas del modelo. Cada vista tiene asociado un componente controlador.
- **Controlador (Controller):** Reciben las entradas, usualmente como eventos que codifican los movimientos o pulsaciones de botones del mouse, pulsaciones de teclas, etc. Los eventos son traducidos a solicitudes de servicios (“services requests”) para el modelo o la vista.

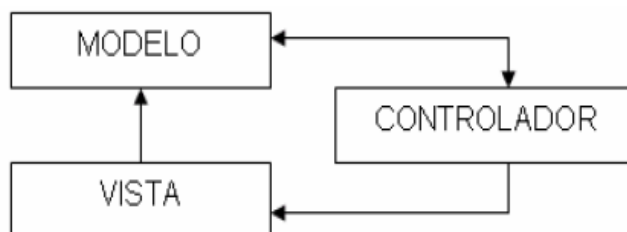
Se han desarrollado a lo largo de los años, desde la presentación de este patrón a la comunidad científica variantes fundamentales, que se presentan brevemente a continuación.

**Variante I:** Variante en la cual no existe ninguna comunicación entre el Modelo y la Vista, y esta última recibe los datos a mostrar a través del Controlador.

**Variante II:** Variante en la cual se desarrolla una comunicación entre el Modelo y la Vista, donde esta última al mostrar los datos los busca directamente en el Modelo, dada una indicación del Controlador, disminuyendo el conjunto de responsabilidades este último.

**Variante III:** Variante en la cual se diversifican las funcionalidades del Modelo teniendo en cuenta las características de las aplicaciones multimedia, donde tiene un gran peso las medias utilizadas en estas. (Ricardo, 2008)

Para la realización de la arquitectura de la presente investigación se utilizó la Variante II del patrón MVC.



**Fig 9. Modelo-Vista-Controlador**

### 3.4 Patrones de Diseño

De manera general los patrones constituyen soluciones simples y elegantes a problemas específicos y comunes del diseño orientado a objetos, basados en la experiencia y que se ha demostrado que funcionan.

Los patrones son parejas de problema/solución con un nombre, que codifican nuevos principios y sugerencias relacionados frecuentemente con la asignación de responsabilidades. Asignar correctamente las responsabilidades es muy importante en el diseño orientado a objetos. (Larman, 2000)

#### 3.4.1 Patrones GOF

Los Patrones GOF generalmente se evidencian en clases que son creadas debido al uso de un patrón específico. Existe un grupo de patrones de este tipo definidos para el diseño y con el propósito de crear una arquitectura robusta para el sistema a desarrollar. Del gran número de patrones propuestos por la pandilla de los cuatro o simplemente GOF (Singleton, Builder, Prototype y Factory), se propone el uso del patrón Fábrica Pura. Este tiene como principal función crear objetos sin importar en qué momento y a qué clase pertenecen.

##### 3.4.1.1 Fábrica Pura

Se le llama fábrica, factoría o factory a una clase que implemente uno o varios métodos de creación, que son los que se encargan de crear instancias de objetos (estas instancias pueden ser de esta misma clase o de otras). Esta clase tiene entre sus responsabilidades la creación de instancias de objetos, pero puede tener también otras responsabilidades adicionales. Los métodos de creación pueden ser estáticos.

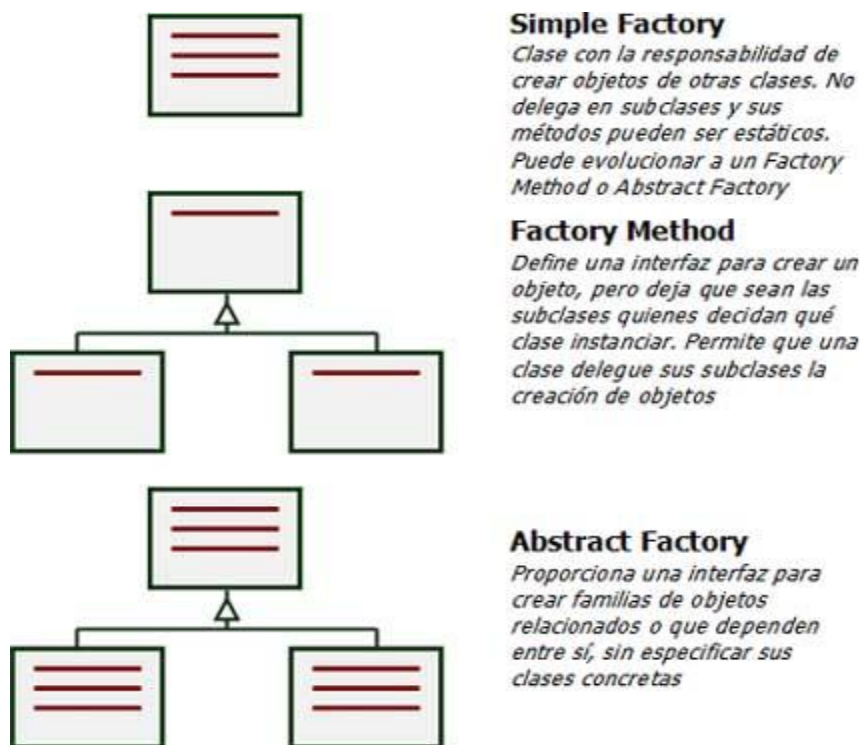
Existen diferentes "tipos" de fábricas. A continuación se enumeran cada una de ellos, a los efectos de brindar con claridad sus significados:

**Simple Factory:** Clase utilizada para crear nuevas instancias de objetos.

**Factory Method:** Define una interfaz para crear objetos, pero deja que sean las subclases las que deciden qué clases instanciar.

**Abstract Factory:** Proporciona una interfaz para crear familias de objetos relacionados o que dependen entre sí, sin especificar sus clases concretas.

En la (Figura 9.1) se muestran los diferentes tipos de factorías con las que trabajaremos en este artículo. En cada caso, cada cuadro representa una clase y cada línea es un método en esa clase:



**Figura 9.1** Tipos de fábricas

Para el desarrollo de la Herramienta Autor de Guiones de Contenido, se escogió el patrón Fábrica para optimizar el proceso de creación de los objetos.

### 3.5 Diagramas de clases del análisis

Los diagramas de clases del análisis expresan la definición y relación entre las clases, están compuestos por clases que pueden ser de tres tipos fundamentales: interfaz, controladora y entidad.

*Las clases interfaz* son las que se encargan de brindar un entorno gráfico amigable y entendible para que el usuario interactúe con el sistema de manera que modelan la interacción entre los actores y el sistema.

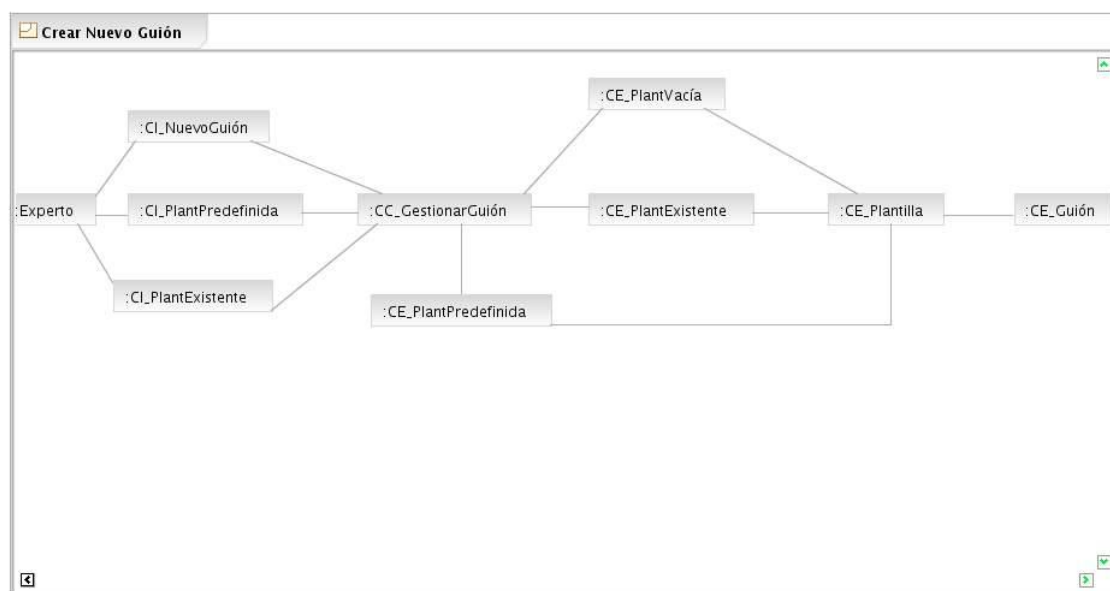


Las clases controladoras se encargan de captar toda la información que es enviada a través de las clases interfaz y guardar esos datos a través de métodos en la Base de Datos (BD), coordinando la realización de uno o unos pocos casos de uso, así como las actividades de los objetos que implementan la funcionalidad de los casos de uso.

Las clases entidades generalmente modelan información que posee larga vida y que es a menudo existente, estas clases son las futuras tablas de la base de datos.

Los diagramas de clases del análisis además contienen relaciones que evidencian la navegabilidad y dependencia entre clases, de esta forma se identifican relaciones de asociación, generalización y especificación, entre otras.

A continuación se muestran los diagramas correspondientes a las clases del análisis:



**Fig 10 Crear Nuevo Guión**

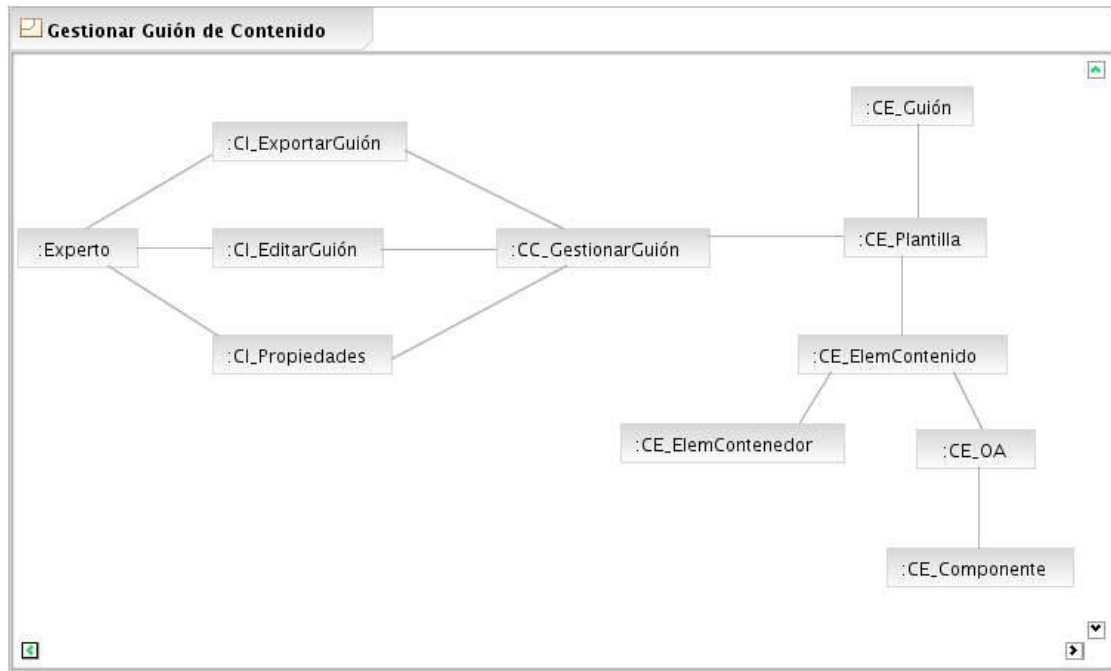


Fig 11 Gestionar Guión de Contenido

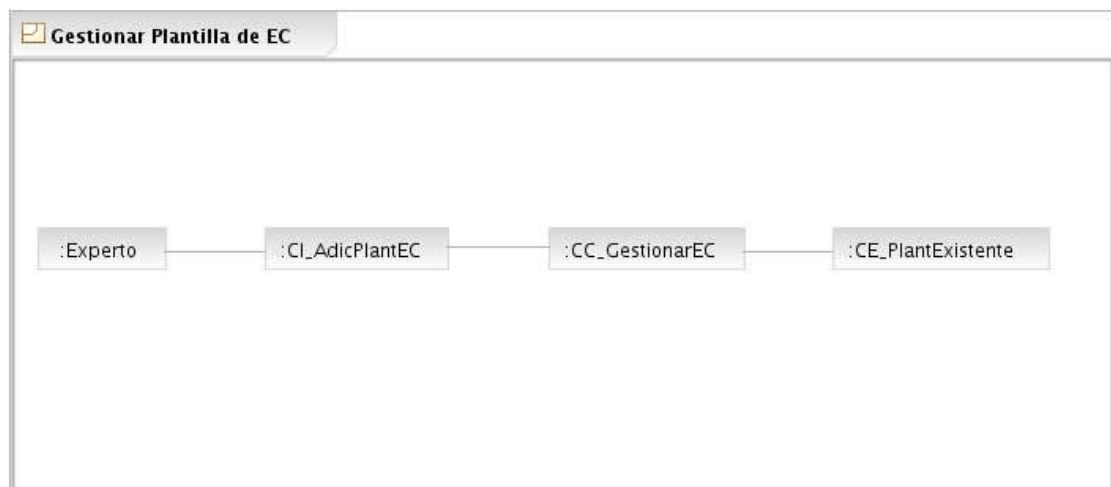


Fig 12 Gestionar Plantilla de EC



Fig 13. Gestionar Plantilla de OA

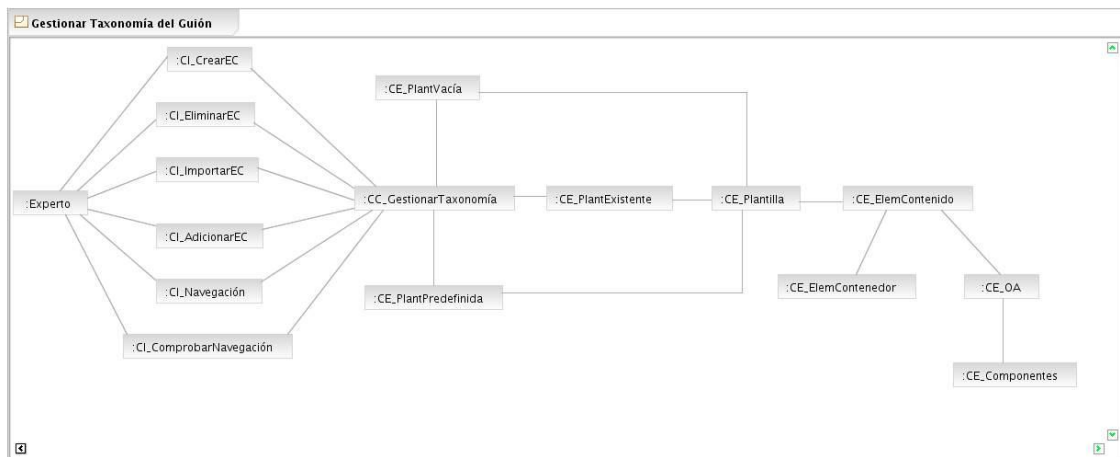


Fig 13. Gestionar Taxonomía del Guión

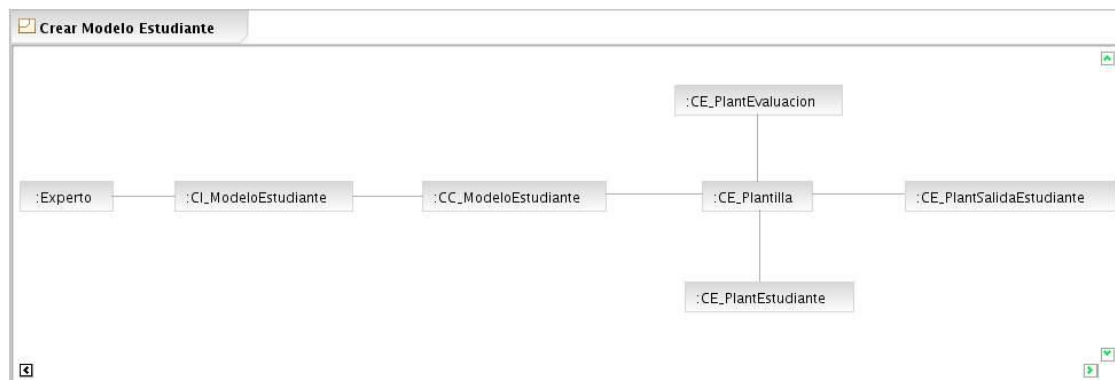


Fig 14 Modelo Estudiante

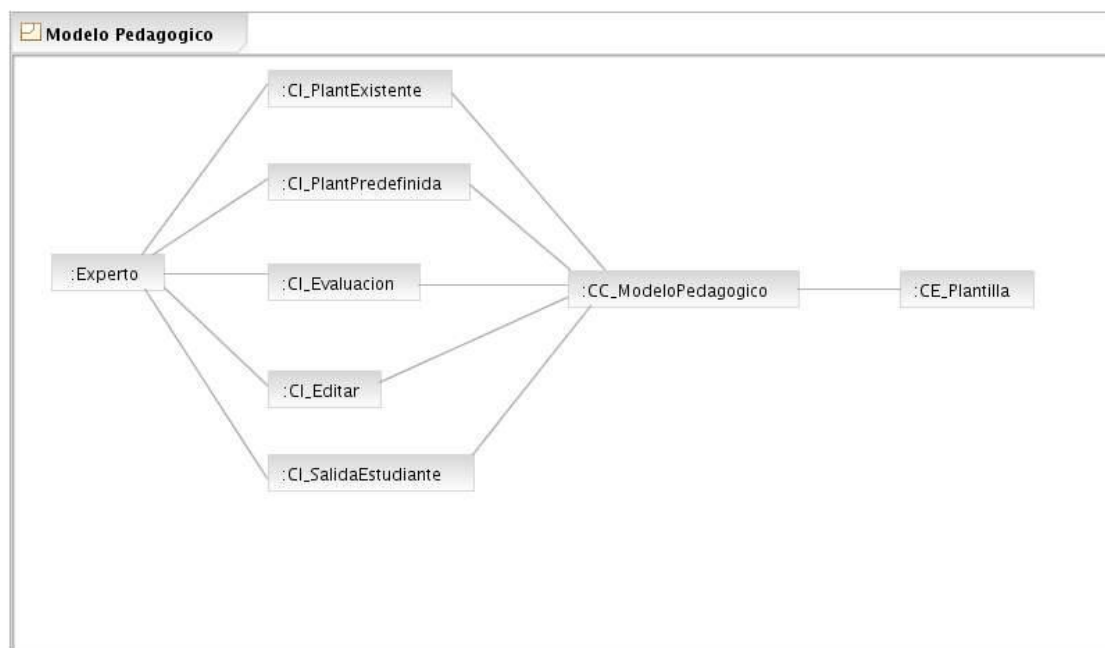


Fig 15 Modelo Pedagógico

### 3.6 Diagramas de clases del diseño

Un diagrama de clases es un tipo de modelo estático que describe la vista estática del sistema. Un propósito de estos diagramas es definir una base para otros diagramas donde otros aspectos del sistema son mostrados (tales como los estados de los objetos o la colaboración entre ellos mostrados en los diagramas dinámicos). En estos se incluyen las clases interfaz y las clases a implementar por los desarrolladores, además de la relación existente entre ellas.

#### 3.6.1 Descripción de clases del Diseño

<b>Nombre : Crear Nuevo Guión</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
EditTitulo	Edit
EditAutor	Edit
EditFecha1	Edit
Version	Edit
TextEditPlantilla	TextEdit

RadioVacía	RadioButton
RadioPredefinida	RadioButton
RadioExistente	RadioButton
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Void CrearGuion()
Descripción:	Función que recoge los datos, y les envía, para crear el guión.
<b>Nombre:</b>	void ButtoSig
Descripción:	Muestra al usuario la clase interfaz siguiente
<b>Nombre:</b>	void ButtoCancelar
Descripción:	Cierra la ventana.

<b>Nombre: CrearNuevoGuión</b>	
<b>Tipo de clase : Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	CrearNuevoGuion()
Descripción:	Función que permite crear un Nuevo Guión

<b>Nombre: CC_RepositorioCrearGuion</b>	
<b>Tipo de clase : Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	AdicionarGuion()
Descripción:	Función que permite adicionar el guión.
<b>Nombre:</b>	EliminarGuion()
Descripción:	Función que permite eliminar el guión.

<b>Nombre :</b>	ModificarGuion()
Descripción:	Función que permite modificar el guión.
<b>Nombre :</b>	SeleccionarGuion()
Descripción:	Función que permite seleccionar el guión.

<b>Nombre : Plantilla</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad :</b>	
<b>Nombre :</b>	GetTipo()
Descripción:	Devuelve el tipo de Plantilla.
<b>Nombre :</b>	GetPlantilla()
Descripción:	Devuelve la Plantilla.

<b>Nombre : Guion</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
Autor	String
Objetivos	String
PalabrasClaves	String
<b>Para cada responsabilidad :</b>	
<b>Nombre :</b>	GetNombre()
Descripción:	Devuelve el nombre del guion.
<b>Nombre :</b>	GetAutor()
Descripción:	Devuelve el nombre del autor.

<b>Nombre :</b>	GetObjetivos()
Descripción:	Devuelve los objetivos del curso.
<b>Nombre :</b>	GetPalabrasClav()
Descripción	Devuelve las palabras claves.

<b>Nombre : EditarGuion</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
EditAutor	Edit
EditFechal	Edit
EditFechaT	Edit
EditVersion	Edit
EditTitulo	Edit
TexEditEducativo	TextEdit
TexEditInstructivo	TextEdit
EditClaves	Edit
MemoClaves	Memo
ButtonNext	Button
<b>Para cada responsabilidad:</b>	
<b>Nombre :</b>	Cancelar()
Descripción:	Cierra la ventana.
<b>Nombre :</b>	Aceptar()
Descripción:	Edita el guión

<b>Nombre : MostrarPropiedades</b>
<b>Tipo de clase : Interfaz</b>

Atributo		Tipo
<b>Para cada responsabilidad:</b>		
<b>Nombre:</b>	Cancelar()	
Descripción:	Se cierra la ventana.	
<b>Nombre:</b>	Aceptar()	
Descripción:	Guarda las Propiedades mostradas	

<b>Nombre: ExportarGuion</b>		
<b>Tipo de clase : Interfaz</b>		
Atributo		Tipo
<b>Para cada responsabilidad:</b>		
<b>Nombre:</b>	Cancelar()	
Descripción:	Se cierra la ventana.	
<b>Nombre:</b>	Exportar()	
Descripción:	Exporta el guión.	

<b>Nombre: GestionarGuion</b>		
<b>Tipo de clase : Controladora</b>		
Atributo		Tipo
<b>Para cada responsabilidad:</b>		
<b>Nombre:</b>	Exportar()	
Descripción:	Exporta el guión	
<b>Nombre:</b>	Editar()	
Descripción:	Edita el guion de Contenido.	
<b>Nombre:</b>	MostrarPropiedades()	
Descripción:	Muestra las propiedades del guión.	



--	--

<b>Nombre : RepositorioGestGuion</b>	
<b>Tipo de clase : Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre :</b>	AdicionarGuion()
Descripción:	Función que permite adicionar el guión.
<b>Nombre :</b>	EliminarGuion()
Descripción:	Función que permite eliminar el guión.
<b>Nombre :</b>	ModificarGuion()
Descripción:	Función que permite modificar el guión.
<b>Nombre :</b>	SeleccionarGuion()
Descripción:	Función que permite seleccionar el guión.

<b>Nombre : Plantilla</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre :</b>	GetTipo()
Descripción:	Devuelve el tipo de Plantilla.
<b>Nombre :</b>	GetPlantilla()
Descripción:	Devuelve la Plantilla.

<b>Nombre : ElemContenido</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>

Nombre	String
Autor	String
Nivel	String
ReglasNaveg	String
ReglasEval	String
Descripcion	String
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetElemContenido()
Descripción:	Devuelve el Elemento de Contenido

<b>Nombre: ElemContenido</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
Autor	String
Nivel	String
ReglasNaveg	String
ReglasEval	String
Descripcion	String
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetElemContenido()
Descripción:	Devuelve el Elemento de Contenido

<b>Nombre: ElemContenedor</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String

<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetElemContenedor()
<b>Descripción:</b>	Devuelve la plantilla de Elemento Contenedor

<b>Nombre : OA</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
Autor	String
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetOA()
<b>Descripción:</b>	Devuelve la plantilla de Objeto de Aprendizaje

<b>Nombre : CrearElementoTax</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	ButtonCancelar()
<b>Descripción:</b>	Cierra la ventana
<b>Nombre:</b>	ButtonAceptar()
<b>Descripción:</b>	Crea un nuevo elemento en la taxonomía.
<b>Nombre : EditarElementoTax</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
EditNombre	Edit
EditAutor	Edit
EditNivel	Edit

Descripcion	TextEdit
ReglasNaveg	TextEdit
ReglasNaveg	TextEdit
EditFechaInicio	Edit
EditObjetivo	Edit
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	ButtonCancelar()
Descripción:	Cierra la ventana
<b>Nombre:</b>	ButtonAceptar()
Descripción:	Edita el elemento de la taxonomía.

<b>Nombre: Navegacion</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
MemoNavegacion	Memo
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetNavegacion()
Descripción:	Define la forma de navegación por el curso.

<b>Nombre: ImportarElementoTax</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	ButtonCancelar()
Descripción:	Cierra la ventana
<b>Nombre:</b>	ButtonAceptar()

Descripción:	Importa el elemento a la taxonomía.

<b>Nombre : GestionarTaxGuion</b>	
<b>Tipo de clase : Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre :</b>	CrearEC ()
Descripción:	Crea el elemento de contenido en la taxonomía.
<b>Nombre :</b>	EditarEC()
Descripción:	Edita el elemento de contenido.
<b>Nombre :</b>	EliminarEC()
Descripción:	Elimina el elemento de contenido.
<b>Nombre :</b>	ImportarEC()
Descripción:	Importa el elemento de contenido.

<b>Nombre:</b>	DefinirNaveg()
Descripción:	Define la Navegación del estudiante.

<b>Nombre : RegistroTaxGuion</b>	
<b>Tipo de clase : Controladora</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre :</b>	AdicionarElemento()
Descripción:	Adicionar un elemento a la taxonomía.
<b>Nombre :</b>	SeleccionarElemento()
Descripción:	Selecciona un elemento de la taxonomía.
<b>Nombre:</b>	EliminarElemento()
Descripción:	Elimina un elemento de la taxonomía.
<b>Nombre:</b>	ModificarElemento()
Descripción:	Modifica un elemento de la taxonomía.

<b>Nombre : Plantilla</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
<b>Nombre :</b>	GetTipo()
Descripción:	Devuelve el tipo de Plantilla.
<b>Nombre :</b>	GetPlantilla()
Descripción:	Devuelve la Plantilla.

<b>Nombre : ElemContenido</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
Autor	String
Nivel	String
ReglasNaveg	String
ReglasEval	String
Descripcion	String
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetElemContenido()
Descripción:	Devuelve el Elemento de Contenido

<b>Nombre : ElemContenedor</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetElemContenedor()
Descripción:	Devuelve la plantilla de Elemento Contenedor

<b>Nombre : OA</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
Autor	String
<b>Para cada responsabilidad:</b>	

<b>Nombre :</b>	GetOA()
Descripción:	Devuelve la plantilla de Objeto de Aprendizaje

<b>Nombre : Guion</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
Autor	String
Objetivos	String
Palabras Claves	String
<b>Para cada responsabilidad:</b>	
<b>Nombre :</b>	GetNombre()
Descripción:	Devuelve el nombre del guión.
<b>Nombre :</b>	GetAutor()
Descripción:	Devuelve el nombre del autor.
<b>Nombre :</b>	GetObjetivos()
Descripción:	Devuelve los objetivos del curso.
<b>Nombre :</b>	GetPalabrasClav()
Descripción	Devuelve las palabras clave.

<b>Nombre : AdicPlantEstudiante</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
EditNombre	Edit
EditSexo	Edit
EditEdad	Edit



EditCurso	Edit
EditNivel	Edit
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Aceptar()
Descripción:	Adiciona la plantilla estudiante

<b>Nombre: Evaluacion</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
MemoEvaluacion	Memo
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetFormaEval()
Descripción:	Devuelve la forma de evaluación al estudiante.

<b>Nombre: SalidaEstudiante</b>	
<b>Tipo de clase : Interfaz</b>	
<b>Atributo</b>	<b>Tipo</b>
MemoSalida	Memo
RadioBasico	RadioButton
RadioIntermedio	RadioButton
RadioAvanzado	RadioButton
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	Aceptar()
Descripción:	Se le da salida al estudiante.

<b>Nombre: GestModeloPedag</b>	
<b>Tipo de clase : Entidad</b>	

Atributo	Tipo
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	AdicPlantEstudiante()
Descripción:	Adiciona la plantilla del estudiante
<b>Nombre:</b>	DefFormaEvaluacion()
Descripción:	Define la forma de evaluación del estudiante
<b>Nombre:</b>	SalidaEstudiante()
Descripción:	Se le da salida al estudiante.

<b>Nombre: RepositorioGestMP</b>	
<b>Tipo de clase : Entidad</b>	
Atributo	Tipo

<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	AdicionarEst()
Descripción:	Adiciona un estudiante
<b>Nombre:</b>	SeleccionarEst()
Descripción:	Selecciona un estudiante.
<b>Nombre:</b>	EliminarEst()
Descripción:	Elimina un estudiante
<b>Nombre:</b>	ModificarEst()
Descripción	Modifica los datos del estudiante

<b>Nombre: Plantilla</b>	
<b>Tipo de clase : Entidad</b>	
Atributo	Tipo

<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetTipo()
<b>Descripción:</b>	Devuelve el tipo de Plantilla.
<b>Nombre:</b>	GetPlantilla()
<b>Descripción:</b>	Devuelve la Plantilla.

<b>Nombre: Estudiante</b>	
<b>Tipo de clase : Entidad</b>	
<b>Atributo</b>	<b>Tipo</b>
Nombre	String
Edad	Int
Sexo	char
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	GetEvaluacion()
<b>Descripción:</b>	Devuelve la evaluación del estudiante

### 3.6.2 Diagramas de clases del Diseño

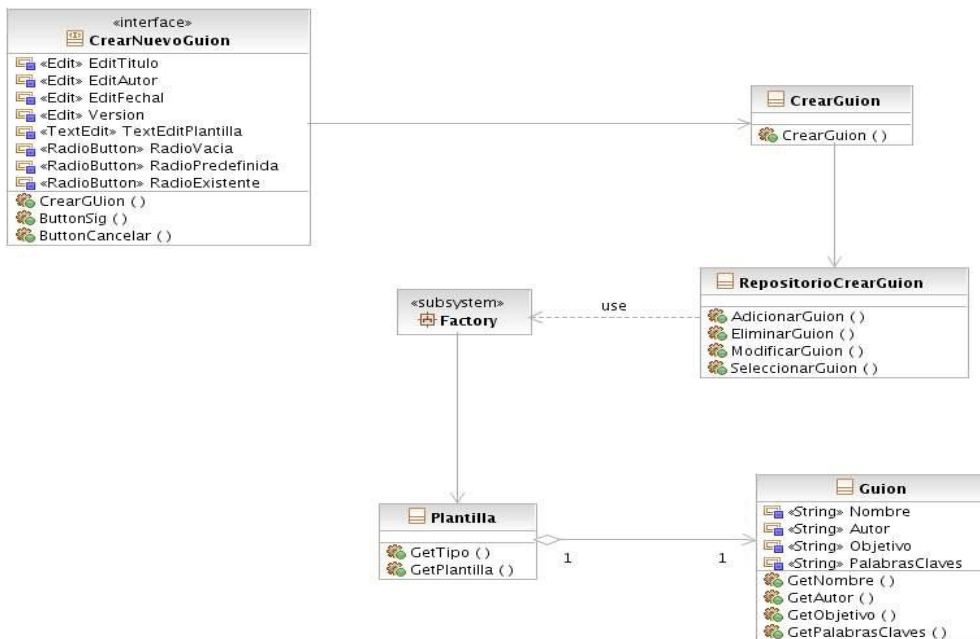


Fig 16. Crear Nuevo Guión

Diagrama de Secuencia “Crear Guión” (Ver Anexo 8)

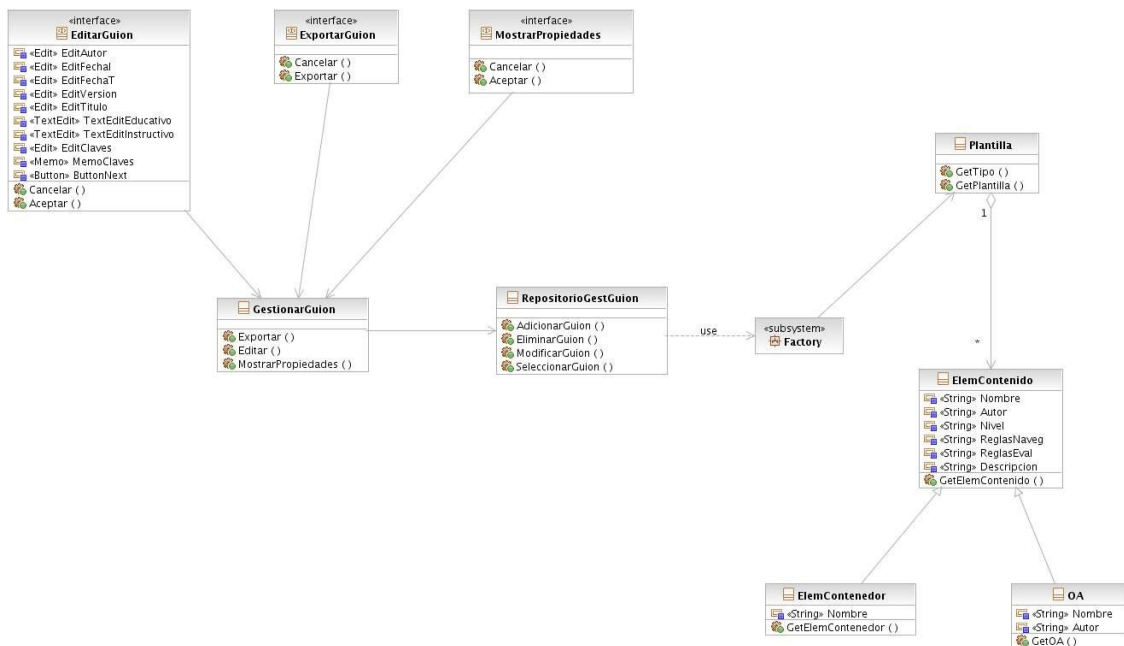


Fig 17 Gestionar Guión de Contenido

Diagrama de Secuencia “Escenario Mostrar Propiedades” (Ver Anexo 9.1)

Diagrama de Secuencia “Escenario Exportar Guión” (Ver Anexo 9.2)

Diagrama de Secuencia “Escenario Editar Guión” (Ver Anexo 9.3)

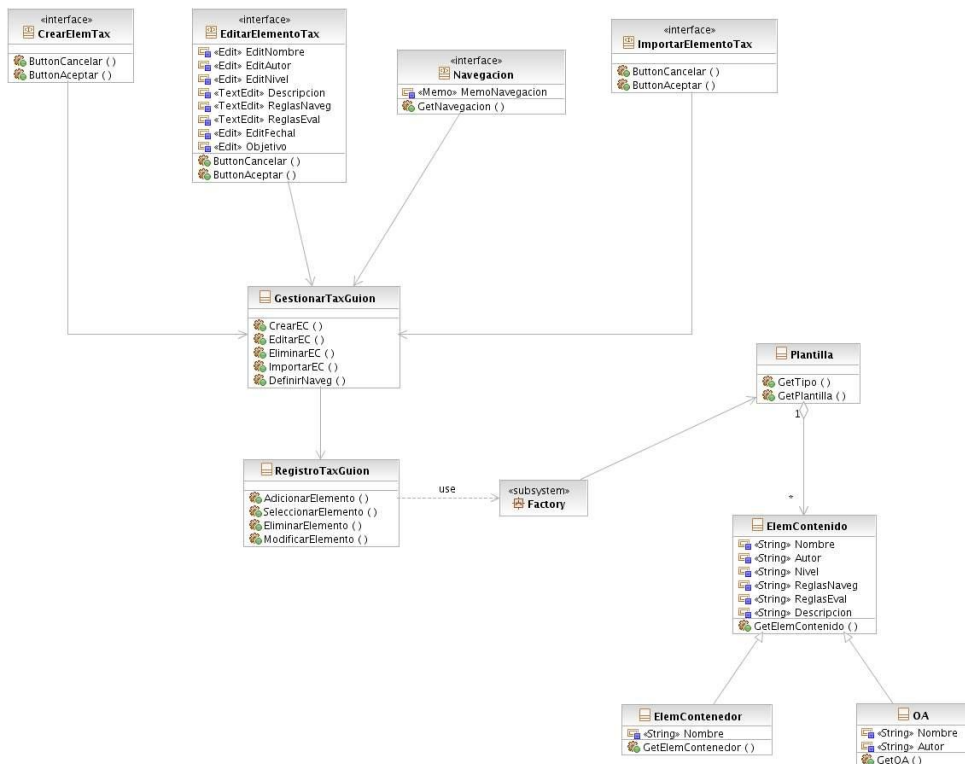


Fig 18. Gestionar Taxonomía del Guión

Diagrama de Secuencia “Escenario Crear Elemento” (Ver Anexo 10.1)

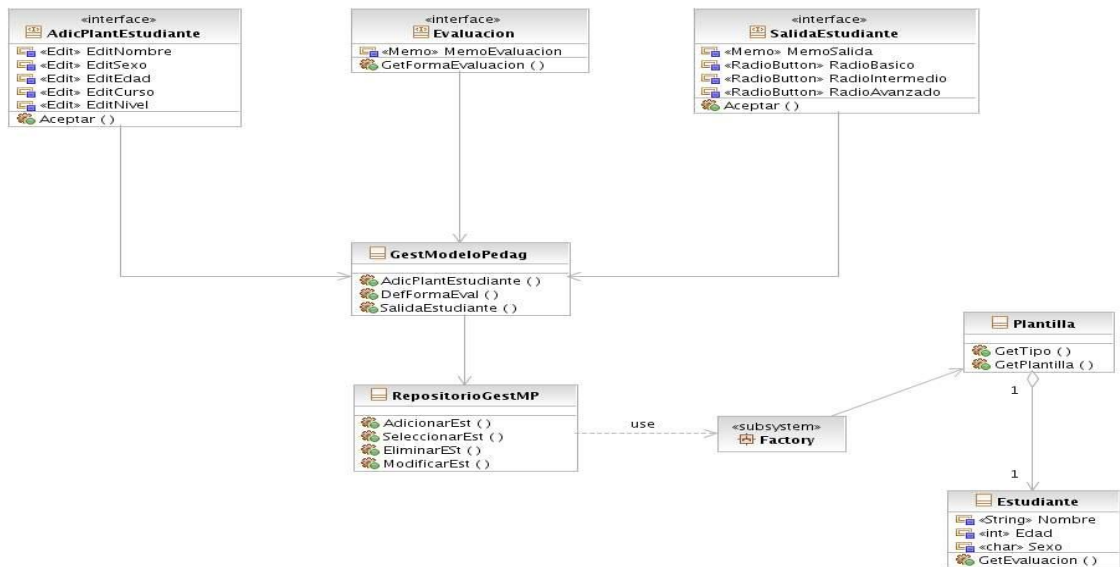


Fig 19. Modelo pedagógico

Diagrama de Secuencia “Escenario” (Ver Anexo 11.1)

Diagrama de Secuencia “Escenario” (Ver Anexo 11.2)

### Conclusiones Generales

Al finalizar el presente trabajo, se consideran cumplidos el objetivo trazado y suficientemente fundamentada la idea a defender. A este planteamiento se puede arribar a partir de las siguientes conclusiones:

Se cumplió correctamente el objetivo general así como las tareas planteadas para la investigación.

Se ofrecieron los elementos teóricos que sirven de sustento científico a la investigación, así como se ha ofrecido cada uno de los aspectos a tener en cuenta en la situación problemática que genera el problema científico.

Con el diseño de la Herramienta de autor se deja plasmado la base para la futura implementación de la herramienta, con el objetivo de integrarla al sistema principal.

Con el diseño de la interfaz de comunicación con el usuario se logró una mejor comprensión de los casos de uso, mejorando la relación entre los desarrolladores y los usuarios.

Con la modelación del dominio se logró de manera visual mostrar los principales conceptos que se manejan en el entorno del sistema en desarrollo.

Con la utilización del patrón Modelo-Vista-Controlador se logró una mejor identificación de las clases brindando una solución simple y clara.

La presente herramienta de autor facilitará el trabajo por parte de los profesores e instructores, en la creación de los guiones de contenido que viabilizarán la producción de los cursos a un esfuerzo y costo mínimo.

El uso de las herramientas de este tipo para generar materiales destinados a enseñar, supone en sí, una forma inequívoca de integración curricular de las tecnologías de la información y la comunicación, que debe ser alentada e impulsada por maestros y autoridades educativas de forma clara y decidida.

### **Recomendaciones**

Enriquecer los elementos conceptuales de la nueva solución propuesta, sobre la base de los conceptos que se traen a colación en la investigación, para brindarle una mayor solidez en el tratamiento de la información.

Valorar el diseño de los estereotipos decorativos, sobre la base de una mejor representación visual de los elementos de la presente aplicación, para aumentar el entendimiento por parte de los especialistas pedagógicos y de diseño; logrando una mayor interacción con el usuario final.

A partir del planteamiento anterior se suscita la siguiente recomendación: Mejorar la utilización de aplicaciones de este tipo, haciéndolas entendibles y claras; logrando eliminar así la brecha de comunicación entre los especialistas en contenido y los desarrolladores.

Buscar soluciones que respondan a prescindir totalmente de las leyes de propiedad para la utilización de herramientas de este tipo; que impiden un adecuado desarrollo de los procesos educativos.

Adaptar la presente herramienta al modo de programación de la herramienta principal, posibilitando una sola línea de arquitectura entre el sistema general y los componentes que la integran.

## Referencias Bibliográficas

- Albert. 2004.** Macromedia. [Online] 2004.  
[http://www.macromedia.com/software/authorware/..](http://www.macromedia.com/software/authorware/)
- ANACHE, I. and M. JOEL. 2004.** *Marcando un hito en el desarrollo de software.* HABANA : 2004.
- 2004.** Authoring Tool Strategies. [Online] 2004.  
[http://www.brandonhall.com/public/execsums/authoring\\_tool\\_strategies.pdf](http://www.brandonhall.com/public/execsums/authoring_tool_strategies.pdf).
- B. De Benito Crosetti, and J. Salinas Ibáñez, En Gómez, J.I.A. and Almenara, J.C. 2002.** "Webtools: Aplicaciones para sistemas virtuales de formación". España. , Málaga : 2002.
- B. De Benito Crosetti, J. Salinas Ibáñez, En Gómez, J.I.A. and Almenara, J.C., Editores. Educar en Red. 2002.** "Webtools: Aplicaciones para sistemas virtuales de formación". Málaga, España : 2002.
- Bangs, P. 2004.** *Introduction to CALL authoring programs.* [Online] 2004.  
[http://www.ict4lt.org/en/en\\_mod2-5.htm](http://www.ict4lt.org/en/en_mod2-5.htm).
- Barrera Yanes, Rafael. 1998.** *Del objetivo al guión interactivo.* . La Habana : 1998.
- Bell, B. 1998.** "Investigate and decide learnings environments: especializing task models for authoring tool design". 1998. pp. 65-105.
- Ceja, Luis Mena. 2000.** "Nuevos ambientes de aprendizaje en el desarrollo del alumno En la UPIICSA". [www.somece.org.mx](http://www.somece.org.mx). [Online] 2000.
- Collazo, Yordanys Piñeiro Gómez Jacqueline Gallardo. 2007.** *PRINCIPIOS ESTRATÉGICOS PARA LA GUÍA DEL PROCESO DE DESARROLLO DE SOFTWARE EDUCATIVO EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS.* UCI, Ciudad de la Habana : 2007.
- Dabagh, Nadda. 2001.** *Authoring tools an Learning Systems: A HIstorical Perspective.* 2001.
- . 2001. *Authoring tools an Learning Systems: A HIstorical Perspective.*
- DEBIAN, C. 2007.** *Acerca de Debian.*
- García, Alejandro Pérez. 2007.** *BOUML, una herramienta CASE de UML gratuita.*
- Hall, B. 2004.** *Authoring Tool Strategies.* [Online] 2004.  
[http://www.brandonhall.com/public/execsums/authoring\\_tool\\_strategies.pdf](http://www.brandonhall.com/public/execsums/authoring_tool_strategies.pdf).
- Hennicker, Rolf y Koch, Nora. 2000.** [Online] 2000. <http://www.pst.informatik.uni-muenchen.de/personen/kochn/Uml2000.pdf>.
- Herramientas. 2000.** Herramientas.  
([http://pulsar.ehu.es/pulsar/buenaspracticaherramientas/az\\_herramientasconte?set\\_language=eu&cl=eu,2004](http://pulsar.ehu.es/pulsar/buenaspracticaherramientas/az_herramientasconte?set_language=eu&cl=eu,2004)). [Online] 2000.
- IEEE. 2001.** Learning Technology Standards Committee. <http://ltsc.ieee.org/wg12/>. [Online] 2001.
- Isasa. 2001.** *Retos de la nueva sociedad de la información.* Aula Magna de la Universidad de San Pablo., s.l. : 2001. Aula Magna de la Universidad de San Pablo..
- James Rumbaugh, Ivar Jacobson, Grady Booch. 2000.** *El lenguaje unificado de modelado. Manual de Referencia.*
- Jimmy. 2001.** Learning Technology Standards Committee. <http://ltsc.ieee.org/wg12/>. [Online] 2001. Learning Technology Standards Committee..
- Larman, C. UML y Patrones. 2000.** 2000.



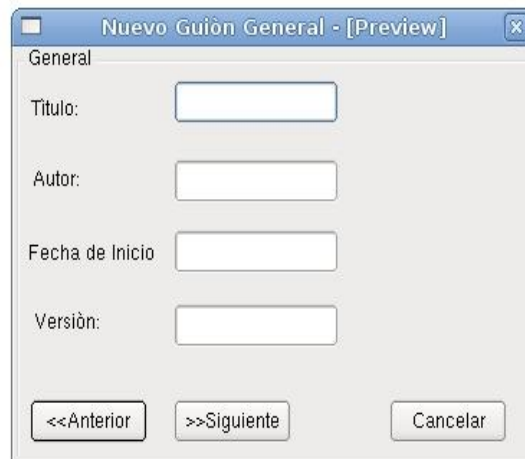
**Bibliografía**

- Leeuwe, M. De. 2004.** *An Introduction to Authoring Tools*. [Online] 2004.  
<http://www.e-learning-site.com/index.html>.
- López, Ramirez and. 2004.** *Propuesta para un sistema de Catalogación y Recuperación de Recursos de Información*. Instituto Superior Politécnico “José Antonio Echeverría”. Ciudad de la Habana, : 2004.
- M.Sc Febe Angel Ciudad Ricardo, Ing. Yosnel Herrera Martínez. 2007.** *DoMet*. Universidad de Ciencias Informáticas, Ciudad Habana : 2007.
- . **2007.** *DoMet*. Universidad de Ciencias Informáticas, Ciudad Habana : 2007.
- MSc. Jose L. MOntero, OFarrill y Dra Elsa Herrero Tunis. 2005.** *"Las Herramientas de Autor en el proceso de producción de materiales educativos en formato digital"*. Instituto Superior Minero Metalúrgico de moa, s.l. : 2005.
- Piattini, P. 1996.** *"Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión."*.
- Pulichino., Joe. 2005 .** *The Content Authoring Research Report*. s.l. :  
<http://etidweb.tamu.edu/listserv.php>, 2005 .
- R. Barchino Plata, J. M. Gutiérrez and S. Otón., 2004.** *Evaluación y Descripción de Contenidos Educativos Reutilizables* . Guadalajara : 2004.
- Ricardo, F. 2008.** [Online] mayo 30, 2008.  
<http://www.monografías.com/trabajos43/patron-modelo-vista..>
- Robertson, J. 2003.** *"What is a content management system?"*.
- Rodríguez, Liván A. 2007.** *Evaluación de la Herramienta de Autor "revolution" para el desarrollo de hiperentornos interactivos de aprendizaje multiplataformas*. Universidad Pedagógica "Felix Varela", Santa Clara : 2007.
- RUMBAUGH and JACOBSON, I. 2000..** *El proceso unificado de desarrollo de software*. La Habana : s.n., 2000.
- RUMBAUGH, J and JACOBSON, I. 2000.** *El proceso unificado de desarrollo de software*. . La Habana : 2000.
- T. Murray, S. Blessing and S. 2003.** *Interactive and Intelligent educational software*. 2003. p. 508.
- T. Murray, S. Blessing and S. Ains worth. 2003.** *Interactive and intelligent educational software*. 2003. p. 341.
- Tunis., MSc. Jose L. MOntero O`Farrill y Dra Elsa Herrero. 2005.** *"Las Herramientas de Autor en el proceso de producción de materiales educativos en formato digital"*. Instituto Superior Minero Metalúrgico de moa., Holguín : 2005.
- Wiley. 2000.** *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy"*.
- Zazpe, Pedro Razquin. 1998.** *Los Sistemas de Autor Multimedia*. Escuela Universitaria de Biblioteconomía y Documentación (EUBD) Universidad Complutense de Madrid. , s.l. : 1998.  
<http://autore2005.ehu.es>.  
<http://www.easyprof.com/home/home.jsp>.

## Anexos

### 1. Crear un Nuevo Guión de Contenido.

#### 1.1 Atributos Generales.

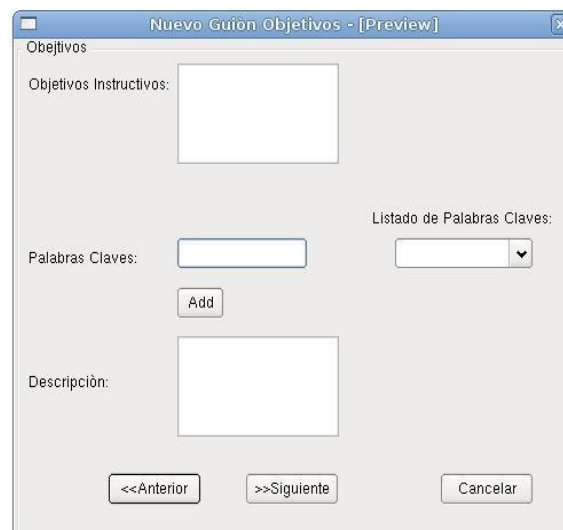


The image shows a dialog box titled "Nuevo Guión General - [Preview]". It has a "General" tab. The fields are:

- Titulo: [Text Input]
- Autor: [Text Input]
- Fecha de Inicio: [Text Input]
- Versión: [Text Input]

At the bottom, there are three buttons: "<<Anterior", ">>Siguiete", and "Cancelar".

#### 1.2 Objetivos

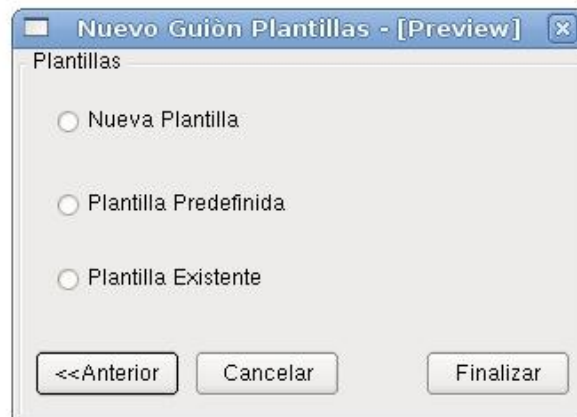


The image shows a dialog box titled "Nuevo Guión Objetivos - [Preview]". It has an "Objetivos" tab. The fields are:

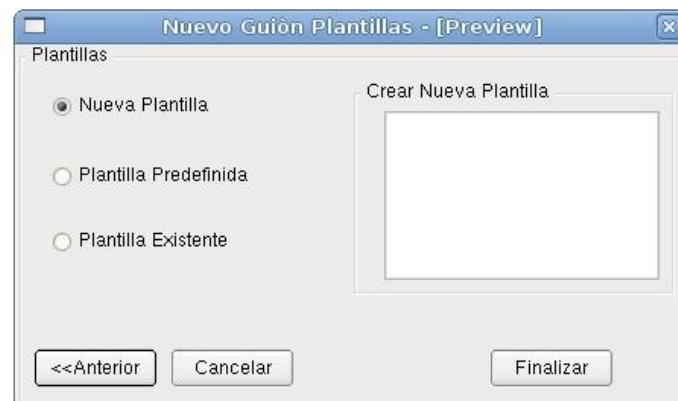
- Objetivos Instructivos: [Large Text Area]
- Palabras Claves: [Text Input] with an "Add" button below it.
- Listado de Palabras Claves: [Dropdown Menu]
- Descripción: [Large Text Area]

At the bottom, there are three buttons: "<<Anterior", ">>Siguiete", and "Cancelar".

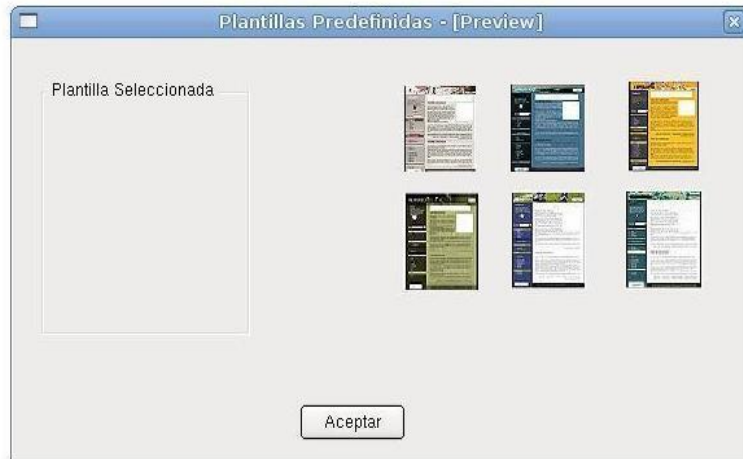
#### 1.3 Plantillas



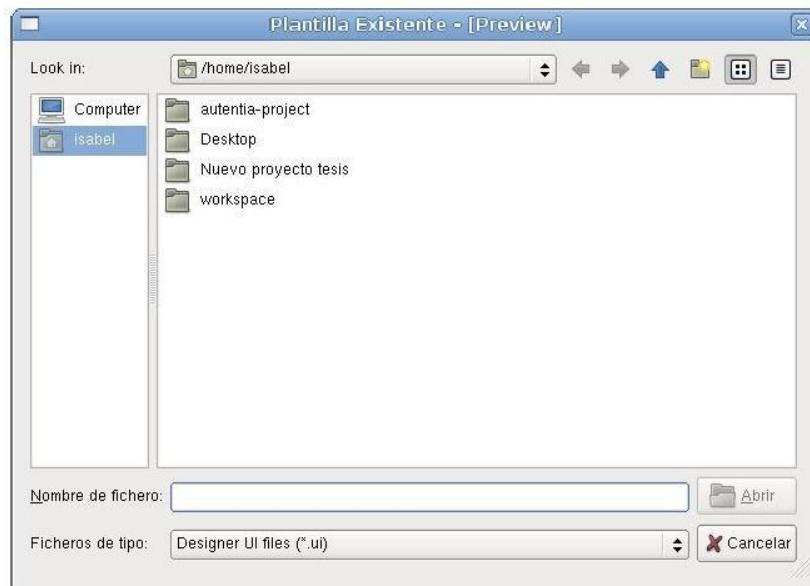
### 1.3.1 Nueva Plantilla



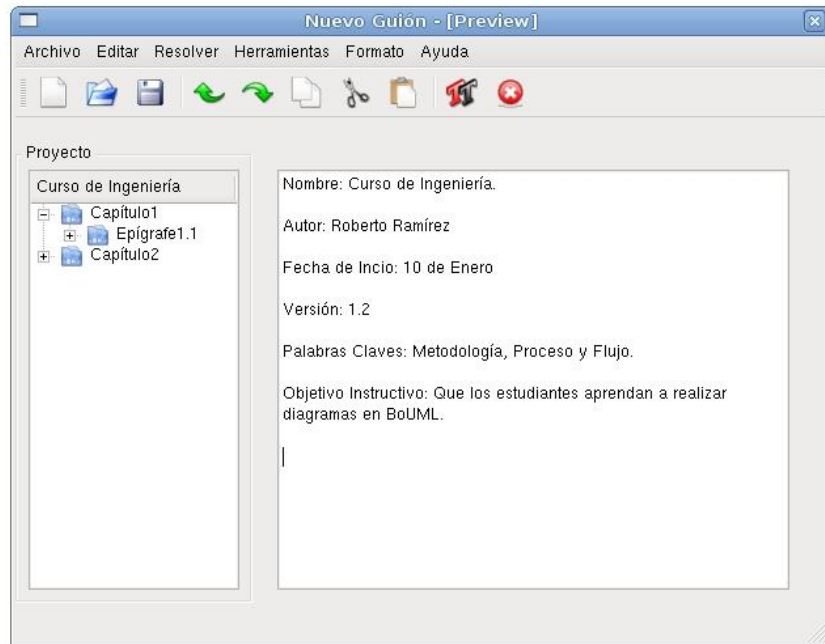
### 1.3.2 Plantilla Predefinida.



### 1.3.3 Plantilla Existente.

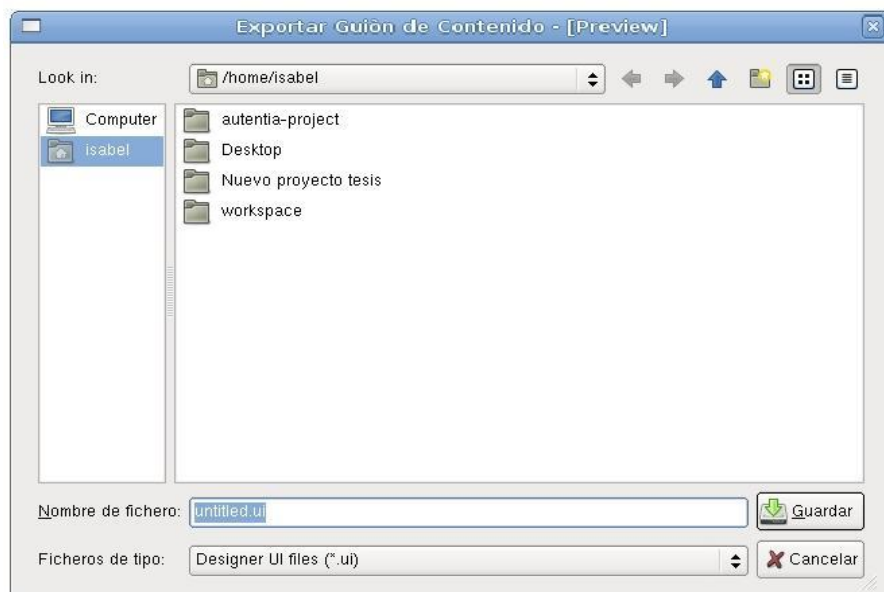


### 1.4 Área de trabajo (Nuevo Guión).



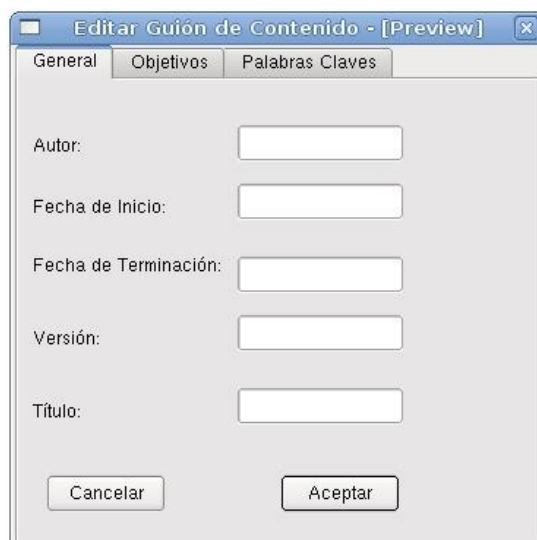
## 2. Gestionar Guión de Contenido

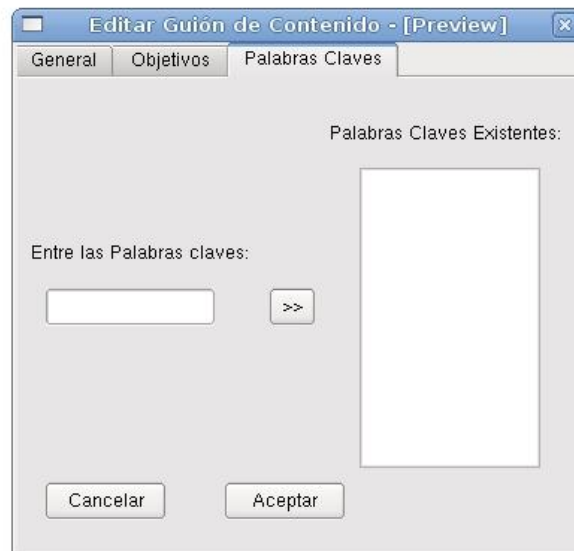
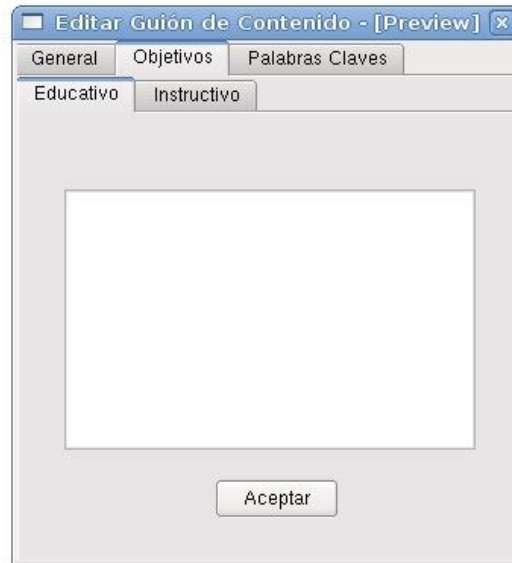
### 2.1 Exportar Guión.



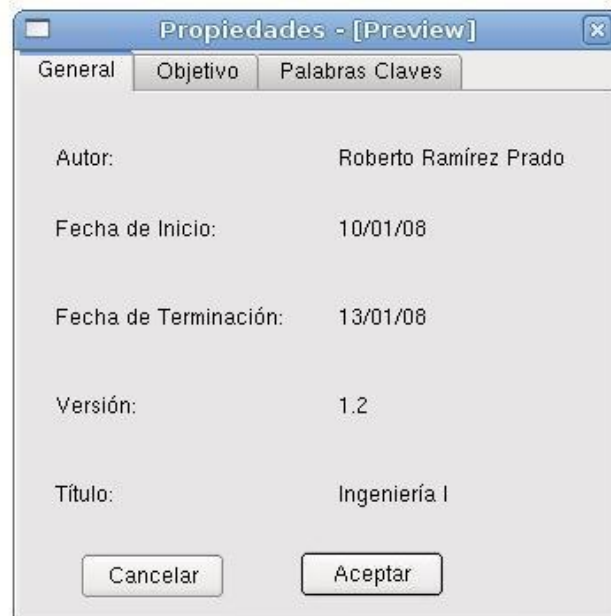


## 2.2 Editar Guión



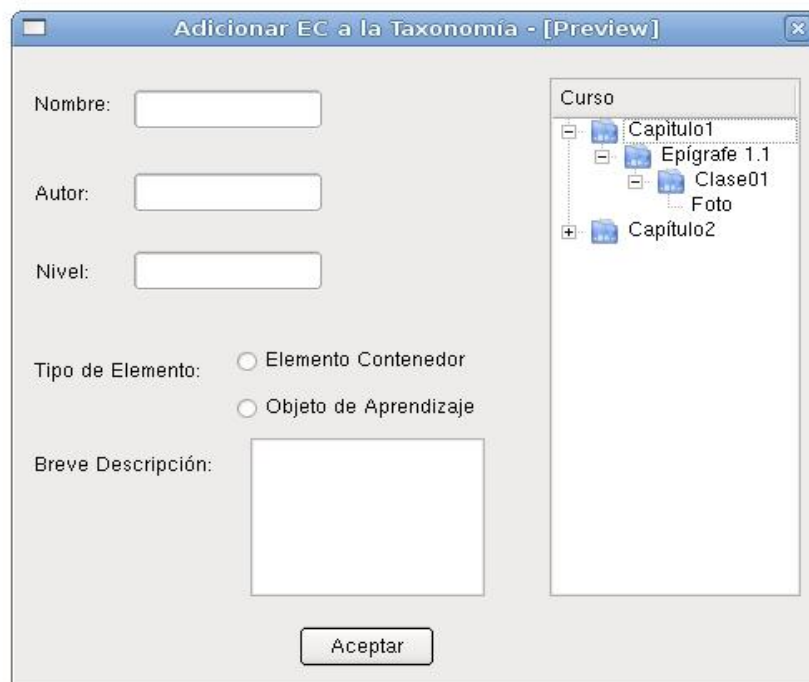


### 2.3 Mostrar Propiedades



### 3. Gestionar Taxonomía del Guión de Contenido

#### 3.1 Adicionar EC a la taxonomía.



#### 3.2 Editar Elemento de Contenido.



Editar Elemento de Contenido - [Preview]

Generales Atributo Valor

Nombre:

Autor:

Nivel:

Breve Descripción:

Cancelar Aplicar Finalizar

Editar Elemento de Contenido - [Preview]

Generales Atributo Valor

Fecha de inicio:

Objetivos

Cancelar Aplicar Finalizar

Editar Elemento de Contenido - [Preview]

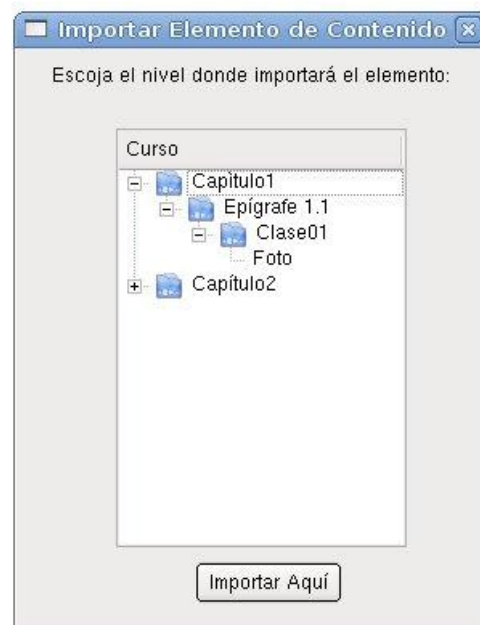
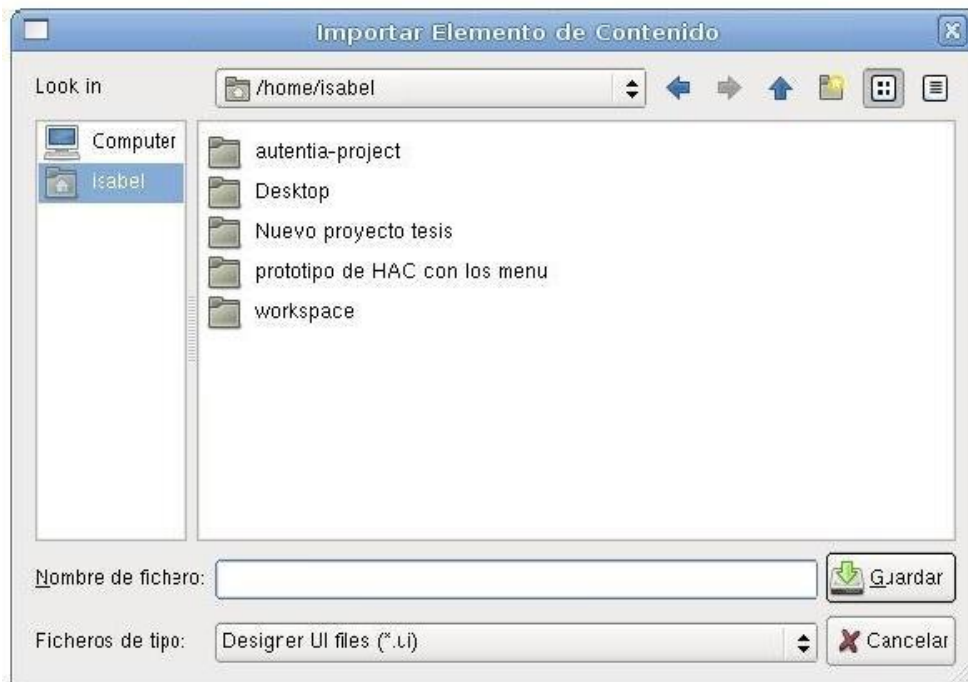
Generales Atributo Valor

Reglas de Navegación REglas de Evaluación

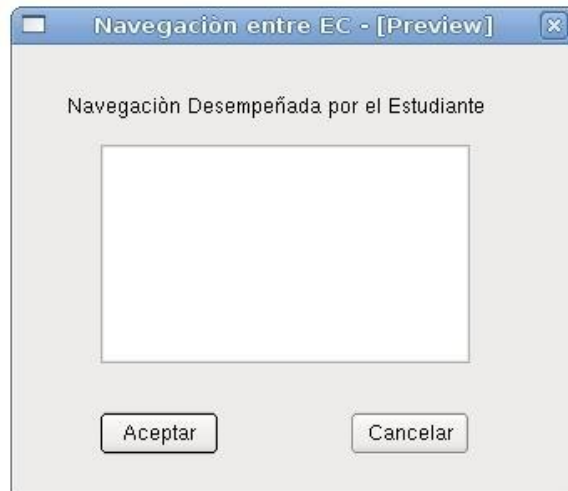
Introducir Reglas de Navegación:

Cancelar Aplicar Finalizar

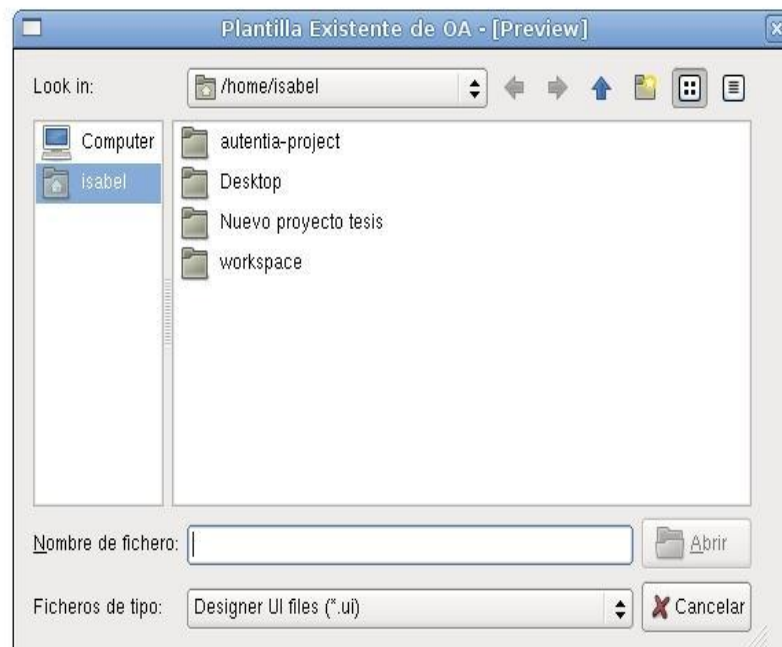
### 3.3 Importar un EC a la taxonomía.

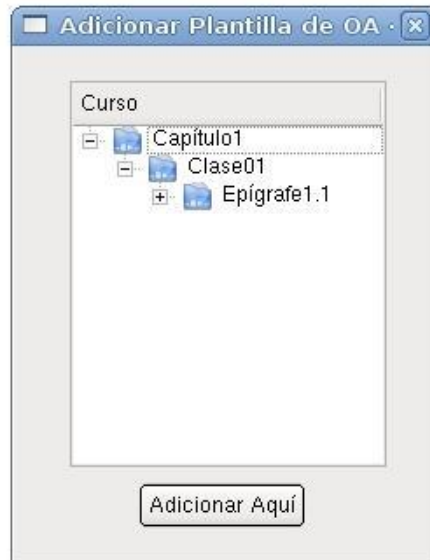


### 3.5 Comprobar Navegación.



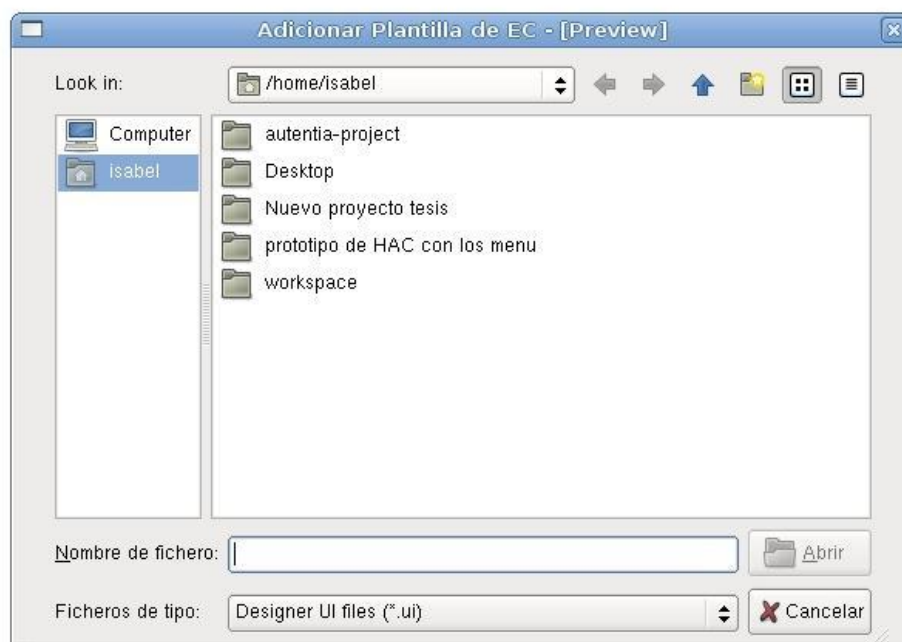
#### 4. Gestionar Plantilla de OA

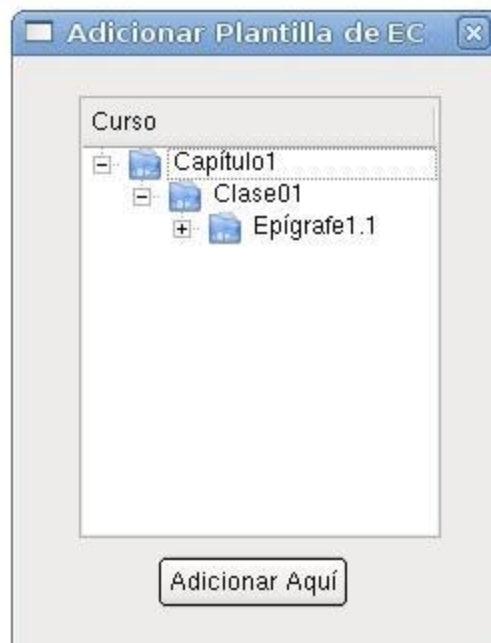




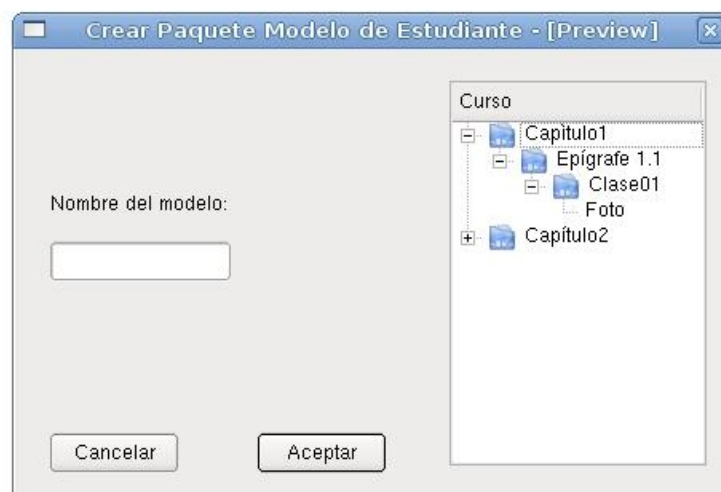
## 5. Gestionar Plantilla de Elemento Contenedor.

### 5.1 Adicionar Plantilla de EC.





## 6. Crear Paquete Modelo de Estudiante



## 7. Gestionar Modelo pedagógico

Editar Plantilla Estudiante - [Pr x]

Seleccione Plantilla de Estudiante a Editar:

Aceptar

Adicionar Plantilla del Estudiante x

Nombre:

Sexo:

Edad:

Curso:

Nivel:

Aceptar

Salida del Estudiante - [Preview] x

Trayectoria del Estudiante:

Definir Categoría:

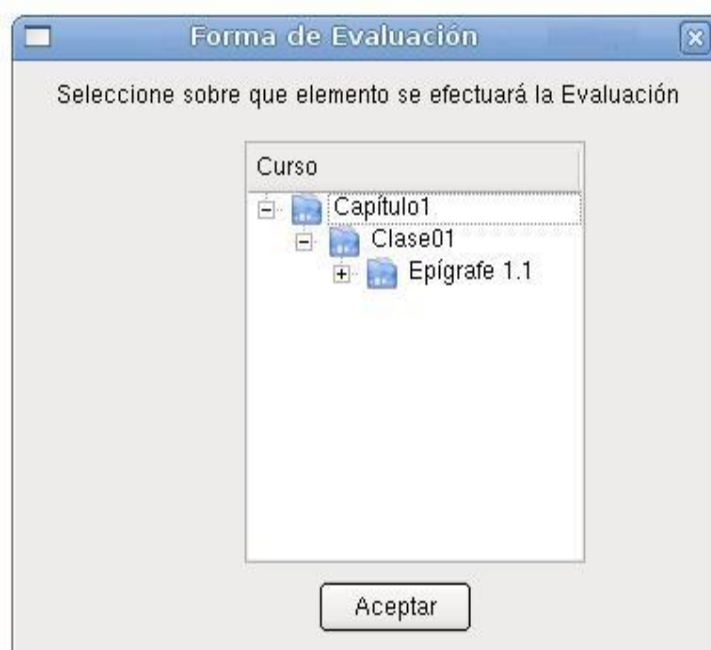
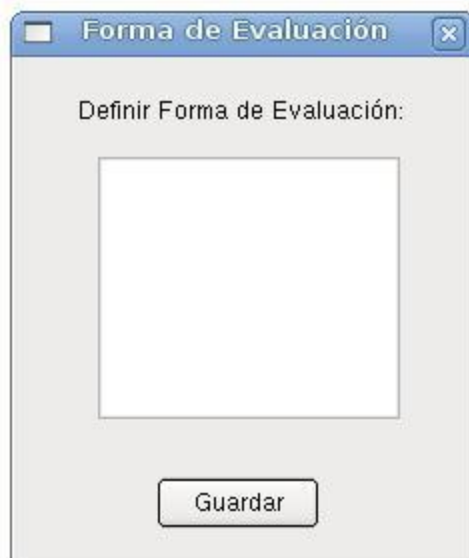
Básico

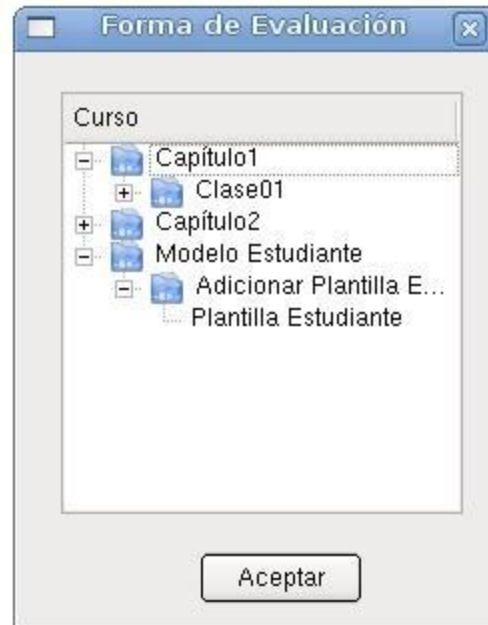
Intermedio

Avanzado

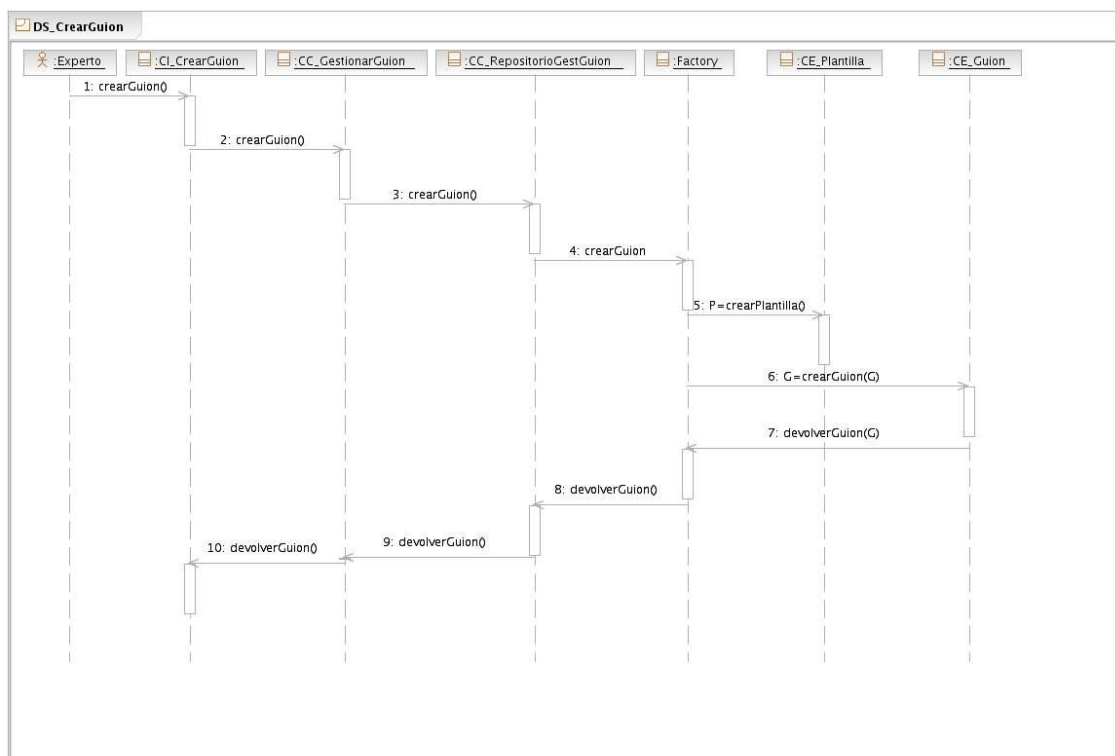
OK

### 7.1 Forma de Evaluación





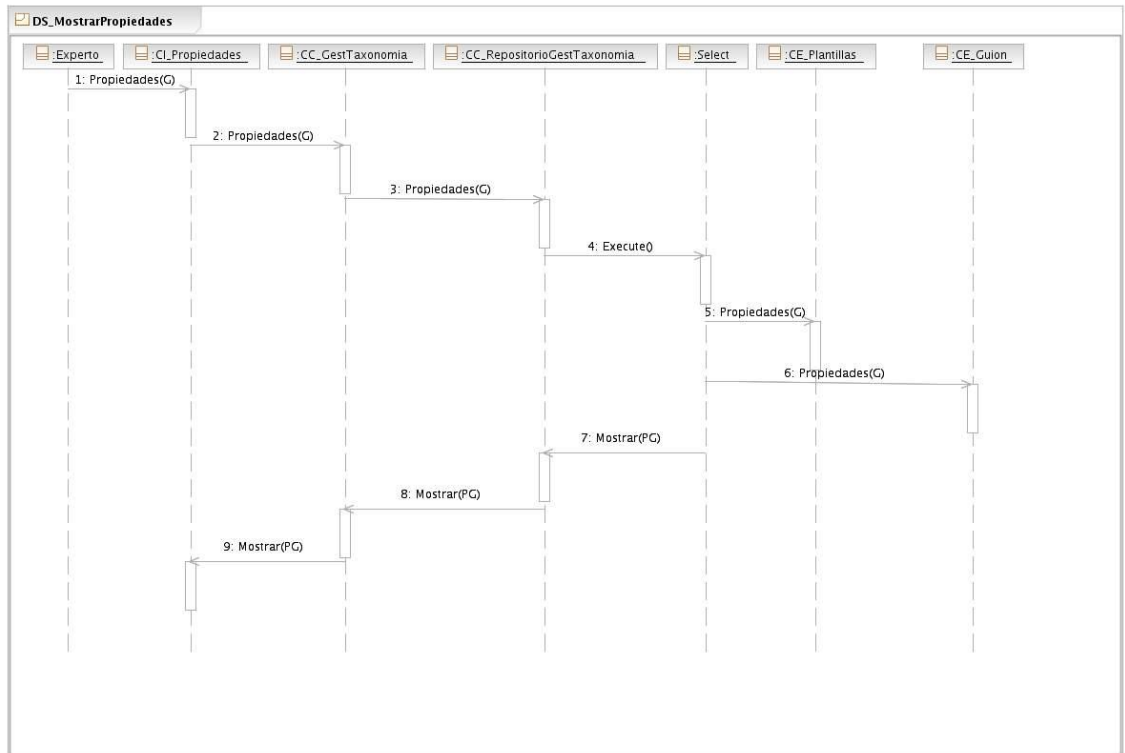
## 8. Diagrama de Secuencia CU Crear Nuevo Guión.



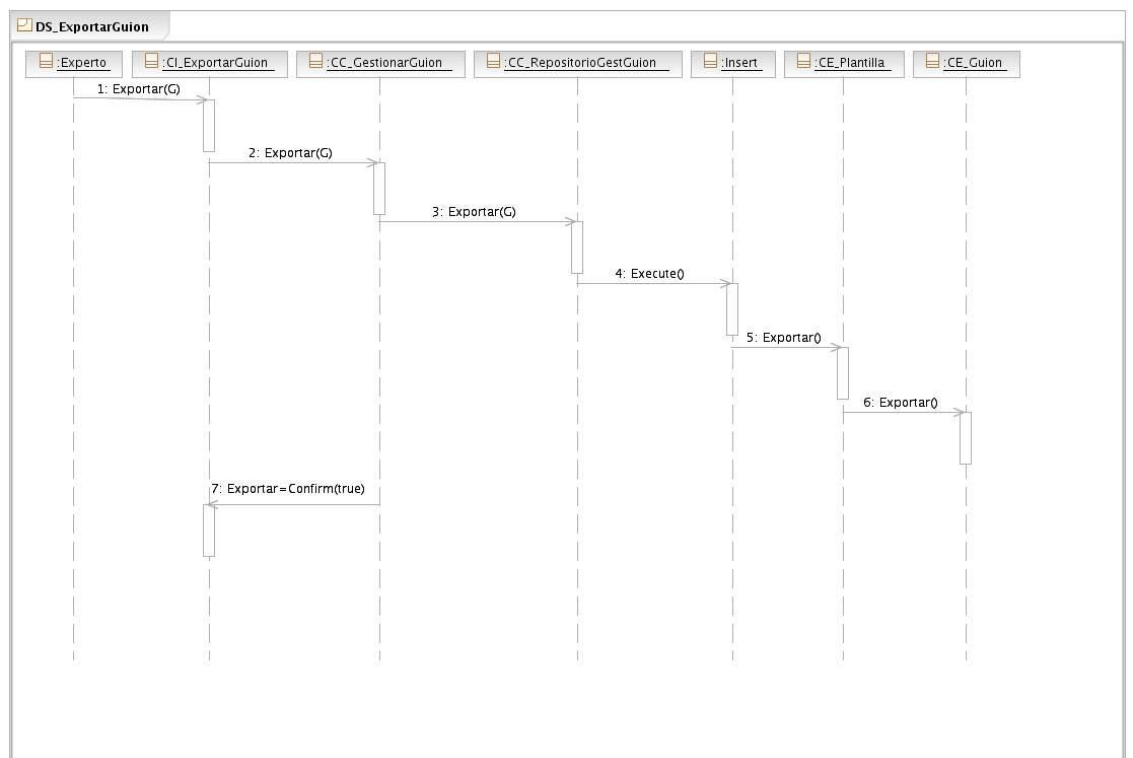
## 9. Diagramas de Secuencia CU Gestionar Guión de Contenido

### 9.1 Escenario Mostrar Propiedades

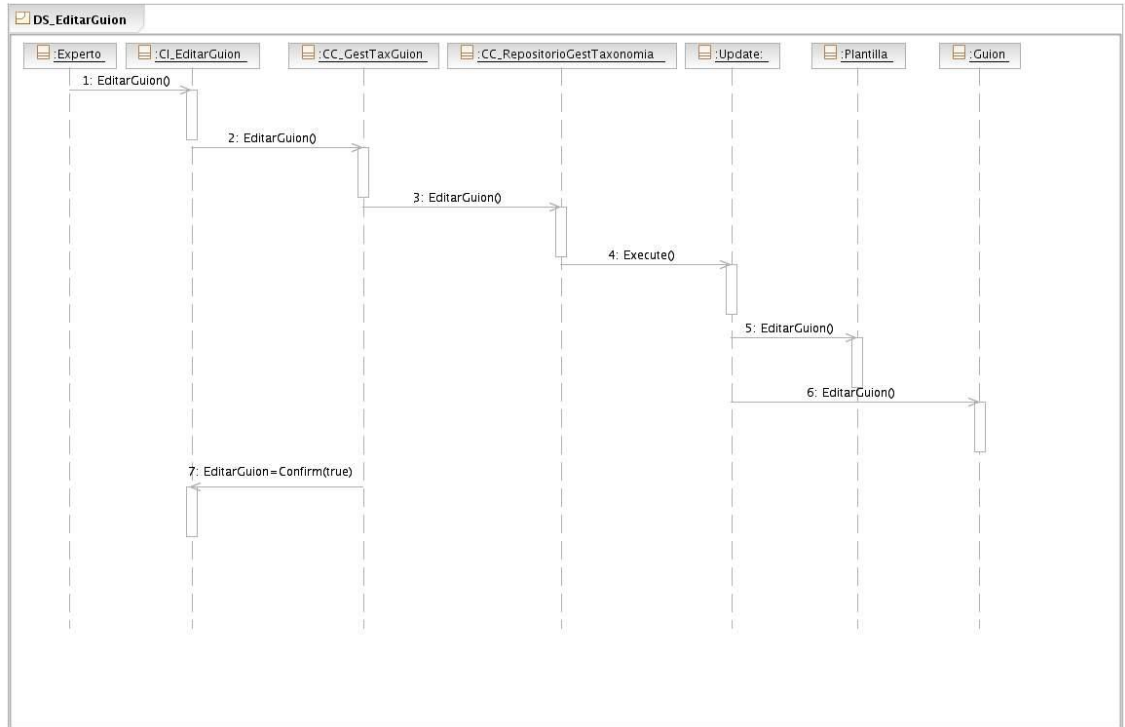




### 9.2 Escenario Exportar Guión

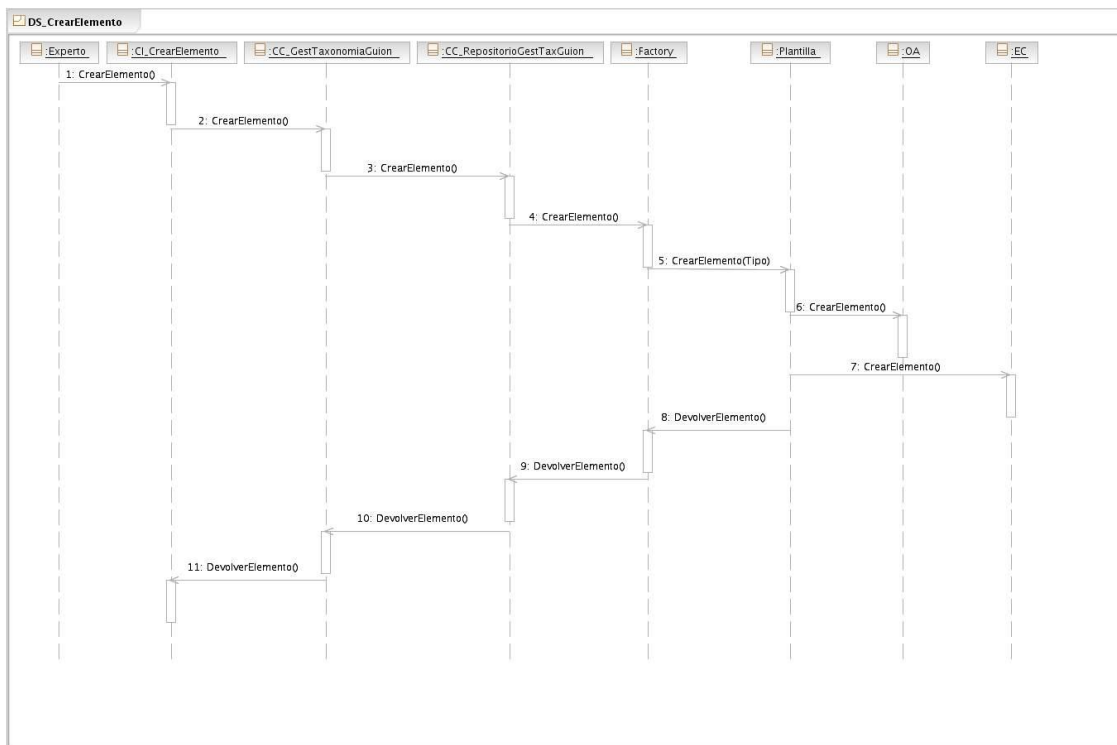


### 9.3 Escenario Editar Guión



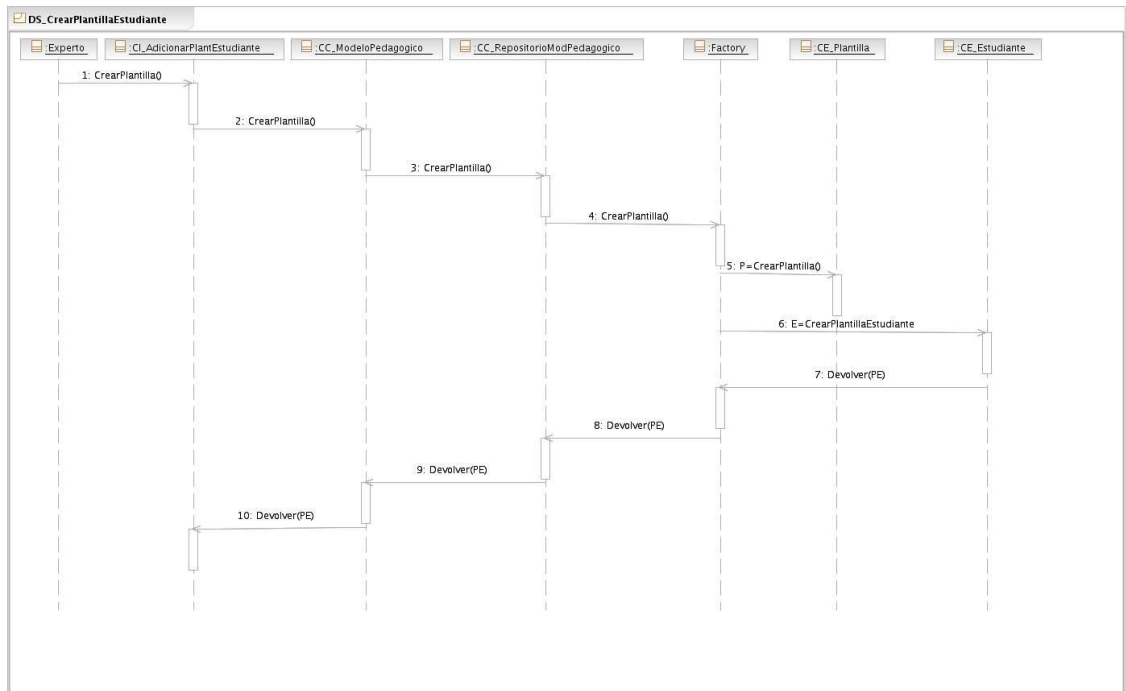
## 10 Diagramas de Secuencia CU Gestionar Taxonomía del Guión

### 10.1 Escenario Crear Elemento

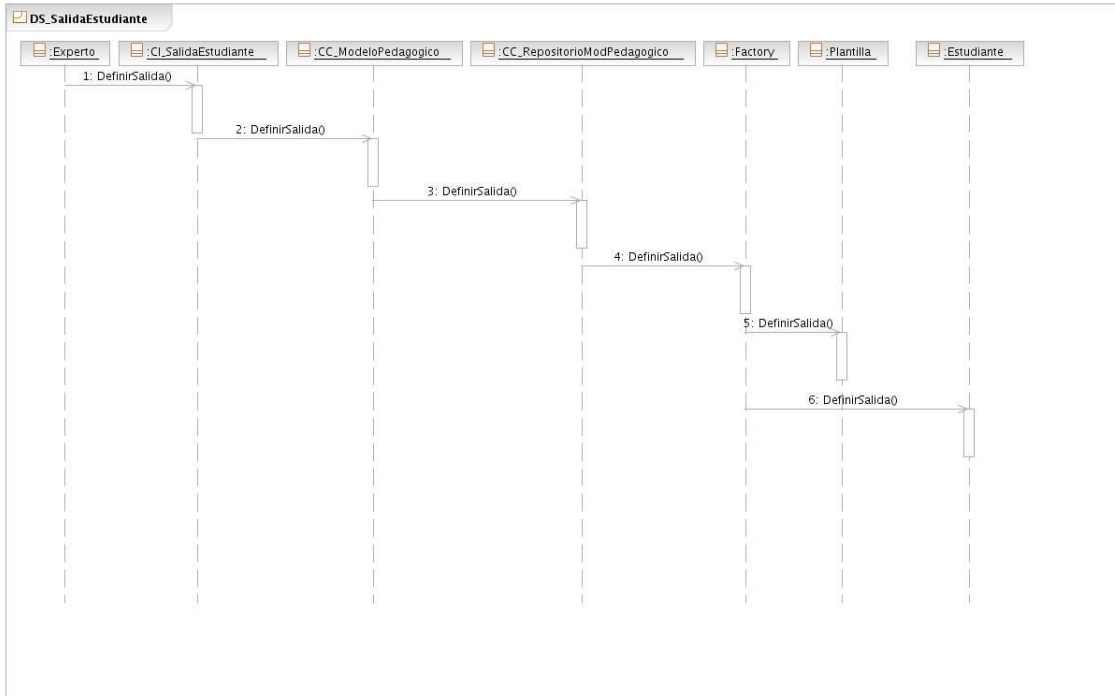


## 11 Diagramas de Secuencia CU Modelo pedagógico

### 11.1 Diagrama de Secuencia “Escenario Crear Plantilla Estudiante”



### 12.2 Diagrama de Secuencia “Escenario Definir Salida Estudiante”



## Glosario

### **E-learning:**

- El e-learning es un término que procede del inglés. Se puede definir como el uso de las tecnologías multimedia para desarrollar y mejorar nuevas estrategias de aprendizaje. En concreto, supone la utilización de herramientas informáticas, tales como CD-ROMs, internet o dispositivos móviles para llevar a cabo una labor docente. La acepción más común para e-learning es la enseñanza a través de internet. En español se utiliza el término teleformación.
- Es un nuevo concepto de educación a distancia en el que se integra el uso de las TIC y otros elementos didácticos para la capacitación y enseñanza. El e-learning utiliza herramientas y medios diversos como Internet, intranets, CD-ROM, presentaciones multimedia, etc. Los contenidos y las herramientas pedagógicas utilizadas varían de acuerdo con los requerimientos específicos de cada individuo y de cada organización.
- Es el conjunto de actividades necesarias para la creación y uso de un entorno de formación a distancia online mediante el uso de tecnologías de la información y comunicaciones.
- Es el desarrollo del proceso de formación, a través del uso de las tecnologías de Internet, facilitando los accesos, promoviendo la tecnología y abaratando los costos, el e-learning debe formar parte integral de los procesos estratégicos de cualquier organización que pretenda liderar en este apoca de alta competitividad.

**Objeto de aprendizaje:** Unidad reusable de información independiente de los medios.

Bloque modular de contenido de teleformación.

- Cumple con las propiedades de un elemento de contenido atómico, tiene un objetivo instructivo y debe estar compuesto por al menos un elemento de información.
- Permiten una organización particular de los contenidos de los cursos.

- Son pequeños componentes educativos digitales y reutilizables de contenidos que abordan un objetivo pedagógico específico.
- Atributos:

Es un Elemento de Contenido, no obstante por las funciones específicas que están asociadas a este como el último en la taxonomía, para la distribución de los contenidos y la interacción de los estudiantes con ellos, posee un conjunto de atributos adicionales.

Tipo de tarea: define que grupo de funcionalidades que posee el objeto de aprendizaje, las tareas están tipificadas de acuerdo a los modelos que se encuentren implementados en ese momento.

- Componentes del OA.
  - 1.1 Textos.
  - 1.2 Imágenes
  - 1.3 Animaciones.
  - 1.4 Applets.
  - 1.5 Otros.
- Interactividad de Tipo
  - 2.1 Pasiva.
  - 2.2 Activa.
  - 2.3 Mixta.
  - 2.4. Otra definida por el Especialista.
- 3. Interactividad de Nivel.
  - 3.1 Alta.
  - 3.2 Media.
  - 3.4 Baja.
  - 3.5 Nula.

- Operaciones:

**Facets:** Proporciona los elementos de información que componen al objeto de aprendizaje.

Instrucción Asistida por Computadora: este enfoque pretende facilitar la tarea de educador, sustituyéndole parcialmente en su labor. El software educacional

resultante generalmente presenta una secuencia (a veces establecida con técnicas de inteligencia artificial) de lecciones, o módulos de aprendizaje. También generalmente incluye métodos de evaluación automática, utilizando preguntas cerradas. Las críticas más comunes contra este tipo de software son:

1. Los aprendices pierden el interés rápidamente, intentan adivinar la respuesta al azar.
2. La computadora es convertida en una simple máquina de memorización costosa.
3. El software desvaloriza, a los ojos del aprendiz, el conocimiento que desea transmitir mediante la inclusión de artificiales premios visuales.

**Software educacional o educativo:** es software cuya finalidad primordial es, dependiendo el punto de vista: la enseñanza, o el aprendizaje. Abarca desde programas para el nivel preescolar, que tienen un gran componente lúdico, a programas de mecanografía y aplicaciones para aprender lenguas extranjeras.

Así como existen profundas diferencias entre las filosofías pedagógicas, así también existe una amplia gama de enfoques para la creación de software educativa atendiendo a los diferentes tipos de interacción de debería existir entre los actores del proceso de enseñanza aprendizaje: educador, aprendiz, conocimiento, computador.

**Software:** programática o equipamiento lógico es el conjunto de programas que puede ejecutar el hardware para la realización de las tareas de computación a las que se destina. Se trata del conjunto de instrucciones que permite la utilización del ordenador o computador (pc, personal computer). El software es la parte intangible de la computadora, es decir programas, aplicaciones etc

**Software Educativo Abierto:** este (no confundir con *software de código abierto* o *software de fuente abierta*) por el contrario enfatiza más el aprendizaje creativo que la enseñanza. El software resultante no presenta una secuencia de contenidos a ser aprendida, sino un ambiente de exploración y construcción virtual, también conocido como micromundo. Con ellos los aprendices, luego de familiarizarse con el software, pueden modificarlo y aumentarlo según su interés personal, o crear proyectos nuevos teniendo como base las reglas del micromundo. Las críticas más comunes contra este tipo de software son:

- En un ambiente donde se use software educacional abierto, no todos los

aprendices aprenderán la misma cosa, y por consiguiente los métodos de evaluación tradicionales son poco adecuados.

- La dirección de tales ambientes de aprendizaje requiere mayor habilidad por parte del educador.

**Multimedia:** es un sistema que utiliza más de un medio de comunicación al mismo tiempo en la presentación de la información como texto, imagen, animación, video y sonido.

- 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 computador, este vocablo entró a formar parte del lenguaje habitual.

**Medio de comunicación:** es un centro emisor a través del cual se transmiten señales mediante un código. Es posible hacer una gran clasificación de los medios de comunicación: prensa, radio, televisión, internet, agencias de información.

**Información:** es un conjunto organizado de datos, que constituyen un mensaje sobre un determinado ente o fenómeno.

**Guión de contenido:** El guión contenido es el documento que precisa objetivos, estrategia pedagógica, dinámica comunicativa, secuencia lógica de contenidos y las ideas principales de diseño, para dar respuesta a una necesidad educativa específica y que debe ser concretada en la elaboración de un Software Educativo.

**Herramienta de Autor:** Son herramientas de software, que habilitan diseños instruccionales. Permitiéndoles a los educadores el diseño de la multimedia interactiva y el aprendizaje de entornos de hipermedia, sin ningún conocimiento de lenguajes de programación.

**Elementos de Contenido:** Incluirá todo lo relativo a la contención de los contenidos.

**Plantillas:** Patrón que sirve como modelo para organizar una información determinada.

**Proceso Enseñanza-Aprendizaje:** Proceso esencialmente interactivo y comunicativo, de intercambio de comunicación, compartiendo experiencias, conocimientos y vivencias, que logran una influencia mutua en las relaciones interpersonales.

**Tecnología:** La Tecnología es una característica propia del ser humano consistente en la capacidad de éste para construir, a partir de materias primas, una gran variedad de objetos, máquinas y herramientas, así como el desarrollo y perfección en el modo de fabricarlos y emplearlos con vistas a modificar favorablemente el entorno o

conseguir una vida más segura. El ámbito de la Tecnología está comprendido entre la Ciencia y la Técnica propiamente dichas, Por tanto el término "tecnológico" equivale a "científico-técnico". El proceso tecnológico da respuesta a las necesidades humanas; para ello, recurre a los conocimientos científicos acumulados con el fin de aplicar los procedimientos técnicos necesarios que conduzcan a las soluciones óptimas. La Tecnología abarca, pues, tanto el proceso de creación como los resultados. Dependiendo de los campos de conocimiento, tenemos múltiples ramas o tecnologías: mecánica, materiales, del calor y frío, eléctrica, electrónica, química, bioquímica, nuclear, telecomunicaciones, de la información.

**Proceso:** Un proceso puede ser definido como un conjunto de actividades enlazadas entre sí que, partiendo de una o más entradas las transforma, generando un resultado.

**Cursos:** Un curso es, en definitiva, un marco en el cual los distintos protagonistas del proceso (profesores, tutores y alumnos) pueden interactuar entre sí de forma instantánea, en cualquier momento, y directa, a la vez que encuentran las herramientas para hacer efectivas sus respectivas tareas.

**Sistemas de autor:** Son un tipo de programas informáticos que facilitan la creación de productos multimedia a usuarios con conocimientos de programación. Es un programa de ordenador diseñado para facilitar la creación de material educativo multimedia a profesores no especializados en informática.

**Adobe Flash:** Es una aplicación en forma de estudio de animación que trabaja sobre "Fotogramas" destinado a la producción y entrega de contenido interactivo para diferentes audiencias alrededor del mundo sin importar la plataforma.

**Proceso Desarrollo Software:** Define quien está haciendo qué, cuando y cómo alcanzar un determinado objetivo. Proporciona normas para el desarrollo eficiente de software de calidad.

**Sistema Gestor de Aprendizaje:** Sistema que gestiona los procesos de integración entre las herramientas. Gestiona las funcionalidades y maneja el contenido educativo.

**Entornos Virtuales de Aprendizaje:** Una herramienta interactiva de enseñanza y aprendizaje que permite a las instituciones educativas y a sus docentes y alumnos, acceder a más y mejores recursos para la formación.

**Repositorio:** Archivos digitales de productos intelectuales de carácter académico accesible a los usuarios con pocas o ninguna barrera e interoperables

**Intérprete:** Programa que realiza un análisis de la aplicación escrita en un lenguaje no-máquina (fácil de entender y trabajar con él) y lo convierte en lenguaje máquina entendible para el ordenador.

**Proyector:** Herramienta para proyectar los objetos de aprendizaje desarrollados.



**Metamedios:** Medio que crea otros medios. Engloba en su interior toda la producción de los diferentes medios de comunicación.

**LCMS:** Es un manejador de contenidos. Es un sistema que permite administrar el contenido.

**SCORM: *Sharable Content Object Reference Model*.** Es una especificación que permite crear objetos pedagógicos estructurados. Los sistemas de gestión de contenidos en web originales usaban formatos propietarios para los contenidos que distribuían. Como resultado, no era posible el intercambio de tales contenidos. Con SCORM se hace posible el crear contenidos que puedan importarse dentro de sistemas de gestión de aprendizaje diferentes, siempre que estos soporten la norma SCORM.

**Learning objects:** Una entidad digital, autocontenible y reutilizable, con un claro propósito educativo, constituido por al menos tres componentes internos editables: contenidos, actividades de aprendizaje y elementos de contextualización. A manera de complemento, los OA han de tener una estructura (externa) de información que facilite su identificación, almacenamiento y recuperación: los metadatos.

**Distro:** Una distribución de Linux **es** una variante de ese sistema operativo **que** incorpora determinados paquetes de software para satisfacer las necesidades de un grupo específico de autores.