



Facultad 8

# Multimedia Aprendiendo GNU/Linux



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

Autores: Yaidelyn Guerrero Cruz  
Yanet Cabrera Rodríguez  
Tutor: Ing. Harold Ordaz Valdés

Ciudad de la Habana, junio de 2007  
“Año 49 del triunfo de la Revolución”

## **Declaración de Autoría.**

Declaramos ser autores de la presente tesis y reconocemos a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los \_\_ días del mes de Julio del 2007.

Yaidelyn Guerrero Cruz

Yanet Cabrera Rodríguez

Ing. Harold Ordaz Valdés

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Autor

\_\_\_\_\_

Firma del Tutor

No hay más que asomarse a las puertas de la tecnología y la ciencia contemporáneas para preguntarnos si es posible vivir y conocer ese mundo del futuro sin un enorme caudal de preparación y conocimientos...

Fidel Castro

En primer lugar quisiera agradecer a mi padre quien me ha dado lo mejor de él: su amor, para que hoy yo este aquí. A mi hermana que es mi ejemplo, mi amiga, mi guía y mi apoyo. A mis segundos padres, mis tíos, a quienes les agradezco el haberme acogido como una hija más, y a los cuales admiro y quiero mucho. A mis hermanos, entre los que incluyo a mi prima Sonia y a Sergito. A mi madre que a pesar de la distancia, la quiero mucho. A mi nueva familia, quienes ya forman parte esencial de mi vida. A Mario a quien adoro, y quien me ha apoyado siempre, me ha comprendido y ha soportado mis malcriadeces. A mis amigos que son muchos, pero especialmente a mis amigas Amiris, Kenia y Yaine. A Lester por estar cuando lo necesité. A mis profesoras Haydeé y Alina, quienes formaron parte esencial en mi vida en la universidad. A Yanet, mi compañera de tesis, gracias por querer las cosas perfectas, por enseñarme y soportarme durante todo este tiempo. A Abel, por todo su apoyo y dedicación. A mi tutor Harold por su constante preocupación. Y a todos aquellos que de una forma u otra me han contribuido a la realización exitosa de este trabajo.

Yaidelyn Guerrero Cruz

Primeramente quiero darle las gracias a mi mamá por ser mi apoyo incondicional de toda la vida y darlo todo porque mis sueños se realicen. A mi papá por darme aliento, consejos, guiarme en todo momento y brindarme toda su confianza. A mis hermanos, abuela y familia por esperar siempre lo mejor de mí. A mis compañeros de todos estos años por toda la ayuda brindada, en especial a Ismael y Eduardo, que durante esta etapa no repararon en compartir sus conocimientos. A Lisset por ser mi confidente y amiga, por estar siempre que la necesité. A Mary, mi amiga y hermana de todos los tiempos, gracias por tu apoyo y preocupación. A Yaidelyn, mi compañera de tesis, por soportar todas mis críticas en el trabajo y mi manera tan exquisita de querer que se realicen las cosas. A Yudisbel, por brindarme todo su amor cada día y ser tan especial. Un agradecimiento a Lester y Abel por contribuir de la manera que lo hicieron en la realización de este trabajo. A mi tutor, Harold, por su preocupación y dedicación constantes para que este trabajo se realizara exitosamente.

A nuestro comandante, por ser autor intelectual de este programa tan lindo que es la Universidad de las Ciencias Informáticas y por darme la oportunidad de ser partícipe del mismo.

Yanet Cabrera Rodríguez

A nuestros queridos padres por todo el amor y la entrega, a nuestros hermanos por su apoyo y cariño, a nuestra familia por la confianza depositada y a todas las personas que hicieron posible la realización exitosa de este trabajo.

## Resumen

Las nuevas tecnologías de la información forman parte de la vida cotidiana, el cómo dominar herramientas avanzadas y los nuevos software que surgen marcan un reto para todos aquellos que desean cada día enriquecer sus conocimientos. El surgimiento de herramientas que contribuyan al estudio de diferentes temáticas hace posible que se logre un aprendizaje más rápido. El presente trabajo tiene como objetivo principal la implementación de un software con tecnología Multimedia en el que se mostrará de forma organizada información básica sobre el manejo de GNU/Linux, como un material de apoyo para el estudio de todas aquellas personas que tengan interés en este sistema operativo. Este documento recoge los resultados de todo el trabajo investigativo realizado. Se exponen argumentos para demostrar que la situación problemática requiere de una aplicación ofreciéndoles a los usuarios, a través de una interfaz sencilla, un material de estudio sobre el Sistema Operativo GNU/Linux de una forma amena y de fácil uso. Posteriormente se hace un análisis comparativo acerca de las tecnologías existentes y se selecciona la más apropiada para el desarrollo de aplicaciones que usan tecnología multimedia. Se muestran los resultados del diseño de la propuesta del sistema y finalmente se incluye el estudio de factibilidad del proyecto, en el que se determina el tiempo necesario para desarrollar la aplicación y el costo, además de los beneficios tangibles e intangibles que la misma aporta. Se plantean las conclusiones a la que se arriba con la investigación así como las recomendaciones dirigidas a mejorar la aplicación.

---

Introducción .....	1
CAPÍTULO 1.....	6
Fundamentación del tema .....	6
1.1 Introducción .....	6
1.2 El software multimedia y GNU/Linux.....	6
1.3 Análisis de otras soluciones existentes .....	9
1.4 Descripción del objeto de estudio.....	10
1.5 Conclusiones .....	12
CAPÍTULO 2.....	13
2.1 Introducción .....	13
2.2 Tendencias actuales al desarrollo de Multimedia.....	13
2.3 Fundamentación de la metodología a utilizar.....	15
2.3.1 El Proceso Unificado de Desarrollo de Software (RUP).....	15
2.3.2 Lenguaje Unificado de Modelado (UML) .....	17
2.3.3 Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA – L).....	18
2.3.4 Extreme Programming (XP).....	20
2.3.5 Relationship Management Methodology (RMM) .....	20
2.4 Herramientas para el desarrollo de software con tecnología multimedia .....	21
2.4.1 Macromedia Director .....	21
2.4.2 NeoBook.....	22
2.4.3 Macromedia FreeHand MX.....	23
2.4.4 Macromedia Flash .....	23
2.4.5 Mediator.....	24
2.4.6 Revolution.....	24
2.4.7 ToolBook .....	25
2.4.8 Adobe Illustrator .....	26
2.4.9 Adobe Photoshop .....	27
2.4.10 Wink.....	28
2.5 Lenguajes .....	28
2.5.1 Lenguajes de marcas .....	28
2.5.2 Lenguajes de programación .....	30
2.6 Conclusiones .....	31
CAPÍTULO 3.....	33
3.1 Introducción .....	33
3.2 Especificación del contenido .....	33
3.3 Descripción de la solución propuesta.....	34
3.4 Descripción de la funcionalidad.....	34
3.4.1 Requerimientos funcionales .....	34
3.4.2 Requerimientos no funcionales .....	37
3.5 Modelo de Dominio.....	39
3.5.1 Descripción del modelo de Dominio .....	39
3.5.1 Diagrama de navegación.....	39
3.6 Modelo de casos de uso del sistema .....	40
3.7 Conclusiones .....	54
CAPÍTULO 4.....	55
Construcción de la solución propuesta.....	55
4.1 Introducción .....	55
4.2 Diagramas de presentación del modelo de análisis.....	55
4.2.1 Diagrama de presentación Pantalla Principal.....	56

---

4.2.2 Diagrama de presentación Módulo Temas.....	57
4.2.3 Diagrama de presentación del Módulo Glosario.....	58
4.2.4 Diagrama de presentación del Módulo Galería de Imágenes .....	59
4.2.5 Diagrama de presentación Módulo Videos.....	60
4.2.6 Diagrama de presentación Pantalla Salir .....	61
4.2.7 Diagrama de presentación Pantalla Ejercicios .....	62
4.3 Diagrama de clases de diseño. ....	62
4.3.1 Mapeo de clases y jerarquía de Medias.....	63
4.3.2 Diagrama de clases general.....	67
4.3.3 Diagrama de clase de diseño CUS Mostrar Ayuda del Sistema .....	67
4.3.4 Diagrama de clase de diseño CUS Controlar Sonido.....	68
4.3.5 Diagrama de clase de diseño CUS Permitir Salida .....	69
4.3.6 Diagrama de clase de diseño CUS Permitir Navegación .....	70
4.3.7 Diagrama de clase de diseño CUS Mostrar Contenido de un tema. ....	71
4.3.8 Diagrama de clase de diseño CUS Interactuar con palabras calientes.....	72
4.3.9 Diagrama de clase de diseño CUS Visualizar media. ....	73
4.3.10 Diagrama de clase de diseño CUS Mostrar Galería.....	74
4.3.11 Diagrama de clase de diseño CUS Imprimir.....	75
4.3.12 Diagrama de clase de diseño CUS Mostrar Glosario. ....	76
4.3.13 Diagrama de clase de diseño CUS Interactuar con ejercicios.....	77
4.4 Modelo de implementación.....	77
4.4.1 Diagrama de componentes.....	79
4.4.2 Diagrama de componentes del paquete Temáticas .....	80
4.4.3 Diagrama de componentes del paquete glosario .....	83
4.5 Principios del diseño.....	83
4.5.1 Estándares de la interfaz de la aplicación .....	84
4.5.2 Estándares de codificación.....	85
4.6 Descripción de archivos XML .....	86
4.6.1 Descripción del archivo XML Menú .....	87
4.6.2 Descripción del archivo XML Contenido.....	88
4.6.3 Descripción del archivo XML Ayuda .....	88
4.6.4 Descripción del archivo XML Glosario.....	88
4.6.5 Descripción del archivo XML Ejercicios SU .....	89
4.6.6 Descripción del archivo XML Ejercicios VF .....	90
4.6.7 Descripción del archivo XML Imagen .....	90
4.6.8 Descripción del archivo XML Video .....	91
4.7 Diagrama de Despliegue .....	91
4.8 Conclusiones .....	92
CAPÍTULO 5.....	93
5.1 Introducción .....	93
5.2 Estimación de Tiempo de Desarrollo, Costo y Esfuerzo .....	93
5.3 Beneficios tangibles e intangibles. ....	100
5.3.1 Tangibles .....	100
5.3.2 Intangibles .....	100
5.4 Análisis de costo-beneficio. ....	101
5.5 Conclusiones .....	101
Conclusiones.....	103
Recomendaciones.....	104
Referencias bibliográficas .....	105
Bibliografía.....	107
Glosario .....	110

## Introducción

El desarrollo vertiginoso de las modernas tecnologías, ha hecho posible el surgimiento de nuevos sistemas operativos. El cómo dominarlos para que sean más accesibles a todos, constituye hoy un reto, por lo que se hace inevitable la búsqueda de nuevos métodos de aprendizaje que satisfagan las emergentes necesidades de conocimiento en el tema.

Es hoy una realidad que el software propietario trae consigo inconvenientes para Cuba; por lo que el país se ha trazado la gran misión de migrar a la plataforma informática GNU/Linux, teniendo en cuenta la relevante significación que puede verse en el ámbito político, económico y tecnológico.

Como todo proceso de cambio arrastra dificultades, se ha decidido hacer la migración con un alto nivel de preparación que evite mayores problemas.

Hoy se ven en distintos centros del país, principalmente educacionales, el comienzo de este proceso. Son varios los lugares donde se imparten conocimientos acerca de GNU/Linux, como son las universidades del país, los preuniversitarios, los institutos politécnicos de informática y algunas empresas.

En la Universidad de las Ciencias Informáticas (UCI), ente productor y exportador de software, existe hoy una facultad (la número 10) cuyos productos se desarrollan casi en su totalidad sobre plataformas libres. Este facultad es la encargada de promover e incentivar el interés por el estudio del sistema operativo GNU/Linux al resto de la Universidad. Poseen para esto un sitio Web publicado en una de las máquinas de sus laboratorios, que lo administran los mismos estudiantes de esta facultad bajo la tutoría de un profesor.

En este sitio los estudiantes pueden encontrar bibliografía acerca de este sistema operativo, así como cursos que sirven de guía para el aprendizaje de cada uno de los miembros de la facultad, y del resto del estudiantado universitario.

Sin embargo, no existe aún un medio que permita de forma interactiva y más atractiva consultar sobre el trabajo con este sistema operativo.

Esta situación se extiende a los Institutos Politécnicos de Informática (IPI), nuevo programa de la Revolución que posibilita la expansión del conocimiento de las Ciencias de la Informática. En estas instituciones se unen otros factores más que dificultan el estudio de este Sistema Operativo, como son la escasez de profesores preparados en el tema, insuficientes materiales de apoyo a la docencia y el impedimento de búsquedas en Internet, como alternativa a la poca bibliografía existente que aborden esta temática. Sintetizando las ideas se puede asegurar que: no existe suficiente material de apoyo, personal, ni medios para llevar a cabo el estudio y aplicación del Sistema Operativo GNU/Linux, por lo que es una realidad que actualmente es solo utilizado en el país por una minoría, los más atrevidos por llamarlos de algún modo; la necesidad de estos materiales que permitan consultar sobre el estudio de este sistema es hoy imprescindible.

Por tanto el **problema a resolver** es: ¿Cómo facilitar un medio que sirva de apoyo y consulta en el estudio del Sistema Operativo GNU/Linux?

El **objeto de estudio** está orientado al proceso de producción de software con tecnología Multimedia. El **campo de acción** que abarca este trabajo, incluye: El proceso de producción de software con tecnología Multimedia como material de apoyo para el estudio de GNU/Linux.

Como **idea a defender** se puede plantear:

Si se confecciona un software con tecnología Multimedia sobre el sistema operativo GNU/Linux, entonces se podrá facilitar un material de apoyo para el estudio del mismo de manera interactiva y más atractiva.

El **objetivo general** de esta investigación es desarrollar un software con tecnología multimedia que sirva de apoyo al estudio del sistema operativo GNU/Linux.

De ahí se derivan los siguientes **objetivos específicos**:

- Definir y organizar los contenidos hipermedia para la enseñanza del sistema operativo GNU/Linux mediante un software con tecnología multimedia.

- Desarrollar un software con tecnología multimedia, fiable, atractivo y de alta calidad.
- Permitir un fácil mantenimiento de la aplicación mediante la nueva incorporación de información hipermedia que se muestre en el producto multimedia.

Dentro de las **tareas** que se proponen para dar solución a los objetivos planteados están:

1. Recopilar y organizar la documentación existente acerca del sistema operativo GNU/Linux.
2. Investigar sobre las metodologías usadas para el desarrollo de software con tecnología multimedia.
3. Investigar sobre las herramientas que soportan las metodologías usadas para el desarrollo de aplicaciones con tecnología multimedia.
4. Investigar sobre las herramientas para el desarrollo de software con tecnología multimedia.
5. Analizar, diseñar e implementar el software con tecnología multimedia de manera que permita soportar nuevos contenidos hipermedia de una forma dinámica y que sea amigable para el usuario.

Con el propósito de desarrollar las tareas planteadas, se tuvieron en cuenta para desarrollar el presente trabajo, los métodos de investigación siguientes:

Métodos Empíricos: Se utilizaron estos tipos de métodos para recopilar información y recoger datos que estuviesen relacionados con las aplicaciones con tecnología multimedia y el sistema operativo GNU/Linux. Dentro este tipo de método, se utilizó **la tormenta de ideas** que facilitó ideas sobre la forma de proceder a la hora de investigar, principalmente a la hora de realizar búsquedas en Internet y en bibliografías relacionadas con el tema, además ayudó en los resultados obtenidos, la bibliografía localizada y las opiniones que fueron tomadas de los miembros de la facultad 10 específicamente del líder del proyecto “Unicornio” Abel Meneses y algunos compañeros miembros del Instituto Politécnico de Informática “Carlos Marx” en la provincia de Matanzas. También el método de **la observación** fue útil para conocer detalles importantes de las nuevas tecnologías de la informática y las comunicaciones.

Por otra parte también se encuentran los métodos teóricos utilizados, entre ellos: **La revisión bibliográfica y el de análisis y de síntesis**, la revisión bibliográfica conformada por la revisión de una serie de publicaciones, libros, documentos electrónicos, monografías. Muchas de estas bibliografías se han encontrado en Internet, páginas Web, trabajos de curso, libros electrónicos y los apuntes citas o datos que se han utilizados han sido sintetizados.

Por medio de este estudio como resultado se espera obtener un software con tecnología multimedia, y de un carácter informativo, fácil de utilizar y que sea agradable a los usuarios, permitiendo dar solución a la problemática planteada.

La actualidad y necesidad del presente trabajo se centra en que el aprendizaje es un factor permanente en la vida del hombre y se manifiesta en su actividad cotidiana. La tecnología multimedia se caracteriza por un extenso uso como medio interactivo, informativo, de orientación, medio didáctico y como libro electrónico.

Sin duda el uso de estos atractivos materiales favorecen los procesos de enseñanza y aprendizaje grupales e individuales, avivando el interés, manteniendo una continua actividad intelectual y proponiendo aprendizajes a partir de los errores, facilitando la evaluación y el control. (SALINAS 1996)

El Software Libre fomenta sobre todo una actitud, la de compartir el conocimiento, por lo que se convierte en un motor ideal para la innovación y el desarrollo, facilitando el avance de la ciencia. La necesidad de la capacitación de los estudiantes y profesionales en el trabajo con GNU/Linux, así como la barrera del alcance tecnológico a Internet a la que se enfrenta el país, exhorta a la tarea de utilizar la tecnología multimedia para facilitar nuevos medios de aprendizaje que contribuyan a la superación y el adiestramiento en este sistema operativo.

GNU/Linux es usado por millones de usuarios y existen numerosas distribuciones y cada una de ellas puede incluir cualquier número de software adicional, como algunos que facilitan la instalación del sistema y una enorme variedad de aplicaciones, entre ellas, entornos gráficos, suites ofimáticas, servidores Web, servidores de correo, servidores FTP, entre otras más.

## **Estructuración del contenido**

**Capítulo 1:** *Fundamentación del tema:* En este capítulo se hace alusión al estado del arte de la temática planteada, a la descripción del objeto de estudio, así como se analiza otra solución existente a la temática que se trata.

**Capítulo 2:** Tendencias y tecnologías actuales a considerar: En este capítulo se hace un análisis de las tendencias actuales al desarrollo de software Multimedia. Se analiza la metodología a utilizar para el análisis y diseño del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Y finalmente se exponen las propuestas de herramientas a utilizar para el desarrollo de la aplicación

**Capítulo 3:** Descripción de la solución propuesta: En este capítulo se realiza el modelado del negocio del sistema a través de un modelo de dominio. Descripción de la información que se maneja, de la aplicación que se propone así como de todos los requisitos funcionales y no funcionales que esta debe cumplir.

**Capítulo 4:** Construcción de la solución propuesta: Incluye la definición del modelo de análisis del sistema y de este el modelo de clases. Describe los diagramas de secuencia del modelo del diseño para cada realización de los casos de uso. Muestra el diagrama de clases del diseño y la descripción de cada una de las clases.

**Capítulo 5:** Estudio de la factibilidad: Incluye todo el estudio de la factibilidad del producto a desarrollar así como un análisis de los beneficios que aportará la utilización del mismo.

# CAPÍTULO 1

## Fundamentación del tema

### 1.1 Introducción

En el presente capítulo se hace alusión al estado del arte de la temática planteada, a la descripción del objeto de estudio, se analiza otra solución existente a la temática que se trata y se describe el objeto de estudio.

### 1.2 El software multimedia y GNU/Linux.

No son pocas las ventajas que hoy ofrece el software propietario. Las aplicaciones de punta a nivel mundial son propietarias, ofreciendo muy buen acabado tanto de estética, como de usabilidad de la aplicación, así como un buen soporte de hardware.

No se pueden negar los aportes que Microsoft Windows ha hecho desde su aparición al desarrollo de la Informática, sobre lo cual se puede agregar que han sido grandes e importantes. Desgraciadamente para muchos, el desarrollo de este sistema operativo se ha visto limitado precisamente por su política de imposición de sus esquemas y pautas, con o sin intención ha cerrado las puertas a opciones que se presenten diferentes a las suyas, esto ha provocado el estancamiento de su desarrollo.

La utilización de este sistema operativo en Cuba plantea notables desventajas, dadas primordialmente por las siguientes razones:

- La primera, por ser un software propietario, o sea, que exige el pago para su adquisición y por la licencia de uso.
- En segundo lugar y derivado de esta, la dificultad, en términos de imposibilidad, de contextualizar la mayoría del software producido para este sistema, para su adecuación a las necesidades de una esfera en específico.
- En tercer lugar, por los bugs o errores de programación y vulnerabilidades que posee su núcleo, escondidos en su código y que resulta imposible corregir por cuanto no se cuenta con el código

fuelle, problemas que se enfatizan sobre todo desde el momento que Cuba se conecta a Internet en octubre de 1996; y la demora en la aparición de los parches - o actualizaciones - para su corrección.

El sistema operativo GNU/Linux, clasificado dentro de los estándares del software libre, a diferencia de Windows resuelve las cuestiones planteadas anteriormente, es decir, cualquiera puede obtener el código fuente, descargándolo de Internet, usarlo e incluso modificarlo, sin pagar por ello, siempre y cuando respete la GPL (General Public License); además los errores de programación, por esta misma razón, son corregidos en poco tiempo por una amplia comunidad de programadores a nivel mundial que en cuestión de semanas, a veces días, son puestos a disposición del mundo entero.

Dicho de otra manera, GNU/Linux como software libre que es, ofrece una serie de libertades: (STALLMAN 2002)

- Ejecutar el programa con cualquier propósito (privado, educativo, público, comercial, etc.)
- Estudiar y modificar el programa, adaptándolo a las necesidades de cada usuario (para lo cuál es necesario poder acceder al código fuente)
- Distribuir copias.
- Mejorar el programa y hacer públicas las mejoras a los demás, de modo que todas las personas que usen este sistema operativo se beneficien.

Por estas razones, el país se ha trazado como meta la migración total a esta plataforma informática, teniendo en cuenta la relevante significación que puede verse desde tres ámbitos diferentes:

- Político: Representa la no utilización de productos informáticos que demanden la autorización de sus propietarios (licencias) para su explotación.
- Económico: Su utilización no implica gastos adicionales por concepto de cambio de plataforma de software, por cuanto es operable en el mismo soporte de hardware con que se cuenta actualmente.
- Tecnológico: Permite su adaptación a los contextos de aplicación, al contar con su código fuente, lo cual garantiza un mayor por ciento de efectividad, además la corrección de sus errores de programación y obtención de las actualizaciones y nuevas versiones.

El cambio de plataforma supondrá uno de los cambios más importantes de la Revolución Informática y requiere ser enfrentado con seriedad, con una visión estratégica para llegar a este con un alto nivel de preparación que permita eliminar los inconvenientes que todo cambio arrastra.

El desarrollo de las ciencias informáticas aplicadas a la Educación juega un papel sumamente importante en el logro de esta meta. Hoy son diversas las maneras en que se da a conocer este Sistema Operativo. En el país existe un portal de software libre en el que se puede encontrar noticias, foros, chat, descarga y enlaces acerca de este tema. En la UCI, una de sus facultades, la número 10, ya migró casi en su totalidad a software libre y se prepara día tras día en este sentido. En esta facultad se ha estado implementando un sitio Web, en el cual los estudiantes de la Universidad pueden interactuar, hacer preguntas, acceder a información y conocer noticias del mundo de las aplicaciones de GNU/Linux.

La existencia de varios centros como Infomed, quien asumió el Software Libre como estilo de trabajo a partir del año 1992; los Joven Club de Computación que son la institución promotora del conocimiento informático en el país, las universidades donde GNU/Linux es una asignatura opcional en el programa de estudio de carreras como Lic. Matemática, Lic. Física, Lic. Matemática Cibernética, en los Institutos preuniversitarios vocacionales del país (IPVCE) donde los estudiantes se interesan por esta temática, solo por citar algunos ejemplos, nos demuestra que existen personas que usan ésta alternativa, pero que aún son muy pocos.

Cómo atraer la atención y el interés de la mayoría hacia este Sistema Operativo, constituye hoy un reto, por lo que se hace inevitable la búsqueda de nuevos métodos de aprendizaje que satisfagan las emergentes necesidades de conocimiento en el tema.

Las tecnologías informáticas de última generación, pueden contribuir enormemente a solucionar un conjunto de problemas docentes que hoy en día existen, tales como, la carencia de bibliografía actualizada, la motivación de los estudiantes por una rama determinada del conocimiento y el poco capital humano existente preparado en algunas temáticas que resulta vital el trabajo diario en ellas.

La creación de una aplicación con tecnología multimedia con fines desarrolladores, abiertos a la creatividad permite aplicar la articulación de la tecnología computacional, desde la perspectiva de la comunicación visual, con la educación y no como mera capacitación, sino desde un punto de vista didáctico, buscando un modelo de educación que supere el direccionamiento verbalista y sea capaz de ayudar al estudiante a crecer libremente en procesos de responsabilización y de abstracción crecientes.

La característica principal de los sistemas multimedia es su gran flexibilidad así como la alta interactividad que poseen, pues permiten un aprendizaje autoguiado y autoiniciado, en el cual cada persona va construyendo su conocimiento, bien sea de manera individual o colectiva, si se agrega a esto, que evita las muchas dificultades que en países poco desarrollados se presentan al hacer uso de la red de redes, quien ofrece sistemas de enseñanza más sofisticados y con mayores posibilidades, pues resulta el medio ideal para llevar hacia los lugares más recónditos el conocimiento. (RODRIGUEZ 2006)

Hoy Cuba está haciendo un extenso uso de esta tecnología, ejemplificando su utilización en las escuelas primarias, en las cuales ha ocasionado un impacto favorable para la enseñanza de los niños, pues se abordan temáticas que contribuyen al desarrollo intelectual de los más pequeños de una forma agradable y facilita a los maestros la enseñanza. En las universidades estos medios audiovisuales aún no extienden su uso, sin embargo la Universidad de las Ciencias Informáticas ha provocado la iniciación de las mismas, implantando su utilización.

### **1.3 Análisis de otras soluciones existentes**

Internet, red de redes, como antes se ha referido, es la solución inmediata y actual a la búsqueda de bibliografía que permiten obtener un conocimiento sobre el sistema operativo GNU/Linux, sin embargo hoy no está al alcance de todos, por lo que no constituye una solución factible. La UCI es privilegiada en este aspecto, debido a que no solo sus estudiantes a partir de tercer año, pueden acceder a Internet, sino también el portal de software libre que está implementado está disponible para todos sus estudiantes y profesores, pero la UCI no es la realidad de todos los estudiantes del país, donde sí existe la carencia de estas soluciones. Por lo que se plantea que la construcción de una aplicación con tecnología multimedia que aborde esta temática resulta una solución inmediata a la barrera del alcance tecnológico que presentan algunas instituciones educativas del país.

## 1.4 Descripción del objeto de estudio

Para darle respuesta a la situación problemática planteada se propone realizar un software con tecnología multimedia de carácter informativo, que además de permitir al usuario ver un contenido organizado y bien estructurado de la temática GNU/Linux, permita interactuar con este, facilitando el estudio de cada usuario en el tiempo que este requiera y de forma independiente, tomando como punto de referencia el concepto de interactividad se plantea que:

La interactividad como demanda de acción/reacción puede tener distintos medios de expresión: tocar la pantalla, entrar texto en un determinado campo, realizar determinadas manipulaciones usando el mouse – click, doble click, arrastrar, tirar, entre otras y desde el punto de vista del usuario, interactividad es la cantidad de control que éste tiene sobre los contenidos. (EDUCAR 2007)

La verdadera interactividad, implica una modificación en la propia estructura del "relato" multimedia: se pasa de una presentación lineal donde todo el control es del autor a la multilinearidad donde el control es del usuario. El usuario es así el dueño de la secuencia y del tiempo dedicado al contenido. (EDUCAR 2007)

La propuesta de solución planteada va dirigida a personas que tengan algunos conocimientos básicos de la informática y deseen usar el software con tecnología multimedia como un material más de apoyo para el estudio sobre el sistema operativo GNU/Linux de una forma sencilla, además las temáticas que se abordan son de una complejidad mínima, orientada principalmente al trabajo modo usuario, trayendo consigo la posible iniciación en el trabajo con el software libre.

Algunos de los principios que se tienen en cuenta para la elaboración de la multimedia son: (BERROCO 2007) El principio de la múltiple entrada que plantea que todo cuanto se puede transmitir desde una aplicación multimedia “viajará” por lo que se llaman los canales de comunicación: texto, imagen o sonido. El principio multicanal establece que para lograr una buena comunicación hay que utilizar todos los canales. Un sistema multimedia es el que transmite una información mediante imagen, sonido y texto de forma sincronizada, y que hace uso adecuado de la capacidad de usar los diferentes canales de comunicación.

El principio de interactividad que es un recurso propio de sistemas informáticos y permite acceder a cualquier tipo de información rompiendo radicalmente con la linealidad o secuencialidad, con el único objetivo de reforzar el mensaje que se quiere transmitir. En una aplicación con tecnología multimedia es necesario establecer niveles de interacción siempre y cuando éstos no afecten el objetivo del mensaje original. Así, el usuario debe interactuar con la aplicación cuando sea estrictamente necesario. Además, se deben evitar los períodos de tiempo excesivamente prolongados en los que el usuario no interviene, como: una lectura de textos extensos en pantalla, secuencias prologadas de sonido e imagen animada. También debe evitarse la interacción basada en la repetición de gestos por parte del usuario.

Por otra parte, el principio de vitalidad se podría resumir diciendo que toda pantalla debe estar viva. Es decir, el usuario debe percibir la aplicación como algo que funciona autónomamente, como un mundo al que se asoma. Con ello se va más allá del principio de interactividad: en la aplicación siempre sucede algo, aunque el usuario no haga nada. El principio de libertad una vez que se ha logrado un diseño interactivo, donde el usuario no es un mero espectador de los acontecimientos, se ha conseguido uno de los principales objetivos de la aplicación: convertir al usuario en actor de la misma. El objetivo del diseñador de una aplicación multimedia es que el usuario piense que navega libremente, mientras que en realidad está inmerso en un esquema de etapas predeterminado.

También se encuentra el principio de retroalimentación el cual es un principio propio de las aplicaciones destinadas a la educación. Se trata de informar a los usuarios aprendices de sus errores, cómo corregirlos y de los progresos conseguidos desde que comenzaron con la aplicación. El principio de atención el cual tiene como objetivo de las aplicaciones multimedia mantener la atención sostenida, es decir, conseguir que el receptor mantenga una actitud continua de expectación ante la aplicación. Para ello se dispone de dos factores: la naturaleza misma de la aplicación y la apariencia, que generan respectivamente atención cognitiva y afectiva. Atención cognitiva es la que se basa en el valor de la información suministrada. Es típica de las aplicaciones profesionales o de contenidos muy particulares. Se hace especialmente atractiva para los usuarios especializados a los cuales va dirigida, y que son capaces de percibir la importancia de la información que se transmite. Para conseguirla hace falta que la información sea relevante y esté bien organizada.

La atención afectiva se basa en el lazo afectivo que se establece entre el usuario y la aplicación. Hay que señalar un recurso que contribuirá siempre a conseguir la atención afectiva: el desenlace literario. Esto consiste en que si se empieza a contar una historia se está sembrando en el receptor una inquietud por conocer el final. Ayuda a establecer atención afectiva enfocar la aplicación como una narración.

Con respecto a los estándares de codificación se señala a continuación, lo que se considera de mayor peso en la concepción de Aprendiendo GNU/Linux, además de los estándares trazados por las aplicaciones utilizadas para la concepción de la multimedia; concernientes a la nomenclatura de los recursos medias utilizados y los ficheros necesarios para la aplicación.

Nomenclatura de los recursos medias: estos serán nombrados utilizando un nombre representativo que sea identificado rápidamente, con una secuencia de dígitos de cualquier tipo siempre empezando por una letra del tipo de media y su extensión correspondiente. Ejemplo lo siguiente: fig1.jpg para las imágenes, video1.swf para los videos, animation1 fla para las animaciones y sound1.mp3 para los sonidos.

### **1.5 Conclusiones.**

Una vez analizado el estado del arte de la temática que se aborda, podemos concluir que la necesidad de preparación de profesionales en el trabajo con el sistema operativo GNU/Linux es hoy una realidad, que la tecnología multimedia puede aportar grandes ventajas para esto, teniendo en cuenta los principios que se siguen para la creación de un software con este tipo de tecnología, al contribuir a la creación de nuevos medios de apoyo al estudio y donde el usuario juega un papel fundamental al ser el único protagonista. En este capítulo se ha evidenciado la importancia que representa la descripción del objeto de estudio, así como el análisis de otras soluciones que puedan resolver la necesidad que existe hoy en día con GNU/Linux. Se ha hecho una propuesta de la audiencia que trabajará con la aplicación, teniendo en cuenta los conceptos generales relacionados para que el software cumpla con las condiciones planteadas.

## CAPÍTULO 2

### Tendencias y tecnologías actuales a considerar

#### 2.1 Introducción

En el presente capítulo, se hace un análisis de las tendencias actuales al desarrollo de software con tecnología Multimedia. Se analiza las diferentes metodologías que se pueden utilizar para el análisis y diseño del sistema, teniendo en cuenta las facilidades que puede aportar al trabajo. Y finalmente se exponen las propuestas de herramientas a utilizar para el desarrollo de la aplicación.

#### 2.2 Tendencias actuales al desarrollo de Multimedia.

En los últimos años se ha producido una gran difusión de materiales multimedia, producidos en gran escala, convirtiéndose en una poderosa herramienta que transforma a los individuos, de receptores pasivos de la información en participantes activos, en un enriquecedor proceso de aprendizaje en el que desempeña un papel primordial la facilidad de relacionar sucesivamente distintos tipos de información, personalizando la educación, al permitir a cada alumno avanzar según su propia capacidad.

El aprendizaje autodidacta permite al los estudiantes aprender a aprender, aprender a emprender y aprender a desaprender para hacer al sujeto hábil y capaz de guiar su propio aprendizaje, logros y destinos.

Sin duda el uso de estos atractivos e interactivos materiales multimedia (especialmente con una buena orientación y combinados con otros recursos: libros, periódicos, entre otros, favorecen los procesos de enseñanza. Algunas de sus principales aportaciones son las siguientes: (MÁRQUEZ 2000)

**Avivar el interés.** Los alumnos suelen estar muy motivados al utilizar estos materiales, y la motivación (el querer) es uno de los motores del aprendizaje, ya que incita a la actividad y al pensamiento. Por otro lado, la motivación hace que los estudiantes dediquen más tiempo a trabajar y, por tanto, es probable que aprendan más.

**Mantener una continua actividad intelectual.** Los estudiantes están permanentemente activos al interactuar con el ordenador y mantienen un alto grado de implicación e iniciativa en el trabajo. La versatilidad e interactividad del ordenador y la posibilidad de "dialogar" con él, les atrae y mantiene su atención.

**Orientar aprendizajes** a través de entornos de aprendizaje, que pueden incluir buenos gráficos dinámicos, simulaciones, herramientas para el proceso de la información que guíen a los estudiantes y favorezcan la comprensión.

**Promover un aprendizaje a partir de los errores.** El "feed back" inmediato a las respuestas y a las acciones de los usuarios permite a los estudiantes conocer sus errores justo en el momento en que se producen y generalmente el programa les ofrece la oportunidad de ensayar nuevas respuestas o formas de actuar para superarlos.

**Facilitar la evaluación y control.** Al facilitar la práctica sistemática de algunos temas mediante ejercicios de refuerzo sobre técnicas instrumentales, presentación de conocimientos generales, prácticas sistemáticas de ortografía, liberan al profesor de trabajos repetitivos, monótonos y rutinarios, de manera que se puede dedicar más a estimular el desarrollo de las facultades cognitivas superiores de los alumnos.

**Posibilitar un trabajo Individual y también en grupo,** ya que pueden adaptarse a sus conocimientos previos y a su ritmo de trabajo (por ello resultan muy útiles para realizar actividades complementarias y de recuperación en las que los estudiantes pueden autocontrolar su trabajo) y también facilitan el compartir información y la comunicación entre los miembros de un grupo.

Las aplicaciones multimedia interactivas tejen cinco tipos de medios: texto, sonidos, gráficos, vídeo y animación. La producción final del software con tecnología multimedia llega a ser interactiva solamente cuando estos pedazos distintos se ponen juntos en una manera organizada. Las multimedia interactivas no siguen un patrón lineal. El lector no tiene que comenzar en la tapa y leer al fondo, pueden moverse alrededor dentro de la información que el autor ha presentado. El autor presenta ideas y proporciona los acoplamientos para que sus lectores naveguen.

## **2.3 Fundamentación de la metodología a utilizar**

La ingeniería del software nace como una disciplina para aplicar los principios, técnicas y herramientas de desarrollo de software. Es la rama de la ingeniería que crea y mantiene las aplicaciones de software aplicando tecnologías y prácticas de las ciencias computacionales, manejo de proyectos, ingeniería, el ámbito de la aplicación, y otros campos. Ofrece métodos y técnicas para desarrollar y mantener software de calidad que resuelven problemas de todo tipo.

Trata con áreas muy diversas de la Informática y de las Ciencias de la Computación, tales como construcción de compiladores, sistemas operativos o desarrollos de Intranet/Internet, abordando todas las fases del ciclo de vida del desarrollo de cualquier tipo de sistemas de información.

Se puede considerar como la ingeniería aplicada al software, esto es en base a herramientas preestablecidas, la aplicación de las mismas de la forma más eficiente y óptima. No es solo de la resolución de problemas, sino más bien teniendo en cuenta las diferentes soluciones, elegir la más apropiada.

Al desarrollar un software siempre surge la interrogante qué metodología usar, el escoger correctamente hace posible que los clientes y los desarrolladores queden satisfechos, por lo que es importante saber escoger cual es la mejor para cada situación. Existen muchas tendencias de metodologías que brindan diferentes marcos que los desarrolladores pueden emplear a la hora de realizar su trabajo. Entre ella se encuentran:

### **2.3.1 El Proceso Unificado de Desarrollo de Software (RUP)**

El proceso unificado de desarrollo, RUP, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. RUP permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Esto permite a los grupos de desarrollo producir aplicaciones informáticas más robustas y flexibles que se adaptan a las necesidades de los usuarios. La correcta aplicación de RUP permite reducir los tiempos de desarrollo, aumentar la calidad de las aplicaciones y disminuir los costes de mantenimiento. “Está basado en componentes, lo cual quiere decir

que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas”. (JACOBSON *et al.* 2000)

RUP en otro concepto es un proceso que define Quién debe hacer Qué, Cuándo y Cómo debe hacerlo. Como su enfoque está basado en modelos, utiliza un lenguaje bien definido para tal fin, el UML. Está preparado para desarrollar grandes y complejos proyectos, unifica los mejores elementos de metodologías anteriores y es orientado a objetos. Dentro de sus características principales se encuentran: ”. (JACOBSON *et al.* 2000)

- Guiado por los Casos de Uso
- Centrado en la Arquitectura
- Iterativo e incremental

A través de un proyecto guiado por RUP, los requerimientos funcionales son expresados en la forma de Casos de Uso, que guían la realización de una arquitectura ejecutable de la aplicación. El ciclo de desarrollo en la vida de un producto de software se divide en 4 fases: inicio, elaboración, construcción y transición, donde RUP propone que cada una de ellas se desarrolle en iteraciones y se basa en dos disciplinas: Disciplina de Desarrollo, en la que se encuentran los flujos de trabajos Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación, Pruebas, Configuración y administración del cambio, Administración del proyecto y Ambiente dentro de la Disciplina de Soporte.

Una particularidad de esta metodología es que, en cada ciclo de iteración, se hace exigente el uso de artefactos, siendo por este motivo, una de las metodologías más importantes para alcanzar un grado de certificación en el desarrollo del software.

La metodología RUP es aplicada a una gran suma de productos y procesos de software en el mundo. Esta no se utiliza para desarrollo de aplicaciones con tecnología multimedia, sin embargo en la búsqueda de una modelación adecuada, el Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA - L) se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario. OMMMA-L modela diversos aspectos de sistema basados en el paradigma

Orientado a Objeto, utiliza el Lenguaje de Modelado Unificado y se integra dentro del Proceso Unificado de Ingeniería del Software. (SAUER. 2004)

### **2.3.2 Lenguaje Unificado de Modelado (UML)**

Lenguaje Unificado de Modelado (UML) es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. UML ofrece un estándar para describir un "plano" del sistema (modelo), incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables. (HISTCHFELD 2000)

UML es un "lenguaje" para especificar y no un método o un proceso. Se usa para definir un sistema de software, para detallar los artefactos en el sistema, para documentar y construir, es el lenguaje en el que está descrito el modelo. Se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational), pero no especifica en sí mismo qué metodología o proceso usar (HISTCHFELD 2000) .

El modelado sirve no solamente para los grandes sistemas, aun en aplicaciones de pequeño tamaño se obtienen beneficios de modelado, sin embargo es un hecho que entre más grande y más complejo es el sistema, más importante es el papel de que juega el modelado por una simple razón: "El hombre hace modelos de sistemas complejos porque no puede entenderlos en su totalidad".

**Los principales beneficios de UML son:** (HISTCHFELD 2000)

- Mejores tiempos totales de desarrollo (de 50 % o más).
- Modelar sistemas (y no sólo de software) utilizando conceptos orientados a objetos.
- Establecer conceptos y artefactos ejecutables.
- Encaminar el desarrollo del escalamiento en sistemas complejos de misión crítica.
- Crear un lenguaje de modelado utilizado tanto por humanos como por máquinas.
- Mejor soporte a la planeación y al control de proyectos.
- Alta reutilización y minimización de costos.

### **2.3.2.1 Rational Rose.**

Existen herramientas Case de trabajo visuales como el Analise, el Designe, el Rational Rose, que permiten realizar el modelado del desarrollo de los proyectos, en la actualidad la mejor y más utilizada en el mercado mundial es Rational Rose y es la que se utiliza en la modelación de este proyecto. Rational Rose es la herramienta de modelación visual que provee el modelado basado en UML. (CORPORATION. 2004)

La Corporación Rational ofrece un Proceso Unificado Racional (RUP) para el desarrollo de los proyectos de software, desde la etapa de Ingeniería de Requerimientos hasta la de pruebas. Para cada una de estas etapas existe una herramienta de ayuda en la administración de los proyectos, Rose es la herramienta del Rational para la etapa de análisis y diseño de sistemas.(CORPORATION. 2004)

Rose es una herramienta con plataforma independiente que ayuda a la comunicación entre los miembros de equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Una de las grandes ventajas de Rose es que utiliza la notación estándar en la arquitectura de software(UML), la cual permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común, además los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con otros componentes del proyecto. (CORPORATION. 2004)

### **2.3.3 Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA – L)**

El desarrollo de sistemas de software con tecnología multimedia requiere un esfuerzo cooperativo de ingenieros de software, desarrolladores de interfaces, expertos en diseño y producción de medias, expertos de dominio y usuarios. El Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L) facilita la modelación de un amplio rango de aspectos de aplicaciones multimedia interactivas de una forma integrada y comprensiva.

OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son:(SAUER)

**Vista Lógica:** modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.

**Vista de Presentación espacial:** modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea.

**Vista de Comportamiento temporal predefinido:** modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML.

**Vista de Control Interactivo:** modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML, sintácticamente igual a este último, más con la diferencia semántica de que en el orden de unir los controles interactivos y predefinidos, no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados; queriendo esto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

### **Características de OMMMA-L basadas en aprovechamiento:**

- OMMMA-L soporta modelamiento de aspectos estructurales, funcionales y dinámicos de sistemas interactivos y su interfaz de usuario.
- El modelo se centra en la funcionalidad de la perspectiva del sistema de software.
- La sintaxis del lenguaje de modelación es definida de forma precisa.
- El lenguaje de modelación tiene semánticas informales e intuitivas.

### **2.3.4 Extreme Programming (XP)**

La programación extrema es una metodología ágil utilizada en el desarrollo de software. La filosofía de XP es satisfacer al completo las necesidades del cliente, por eso, lo integra como una parte más del equipo de desarrollo. XP está diseñada para el desarrollo de aplicaciones de corto plazo que requieren un grupo de programadores pequeño, dónde la comunicación sea más factible. La comunicación es un punto importante y debe realizarse entre los programadores, los jefes de proyecto y los clientes. Esta se basa en hacer Pruebas Unitarias para ver las fallas que pudieran ocurrir, se basa en la reutilización de código es decir en la Refabricación y en la Programación en Pares la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo.

Lo fundamental en este tipo de metodología es la:

- Comunicación: Los programadores están en constante comunicación con los clientes para satisfacer sus requisitos y responder rápidamente a los cambios de los mismos.
- Simplicidad: Codificación y diseños simples y claros.
- Realimentación (Feedback): Mediante la realimentación se ofrece al cliente la posibilidad de conseguir un sistema apto a sus necesidades ya que se le va mostrando el proyecto a tiempo para poder ser cambiado y poder retroceder a una fase anterior para rediseñarlo a su gusto.
- Coraje: Se debe tener coraje o valentía para cumplir los tres puntos anteriores; Hay que tener valor para comunicarse con el cliente y enfatizar algunos puntos, a pesar de que esto pueda dar sensación de ignorancia por parte del programador, hay que tener coraje para mantener un diseño simple y no optar por el camino más fácil y por último hay que confiar en que la realimentación será efectiva.

### **2.3.5 Relationship Management Methodology (RMM)**

RMM fue la primera metodología que se hizo completa, con una definición de fases y no únicamente un modelo de datos, para el desarrollo de software con tecnología multimedia. Está basada en los conceptos implantados en el Modelo de diseño de hipertexto HDM, es decir en las entidades y los tipos de entidades y está basado en un modelo de datos relacional, lo cual se ajusta perfectamente a la gran mayoría de las

aplicaciones existentes. Su objetivo es que la navegación mejore al hacer un análisis de las entidades del sistema. Los mecanismos de acceso a la información son excesivamente simples. Los conceptos de slices y m-slices consisten en la agrupación de datos de una entidad en diferentes pantallas. Estos conceptos forman parte de uno de los aportes más importante de esta metodología.

La metodología RMM propone un proceso basado en 7 fases o etapas en las que el diseñador va modelando la estructura de la aplicación y las posibilidades de navegación de la misma. Exige que se obtenga el esquema Entidad-Relación que gobierna la lógica de la aplicación, pero muchos de los módulos pertenecientes a la aplicación no quedan documentados bajo este esquema. Además, esta metodología es de poca utilidad cuando la volatilidad de la información es baja o de características estables, que es la situación de la información involucrada en un proyecto tipo tutorial. Sólo propone actividades y productos concretos para las fases de análisis y de diseño.

En RMM se representa la aplicación mediante un modelo, denominado RMDM (Relationship Management Data Model). Este modelo es un enriquecimiento del modelo Entidad-Relación y permitirá representar a las aplicaciones multimedia. En este modelo podemos encontrar elementos propios de la propuesta del modelo Entidad-Relación (entidades, atributos y otros) aunque con las extensiones de HDM.

## **2.4 Herramientas para el desarrollo de software con tecnología multimedia**

### **2.4.1 Macromedia Director**

La herramienta Macromedia Director es un programa de autor de fácil manejo. Permite la combinación de texto, gráficos, sonido, animación y vídeo en un documento que se reproduce en el ordenador y que es presentado con múltiples detalles. La filosofía seguida por este programa es la de una línea de tiempo durante el cual irán sucediendo diferentes acontecimientos según vayamos necesiéndolos. Este proceso no tiene por qué ser necesariamente lineal ni continuo sino que permite detenerse en un punto del tiempo y saltar de un punto a otro en esa línea temporal. Director tiene un lenguaje propio de programación “Lingo”, con el que se consigue comportamientos muy sofisticados, pero por defecto hay comportamientos y rutinas que ya vienen preparados.

Macromedia Director incorpora un rango de nuevas capacidades para satisfacer las necesidades evolutivas del desarrollador actual, al mismo tiempo que conserva todo el poder de sus características centrales. Incluye una integración transparente y completa con la familia de productos Macromedia Flash MX, adopción de interfaces eficientes para el usuario de Macromedia MX, soporte para Mac OS X, nuevas y mejores eficiencias en el flujo de trabajo y la habilidad para crear contenido accesible para que las presentaciones enriquecidas de Director puedan ser disfrutadas por personas con discapacidades. (WIKIPEDIA 2007)

### 2.4.2 NeoBook

NeoBook permite crear y distribuir aplicaciones para Windows sin necesidad de programar. Los usuarios que no tengan muchas experiencias en el manejo de esta herramienta, pueden de forma rápida combinar texto, gráficas, sonido, animación y otros elementos para crear programas multimedia interactivos. Puede incluso crear un programa de instalación para su proyecto completo con compresión y capacidad para múltiples discos. Dentro de sus características más importantes se puede decir que: (ORTIZ 2007)

- Usa acciones simples de arrastrar y soltar para crear complejas aplicaciones.
- Importa imágenes e ilustraciones creadas con tu programa de dibujo o diseño favorito.
- Importa documentos de texto realizados por tu programa procesador de texto o usa el editor de texto incluido en NeoBook.
- Despliega sitios Web y contenido directo de Internet dentro de tu aplicación.
- Crea y envía correo electrónico directamente desde tu aplicación.
- Usa el editor de texto incluido para crear texto con tablas, múltiples fuentes y comandos de hipertexto.
- Desarrolla aplicaciones avanzadas utilizando el poderoso lenguaje de scripts incluidos.
- Incluye argumentos condicionales, variables, lectura y escritura de archivos, procesamiento de cadenas, etc. Los comandos del scripts pueden ser escritos manualmente por usuarios experimentados o seleccionados de una lista y añadidos llenando un simple cuestionario.
- Usa efectos de transición entre diferentes pantallas como disolvencias y ondas.
- Almacena y extrae información desde archivos externos y el registro de Windows.
- Crea aplicaciones con ventanas de forma personalizada.
- Escucha sonidos, toca música, abre videos, animaciones y otros archivos multimedia.

- Produce caricaturas animadas usando la utilidad incluida NeoToon™.

### **2.4.3 Macromedia FreeHand MX**

Programa de dibujo e ilustración vectorial, sumergido en un espacio de trabajo flexible e integrado con Macromedia Studio MX racionaliza el desarrollo de los proyectos de Macromedia Flash MX, diseñando proyectos completos en un solo documento.

Permite generar gráficos vectoriales dirigidos a medios impresos y crea gráficos para publicaciones en Internet. Además, ofrece total integración con Macromedia Flash, con lo que se puede arrastrar y colocar muy fácilmente comandos de navegación de ActionScript, y herramientas de extrusión para añadir objetos 3D, vector, borrado y fusión. Incluso se pueden importar películas Flash (SWF) más complejas a un documento de FreeHand MX con más eficiencia. Los archivos SWF colocados se pueden editar con la nueva capacidad de ejecutar y editar de Macromedia Flash MX. (CIBERAULA 2007)

### **2.4.4 Macromedia Flash**

Hace ya algunos años la aparición de flash con sus diferentes versiones, ha traído hasta nuestras pantallas todo un mundo de interactividad, movimiento y color. Su versión 8.0 perfecciona la creación de contenidos interactivos e incorpora innovaciones en vídeo, opciones de texto y filtrado.

Flash diseña gráficos de vectores y gráficos definidas como puntos y líneas en lugar de píxeles. Es decir que los vectores son como un conjunto de instrucciones matemáticas que por medio de valores le dan forma a una imagen. Además permite incluir audio comprimido en diversos formatos como el mp3, importar gráficos creadas con otros programas, formularios y algo de programación. Flash es independiente del navegador y el plugin es universal, por lo que las animaciones diseñadas con este programa se verán casi idénticamente en cualquier plataforma y navegador. La única desventaja que tienen las películas Flash, es que para poder visualizarlas, es necesario tener instalado el plugin, aunque a partir de la versión 4.0 de los navegadores, el plugin ya se incluye dentro de la instalación. Flash una tecnología con mucho futuro por su funcionamiento y facilidad de uso donde cualquiera puede crear sus primeras animaciones luego de algunas horas de trabajarlo. Permite realizar diseños de alta calidad,

puede trabajar con archivos importados (extraídos de otros programas y con otros formatos), puede agregar sonidos, sus imágenes son livianas, fáciles de animar e incluso puede incluir un alto grado de interactividad.

Es usado por más de un millón de profesionales y con una presencia que llega a más del 98% de los ordenadores personales conectados a Internet y más de 100 fabricantes de equipos originales OEM (Original Equipment Manufacturer) están incorporando Flash en sus dispositivos.

### **2.4.5 Mediator**

Sin dudas para el trabajo con presentaciones multimedia Mediator, es una poderosa herramienta con la cual podemos lograr un aspecto profesional en las mismas. Mediator consiste en dos programas: Diseñador de Mediator (Mediator Designer) y el espectador de Mediator (Mediator Viewer). El Diseñador de Mediator es donde se crea los proyectos. Este modo también incluye el modo de prueba, que es donde usted prueba el proyecto va diseñando, este puede compararse con el espectador, solo que su propósito es ir probando el proyecto dentro del diseñador, sin necesidad de buscar el archivo para ejecutarlo. El espectador de Mediator es donde se muestra proyecto después de haber guardado el archivo. (SERGIO MARTÍNEZ MAILLO 2007)

### **2.4.6 Revolution**

Esta es la hermana joven de MetaCard. Esta desarrollada sobre MetaCard por la empresa Run Revolution Ltd. Tiene un lenguaje de programación de alto nivel que se denomina "Transcript". Revolution permite crear aplicaciones con un interfaz de usuario y comportamiento propios del sistema diana, para la mayoría de las plataformas existentes en la actualidad, como son Mac OS X, Mac OS Classic, Windows desde el 95 hasta el XP, GNU/Linux y nueve tipos de sistemas Unix, así como CGIs y aplicaciones de terminal, sin modificar el código escrito. La facilidad de uso es también una de las principales ventajas de esta herramienta, ya que permite usar la opción de arrastrar y soltar o drag and drop de su paleta de controles, para crear la interfaz de usuario de una aplicación. La labor del desarrollador se facilita notablemente con la inclusión de un depurador de código o debugger, con el que poder localizar fácilmente los errores

cometidos en la programación y la posibilidad de colorear, dar formato automático y elegir el estilo de texto que se utilizará para mostrar el código.

Se destacan, entre otras características, el acceso a bases de datos que usen SQL a través de ODBC o directamente en el caso de Oracle, MySQL, PostgreSQL y Valentina, esta última característica sólo se incluye en la edición profesional. El soporte de protocolos HTTP y FTP, así como de sockets para implementar cualquier protocolo de Internet, o el acceso a otras tecnologías específicas de cada plataforma del tipo de QuickTime, AppleScript, AppleEvents o Window registry, son otras de sus cualidades. Se resalta a su vez, la presencia de SDKs (Software Development Kits) para crear módulos en cualquier lenguaje compilado. (AGUSTIN 2007)

### 2.4.7 ToolBook

Es una herramienta de programación que utiliza el lenguaje “OpenScript”, de fácil comprensión y gran potencia. Permite la construcción de forma rápida y sencilla de aplicaciones que combinan los elementos típicos de las aplicaciones Windows (ventanas, menús, iconos, entre otros) con recursos multimedia. Una de las grandes ventajas de este sistema de autor es la posibilidad que brinda para construir interfaces complejas, en cuanto a sus potencialidades y recursos, de una forma sencilla. ([EDUTEKA](#))

Con ToolBook podemos trabajar en dos niveles o modalidades:

**Nivel de autor:** Permite construir la aplicación multimedia.

**Nivel lector:** Permite utilizar la aplicación multimedia.

A nivel de autor, la construcción de una aplicación con ToolBook se basa, como su nombre indica, en construir un libro, diseñando las páginas que lo forman, creando y modificando los objetos incluidos en las páginas y escribiendo el programa con el lenguaje OpenScript. Los elementos utilizados por ToolBook para la creación de aplicaciones, pueden ser organizados jerárquicamente:

***Libro (Book) – Fondo (Background) – Página (Foreground) – Objetos***

Los **libros de ToolBook**, constituyen por si mismos una aplicación como por ejemplo una lección o unidad didáctica y permiten reunir varios libros conformando una aplicación global en la que se incluyen diferentes aplicaciones o libros, de este modo, por ejemplo podríamos reunir todas las unidades didácticas

de una determinada materia o curso de modo que para el estudiante se presentasen como una única aplicación.(EDUTEKA)

Las **páginas de ToolBook** se corresponden con las pantallas diseñadas para cada libro, en cada una de ellas se mostrará una información diferente, para ello se utiliza una programación basada en objetos. Se puede decir que los objetos son los elementos que se incorporan en una página (imágenes, botones, cuadros, animaciones, campos de texto, entre otros). La funcionalidad de los objetos viene determinada por las propiedades de los mismos y por las acciones que pueden realizarse a partir de ellos. Para determinar estas acciones ToolBook utiliza el lenguaje OpenScript. Existe la posibilidad de crear grupos de objetos, esto es cuando se precisa que un conjunto de objetos se comporten de un mismo modo y respondan conjuntamente a una acción del usuario.

Las últimas versiones de ToolBook han supuesto un giro en esta herramienta de autor desde la Enseñanza Asistida por Ordenador hacia la enseñanza basada en Web (WBT: Web Based Training). De este modo el sistema permite distribuir las aplicaciones en formato CD-ROM, o desde la red (Intranet o Internet), exportando el libro a formato HTML.

### **2.4.8 Adobe Illustrator**

Adobe Illustrator es un programa para diseño gráfico, desarrollado y comercializado por Adobe Systems, que ha definido en cierta manera el lenguaje gráfico contemporáneo mediante el dibujo vectorial. El software Illustrator contiene opciones creativas, un acceso más sencillo a las herramientas y una gran versatilidad para producir rápidamente gráficos flexibles para su impresión, vídeo, publicación en la Web y dispositivos móviles. Las principales ventajas de Illustrator son el manejo avanzado de tipografía y la consistencia de los colores a la hora de imprimir. (WIKIPEDIA)

Entre las novedades que ofrece esta herramienta se encuentran:

Adobe Illustrator CS provee opciones que nos permiten mover contenido de Illustrator hacia otros componentes de Adobe Creative Suite, incluyendo Adobe InDesign CS, Adobe GoLive y Adobe Photoshop CS, con facilidad.

Una nueva interface de impresión asegura resultados rápidos y confiables.

El soporte mejorado de Adobe PDF y la avanzada integración con Adobe Acrobat 6.0 funcionan para dar a Illustrator soporte para PDF 1.5, lo que incluye layers PDF.

Illustrator CS cuenta con mejoras en todo su desempeño: la impresión, la visualización de archivos de gran tamaño, el manejo de archivos con múltiples capas, guardar y exportar artes en formatos nativos, de Web o de impresión, es ahora mucho más rápido que con las versiones anteriores. Además, Illustrator CS aprovecha todas las ventajas de las plataformas Windows y Macintosh, incluyendo las nuevas Power Mac G5 de Apple.

### 2.4.9 Adobe Photoshop

Es una aplicación de edición y retoque de imágenes bitmap, jpeg, gif, entre otros, elaborada por la compañía de software Adobe inicialmente para computadores Apple pero posteriormente también para plataformas PC con sistema operativo Windows.

Photoshop en sus primeras versiones trabajaba en un espacio bitmap formado por una sola capa, donde se podían aplicar toda una serie de efectos, textos, marcas y tratamientos. En cierto modo tenía mucho parecido con las tradicionales ampliadoras. En la actualidad lo hace con múltiples capas. (WIKIPEDIA)

Photoshop se ha convertido, casi desde sus comienzos, en el estándar mundial en retoque fotográfico, pero también se usa extensivamente en multitud de disciplinas del campo del diseño y fotografía, como diseño Web, composición de imágenes bitmap, estilismo digital, fotocomposición, edición y grafismos de vídeo y básicamente en cualquier actividad que requiera el tratamiento de imágenes digitales. Aunque el propósito principal de Photoshop es la edición fotográfica, este también puede ser usado para crear imágenes, efectos, gráficos y más, en muy buena calidad.

Photoshop soporta muchos tipos de archivos de imágenes, como BMP, JPG, PNG, GIF, entre otros, pero tiene ciertos formatos de imagen propios como lo son: (WIKIPEDIA)

PSD (Photoshop Document): Es un formato que guarda una imagen como un grupo de capas, métodos de fusión, colores, textos, máscaras, canales de color, canales alfa, trazados, formas, configuración de tonos,

entre otras. Éste es un formato muy popular que incluso es soportado por programas de la competencia. Este formato te permite trabajar con distintas capas después de haber cerrado el programa, al contrario que el .jpg, pero sin embargo ocupa mucho más espacio y no se puede abrir con programas como el visor de imágenes y fax de Windows, luego es necesario tener un programa que lea más formatos de imagen si no tienes Photoshop. (WIKIPEDIA)

- PSB: Es una nueva versión del formato PSD, diseñado especialmente para archivos mayores a 2 GB.
- PDD: Es una versión del PSD que solo soporta las opciones del programa discontinuado PhotoDeluxe.

### 2.4.10 Wink

Hace unos años Macromedia creó Robodemo (ahora llamado Captivate), una aplicación enfocada a la creación de tutoriales o demostraciones interactivas de software, que registraba el uso de cualquier aplicación, así como toda la actividad realizada en la pantalla del ordenador. De este modo, se podían crear simulaciones interactivas basadas en la tecnología de Flash, sin necesidad de tener conocimientos de programación o Flash. El gran inconveniente de Robodemo o Captivate es el precio. Ahora se dispone de una herramienta similar y completamente gratis: Wink, la cual es una herramienta que permite realizar presentaciones y tutoriales y ofrecerlas en formato flash. Algunas de las características más interesantes son:

Freeware: Distribuido como freeware tanto para uso personal como para uso comercial.

Multi-Plataforma: Válido para Windows y para varias versiones de GNU/Linux.

Audio: Permite la grabación de voz para una mejor explicación del tutorial o la demo.

Soporte Multilingual

Formatos de Salida: Macromedia Flash, Standalone EXE, PDF, PostScript, HTML o casi cualquier formato de imagen. Así pues se podría usar Flash/html para la Web, exe para la distribución en usuarios de PEC y PDF en la creación de manuales para imprimir.

## 2.5 Lenguajes

### 2.5.1 Lenguajes de marcas

Un **lenguaje de marcado** o **lenguaje de marcas** es una forma de codificar un documento que, junto con el texto, incorpora etiquetas o marcas que contienen información adicional acerca de la estructura del

texto o su presentación. El lenguaje de marcas más extendido es el HTML, fundamento del World Wide Web. Históricamente, el marcado se usaba y se usa en la industria editorial y de la comunicación, así como entre autores, editores e impresores. (WIKIPEDIA)

Se suele diferenciar entre tres clases de lenguajes de marcado:

**El marcado de presentación:** es aquel que indica el formato del texto. Este tipo de marcado es útil para marcar la presentación de un documento para su lectura, pero resulta insuficiente para el procesamiento automático de la información. El marcado de presentación resulta más fácil de elaborar, sobre todo para cantidades pequeñas de información. Sin embargo resulta complicado de mantener o modificar, por lo que su uso se ha ido reduciendo en proyectos grandes en favor de otros tipos de marcado más estructurados.

**El marcado de procedimientos** está enfocado hacia la presentación del texto, sin embargo, también es visible para el usuario que edita el texto. El programa que representa el documento debe interpretar el código en el mismo orden en que aparece.

**El marcado descriptivo o semántico** utiliza etiquetas para describir los fragmentos de texto, pero sin especificar cómo deben ser representados, o en que orden. Los lenguajes expresamente diseñados para generar marcado descriptivo son el SGML y el XML.

### 2.5.1.1 XML (Extensible Markup Language).

**XML** no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. Es un metalenguaje que define la sintaxis utilizada para definir otros lenguajes de etiquetas estructurados. (MENDOZA) No se trata de un lenguaje de programación. Es un lenguaje estándar que establece un formato para la codificación de datos e información. Algunas de sus destacables características son su conjunto de marcas abiertas y ampliables, su distinción entre la estructura y presentación de documentos, su gestión avanzada de hipervínculos o su modularidad. XML es un lenguaje de marcas con el que se consigue la portabilidad de los datos. Flash permite la interpretación de estos datos e incluso la generación de formatos propios.

XML no es una excepción al soporte de **Actionscript**. El uso del **objeto XML**, destinado exclusivamente a la gestión de archivos y contenidos formateados en este estándar, permite a una película Flash importar y

exportar fácilmente información desde y hacia lenguajes de servidor o bases de datos. XML se encarga de estructurar estos datos de forma tal que puedan ser leídos e interpretados sin problemas por cada una de las partes. (MENDOZA)

### 2.5.2 Lenguajes de programación

Un lenguaje de programación es un lenguaje que puede ser utilizado para controlar el comportamiento de una máquina, particularmente una computadora. Consiste en un conjunto de reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos, respectivamente. Un lenguaje de programación permite a un programador especificar de manera precisa: sobre qué datos una computadora debe operar, cómo deben ser estos almacenados y transmitidos y qué acciones debe tomar bajo una variada gama de circunstancias. Todo esto, a través de un lenguaje que intenta estar relativamente próximo al lenguaje humano o natural. Los lenguajes de programación se determinan según el nivel de abstracción, según la forma de ejecución y según el paradigma de programación que poseen cada uno de ellos. (WIKIPEDIA) Según el paradigma de programación se clasifican en lenguajes orientados a objetos.

En la Programación Orientada a Objetos (POO u OOP según siglas en inglés) se definen los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos) comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas módulos más fáciles de escribir, mantener y reutilizar. (WIKIPEDIA)

#### 2.5.2.1 ActionScript

Es uno de los lenguajes de programación orientado a objetos (OOP), utilizado en especial en aplicaciones Web animadas realizadas en el entorno Flash, la para añadir dinamismo al panorama Web. Fue lanzado con la versión 4 de Flash, y desde entonces hasta ahora, ha ido ampliándose poco a poco, hasta llegar a niveles de dinamismo y versatilidad muy altos en la versión 8 de Flash. (ENCICLOPEDIA)

##### 2.5.2.1.1 ActionScript 2.0

El ActionScript es el lenguaje de programación que ha utilizado Macromedia Flash desde sus comienzos. A grandes rasgos, podemos decir que el ActionScript nos permitirá realizar con Flash todo lo que nos propongamos, ya que nos da el control absoluto de todo lo que rodea a una película Flash. Absolutamente de todo. El ActionScript es, como su nombre indica, un lenguaje de script, esto quiere decir que no hará falta crear un programa completo para conseguir resultados, normalmente la aplicación de fragmentos de código ActionScript a los objetos existentes en nuestras películas nos permiten alcanzar nuestros objetivos. Presenta muchísimos parecidos con el Javascript; si conoce Javascript, la sintaxis y el estilo de ActionScript le resultarán muy familiares. En la mayor parte de las ocasiones, no será necesario "programar" realmente, Flash pone a nuestra disposición una impresionante colección de "funciones" (entiéndase "funciones" como "código ActionScript que realiza una función determinada") ya implementadas que realizan lo que buscamos, bastará con colocarlas en el lugar adecuado. (FREDDIE 2004)

Una de las principales características del ActionScript 2.0 es que está realmente orientado a objetos. Esto aumenta su capacidad enormemente, y facilita el acercamiento de programadores experimentados en otros lenguajes, y familiarizados con los conceptos de la programación orientada a objetos. Aún así, al ser el Flash un entorno de diseño gráfico sigue poseyendo ciertas restricciones desde el código y por ser un lenguaje orientado a objeto, presenta las siguientes características: abstracción, encapsulamiento, herencia y polimorfismo.

ActionScript 2.0 resulta además el aliado perfecto para ajustar y crear la arquitectura de un proyecto orientado a objetos, así como para comprender cómo los componentes de interfaz y las subclases de clip de película encajan en una aplicación de Flash bien estructurada.

### **2.6 Conclusiones**

Una vez analizadas las características más importantes de las tecnologías existentes y las tendencias actuales para el desarrollo de software con tecnología multimedia se arriba a la conclusión de utilizar el proceso de desarrollo de software basado en RUP, como lenguaje de modelado UML y OMMMA – L como extensión del mismo para productos multimedia, teniendo en cuenta que en la actualidad, OMMMA – L se evalúa en diferentes escenarios, como proyectos industriales para la especificación de servicios de

información multimedia, y se investiga características adicionales de sincronía para su especificación en el lenguaje y la formalización de un modelo para la composición dentro y entre los diferentes diagramas de comportamiento.

El software con tecnología multimedia será desarrollado con la herramienta de autor Macromedia Flash, más específicamente en su versión 8.0, debido a que presenta algunas características novedosas como: optimización de fuentes, bibliotecas integradas, mayor potencia de animación, mayor potencia gráfica, mejoras en la importación de vídeo. El diseño de la misma se creará con la herramienta de diseño Adobe Illustrator, el tratamiento de imágenes con la herramienta Photoshop y la creación de videos o animaciones que tratan las temáticas de GNU/Linux con la herramienta Wink.

## CAPÍTULO 3

### Descripción de la solución propuesta

#### 3.1 Introducción

En el presente capítulo se describe y modela la solución propuesta, lo cual tiene como objetivo, facilitar un medio que brinde apoyo al estudio del sistema operativo GNU/Linux.

Además se enumeran los requerimientos funcionales y no funcionales que debe tener el software que se propone, lo que permite hacer una concepción general de la aplicación, e identificar mediante un Diagrama de Caso de Uso, las relaciones de los actores que interactúan con el sistema. Centrándose en el Proceso Unificado de Desarrollo de Software, haciendo uso del Lenguaje Unificado de Modelado (UML Unified Model Language) extendido con el Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L).

#### 3.2 Especificación del contenido

El contenido que se abordará dentro del software con tecnología multimedia, estará dividido inicialmente en 5 temas, que a su vez contienen subtemas. El tema 1 tratará de dar una introducción al usuario de lo que es un sistema operativo, principales características, los principales conceptos de software libre, la filosofía de trabajo y la importancia del uso de este sistema operativo. El tema 2 aborda los contenidos acerca de las diferentes distribuciones de GNU/Linux, los principales conceptos de distribuciones, ejemplos de las mismas, la existencia en Cuba y curiosidades sobre la Fauna de GNU/Linux. El tema 3 ya adentra un poco más al usuario sobre el trabajo en modo consola dentro de este. El tema 4 trata sobre el trabajo en entorno de escritorio, para usuarios finales, y el tema 5 muestra como trabajar con algunas aplicaciones dentro de este sistema operativo. Todo lo anteriormente abordado se muestra en un primer módulo, llamado Temas. El resto de los módulos del software, constituyen un apoyo al contenido que se aborda en los diferentes temas enriqueciendo los mismos a través del uso de las medias, estos módulos son: Galería de imágenes y Galería de videos, Glosario de términos y Ejercicios.

### 3.3 Descripción de la solución propuesta

La solución propuesta consiste en la elaboración de un software con tecnología multimedia que contenga elementos de apoyo para el estudio del sistema operativo GNU/Linux, permitiendo de una forma interactiva y más atractiva este estudio. Teniendo en cuenta todo lo expuesto anteriormente en la sección de metodologías y herramientas, se diseñará la aplicación haciendo uso de la metodología RUP usando OMMMA-L como extensión de UML para el modelado de la misma, Se implementará la aplicación utilizando la herramienta de autor Macromedia Flash en su versión 8.0 y se realizará el diseño gráfico en el Adobe Illustrator.

La aplicación estará dividida en cinco módulos (*Temas, Galería de Imágenes, Galería de Videos, Glosario y Ejercicios*), permitiendo la incorporación dinámica de nuevos temas, ejercicios y medias.

### 3.4 Descripción de la funcionalidad

#### 3.4.1 Requerimientos funcionales

##### Presentación

Referencia	Función
R1	Mostrar presentación inicial del producto.
R2	Mostrar Pantalla principal del producto.

##### Módulo Temas

Referencia	Función
R3	Mostrar temas en forma de menú.
R4	Permitir seleccionar el tema deseado por el usuario.
R5	Visualizar el texto del tema seleccionado.
R6	Mostrar palabras calientes del tema.
R7	Interactuar con palabras calientes.
R8	Mostrar imagen o video referente a las palabras calientes en una ventana desplegable.
R9	Permitir al usuario la opción de abrir y cerrar la ventana desplegable

	(visor) cuando este lo desee, sólo si ya fue visualizado alguna imagen o video.
<b>R10</b>	Mostrar descripción de imagen o video en la ventana desplegable.
<b>R11</b>	Permitir en la ventana desplegable la opción en caso de ser video, de controlar las operaciones de video.

#### Modulo Galería de Imágenes

Referencia	Función
<b>R12</b>	Mostrar al usuario todas las imágenes contenidas en la aplicación, con una resolución menor.
<b>R13</b>	Permitir al usuario seleccionar imagen que desea ver.
<b>R14</b>	Mostrar la imagen seleccionada de manera ampliada.
<b>R15</b>	Permitir la navegabilidad entre las imágenes, es decir, ver la anterior y la próxima.
<b>R16</b>	Mostrar descripción de la imagen visualizada.

#### Módulo Galería de Videos

Referencia	Función
<b>R17</b>	Mostrar al usuario las imágenes con una resolución pequeña que identifican todos los videos contenidos en la aplicación.
<b>R18</b>	Permitir al usuario seleccionar el video que desea ver, a partir de la imagen que lo identifica.
<b>R19</b>	Permitir al usuario opciones de reproducción de video.
<b>R20</b>	Mostrar descripción del video visualizado.
<b>R21</b>	Permitir la navegabilidad entre los videos, es decir, ver próximo video y video anterior.

#### Módulo Glosario

Referencia	Función
------------	---------

<b>R22</b>	Mostrar la definición de todos los términos que presenten dificultad de comprensión.
<b>R23</b>	Permitir al usuario seleccionar una letra y ver las palabras que comienzan con esta letra ordenadas alfabéticamente, cada una con su significado correspondiente.

#### Módulo Ejercicios

Referencia	Función
<b>R24</b>	Mostrar al usuario los distintos tipos de ejercicios que puede realizar.
<b>R25</b>	Permitir al usuario escoger el tipo de ejercicio que desea realizar.
<b>R26</b>	Permitir al usuario ver respuesta correcta del ejercicio, en cualquier momento, incluso sin haber contestado.
<b>R27</b>	Permitir realizar más ejercicios del mismo tipo en todo momento, aún cuando no se ha contestado el ejercicio actual.
<b>R28</b>	Mostrar un mensaje cuando ya se hayan agotado todos los ejercicios del tipo que se está realizando.

#### Requerimientos generales

Referencia	Función
<b>R30</b>	Permitir la navegación por todas las pantallas de la aplicación en todo momento, independientemente de la pantalla donde se encuentre el usuario. Permitir volver a la pantalla principal en cualquier momento.
<b>R31</b>	Permitir el control de audio del sistema desde cualquier pantalla.
<b>R32</b>	Permitir la salida de la aplicación cuando el usuario lo desee.
<b>R33</b>	Mostrar ayuda.
<b>R34</b>	Mantener siempre el último estado del módulo Temas, al navegar por el resto de la aplicación.
<b>R35</b>	Permitir imprimir el contenido de los temas que se exponen en el módulo Temas, además de los términos con su significado existentes en el módulo Glosario.

### 3.4.2 Requerimientos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener. No se trata de describir lo que el software hará sino como lo hará. A continuación se muestran los requerimientos no funcionales de la aplicación.

#### Apariencia o interfaz externa:

- ✓ Diseño sencillo, permitiendo la utilización del sistema de manera fácil.
- ✓ Diseño encuadrado para resoluciones de 800x600, y preparado para verse en otras resoluciones.
- ✓ La profundidad de color será de 24 bits.

#### Requisitos de implementación.

- ✓ Todos los textos e imágenes de la galería deberán cargarse de manera dinámica.
- ✓ Deberá implementarse con la herramienta Flash, versión 7.0 o superior, para obtener un producto multiplataforma.
- ✓ Los datos se almacenarán en XML, así se evita el uso de un gestor o servidor de datos.

#### Servicios generales.

Los servicios generales como: sonido, salir, imprimir siempre estarán visibles al usuario durante toda la navegación que realice por las pantallas de la aplicación.

- ✓ Especificaciones del servicio **sonido**
  - 1) El botón sonido funcionará como un conmutador ON/OFF activando o desactivando según corresponda.
  - 2) Una vez desactivado el sonido, este no se volverá a activar hasta que el usuario no lo vuelva a solicitar.
- ✓ Especificaciones del servicio **salir**
  - 1) Una vez activado este botón se mostrará un mensaje de confirmación de salida y a continuación los créditos y seguido se procederá a la salida total de la aplicación.

#### Seguridad.

La multimedia estará en soporte CD-R, no permitiéndose que se modifique su contenido, excepto por las personas autorizadas a añadir o modificar los contenidos que en ella se abordan. La información

contenida en la multimedia no necesitará estar encriptada, debido a que no hay información sensible, por tanto el contenido de los ficheros se encontrará en texto claro.

### **Político – Culturales.**

Se implementará una multimedia que garantice la integridad de la información, controlando que esté acorde con los valores de nuestra sociedad.

### **Sistema operativo.**

La aplicación podrá ejecutarse en las plataformas de Windows y GNU/Linux.

### **Requerimientos mínimos.**

Se requiere de Flash Player 8.0 para un buen funcionamiento. Los requerimientos de hardware mínimos exigidos para Flash Player 8.0 sobre Windows son:

- Procesador Intel Pentium II de 600 MHz (o equivalente) y versiones posteriores
- 128 MB de RAM.
- Pantalla de 16 bits de 800 x 600 (se recomienda de 32 bits)
- 200 MB de espacio en disco disponible

Para GNU/Linux:

- Procesador Intel (800 MHz o más rápido)
- 64 MB de RAM y 32 MB de memoria gráfica

### **Compatibilidad.**

<b>Windows</b>	
<b>Plataforma</b>	<b>Navegador</b>
Windows 98	Microsoft Internet Explorer 5.x, Netscape 4.7, Netscape 7.x y superior, Firefox 1.x, Mozilla 1.x, AOL 8 u Opera 7.11
Windows Me	Microsoft Internet Explorer 5.5, Netscape 4.7, Netscape 7.x y superior, Firefox 1.x, Mozilla 1.x, AOL 8 u Opera 7.11
Windows 2000	Microsoft Internet Explorer 5.x, Netscape 4.7, Netscape 7.x o superior, Firefox 1.x, Mozilla 1.x, CompuServe 7, AOL 8 u Opera 7.11
Windows XP	Microsoft Internet Explorer 6.0, Netscape 7.x, Mozilla 1.x, Firefox 1.x, CompuServe 7, AOL 8 u Opera 7.11

Linux	
RedHat Enterprise Linux WS v. 3	Mozilla 1.2 y superior, Netscape 7.1
RedHat Linux 9	Mozilla 1.2 y superior, Netscape 7.1
Sun Java Desktop System 1.0	Mozilla 1.2 y superior, Netscape 7.1

### 3.5 Modelo de Dominio

#### 3.5.1 Descripción del modelo de Dominio

Se ha determinado que no es necesario un modelo completo del negocio, por lo que se procede a realizar lo que se conoce como el modelo del dominio. Un modelo del dominio captura los tipos más importantes de objetos que existen o los eventos que suceden en el entorno donde estará el sistema, teniendo como ventaja el permitir ayudar a los usuarios, clientes y desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema.

Se realiza la descripción del modelo de dominio mediante un diagrama de clases UML donde se especifican las principales clases conceptuales que pueden intervenir en el sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema. (Ver Anexo fig 1)

- Se le denominará **usuario** a cualquier persona que interactúe con el sistema.
- Se le denominará **temas** a la estructura básica del contenido de apoyo para el estudio de GNU/Linux.
- Se le denominará **conferencia** al proceso donde los usuarios interactúan en la transmisión de los contenidos.
- Se le denominará **demostraciones** al proceso donde se le explica a los usuarios el cómo hacer determinadas actividades prácticas con el uso de GNU/Linux.

#### 3.5.1 Diagrama de navegación

El diagrama de navegación de la aplicación se puede ver en el Anexo fig 2.

### 3.6 Modelo de casos de uso del sistema

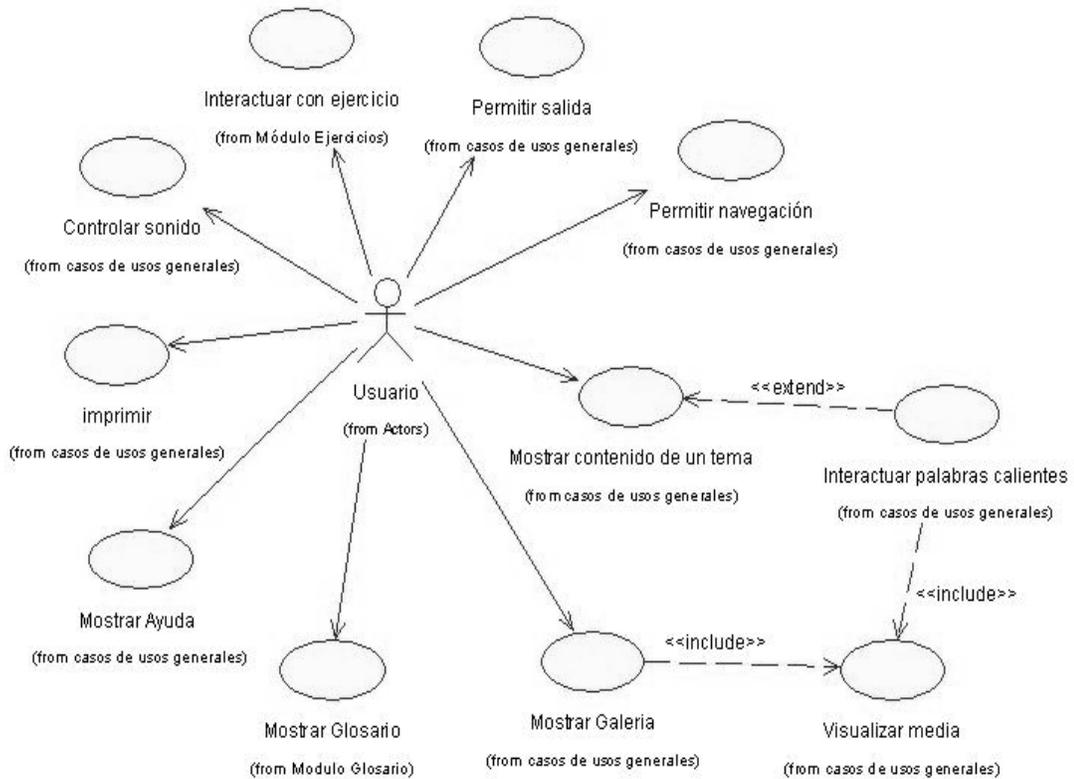
El modelado de casos de uso es la técnica más efectiva para modelar los requisitos del sistema. Los casos de uso se utilizan para modelar el funcionamiento o cómo el cliente desea que funcione el sistema. Utilizando las facilidades que nos brinda el UML, se capturan los requisitos funcionales del sistema y se representan mediante un diagrama de casos de uso. Para ello se definen cuales serían los actores que van a interactuar con el sistema, y los casos de uso que van a representar las funcionalidades del mismo.

#### 3.6.1 Determinación y justificación de los actores del sistema

Actor	Justificación
Usuario	Es la persona que utiliza el sistema para buscar o leer información de un tema determinado.

#### 3.6.2 Descripción y expansión de los casos de uso.

##### 3.6.2.1 Diagrama de CUS.



Ref #	Casos de usos	Prioridad
CUS 1	Mostrar Ayuda del Sistema	Secundario
CUS 2	Controlar Sonido	Secundario
CUS 3	Permitir Salida	Secundario
CUS 4	Permitir Navegación	Crítico
CUS 5	Mostrar Contenido de un tema.	Crítico
CUS 6	Interactuar palabras calientes	Crítico
CUS 7	Visualizar media	Crítico
CUS 8	Mostrar Galería	Crítico
CUS 9	Imprimir	Secundario

<b>CUS 10</b>	Mostrar Glosario	Secundario
<b>CUS 11</b>	Interactuar con ejercicio	Crítico

<b>CUS 1</b>	<b>Mostrar ayuda del sistema</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario solicita la opción de ayuda del sistema. El sistema muestra una pantalla ayuda.	
<b>Propósito</b>	Mostrar información referida a ayuda.	
<b>Referencias</b>	R33	
<b>CU asociado</b>		
<b>Precondiciones</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario desde cualquier pantalla solicita la opción ayuda.	1.1 El sistema se encarga de seleccionar la ayuda correspondiente a la pantalla en que el usuario se encuentra. 1.2 El sistema se encarga de mostrar la ayuda de la información seleccionada.	
<b>Cursos Alternos</b>		
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>		
<b>Prioridad</b>	Secundario	

<b>CUS 2</b>	<b>Controlar sonido</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario solicita la opción de controlar el sonido: activar o desactivar.	
<b>Propósito</b>	Permitir la manipulación del sonido.	
<b>Referencias</b>	R31	
<b>CU asociado</b>		
<b>Precondiciones</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario solicita desactivar el sonido.	1.1 El sistema se encarga de detener el sonido.	
<b>Cursos Alternos</b>		
1. El usuario solicita activar el sonido.	1.1 El sistema activa el sonido.	
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>		
<b>Prioridad</b>	Secundario	

<b>CUS 3</b>	<b>Permitir salida</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario solicita la opción de salir del sistema.	
<b>Propósito</b>	Permitir la salida del sistema.	
<b>Referencias</b>	R32	
<b>CU asociado</b>		
<b>Precondiciones</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario solicita salir del sistema.	El sistema verifica si el usuario desea abandonar realmente la aplicación, mostrando una ventana con las opciones SI y NO.	
2. El usuario selecciona la opción SI.	2.1 El sistema se encarga de finalizar la aplicación.	
<b>Cursos Alternos</b>		
2. El usuario selecciona la opción NO.	2.1 El sistema retorna a la pantalla donde se encontraba el usuario anteriormente.	
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>		
<b>Prioridad</b>	Secundario.	

<b>CUS 4</b>	<b>Permitir Navegación.</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario da un clic sobre el .exe de la aplicación, el sistema muestra la presentación, y al terminar le muestra la pantalla principal, donde a partir de aquí puede moverse de una pantalla a otra según lo desee.	
<b>Propósito</b>	Permitir al usuario navegar por la aplicación de forma global usando los botones del marco y local en las pantallas que tengan navegación interna.	
<b>Referencias</b>	R1, R2, R15, R21, R30, R32, R34.	
<b>CU asociado</b>		
<b>Precondiciones</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario solicita entrar a la aplicación.	1.1 El sistema muestra una presentación inicial, y luego la pantalla principal.	
2. El usuario encontrándose en una determinada pantalla solicita pasar a otra pantalla.	2.1 El sistema a partir de la selección realizada muestra la pantalla correspondiente.	
3. El usuario solicita navegar internamente dentro de una pantalla determinada	3.1 El sistema permite el acceso a los botones que posibilitan la navegación interna entre determinados elementos, dígame elementos de media, como imágenes y videos.	
<b>Cursos Alternos</b>		
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>	Se mostrará la información de la pantalla que solicite el usuario.	
<b>Prioridad</b>	Crítico.	

<b>CUS 5</b>	<b>Mostrar Contenido de un tema.</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario solicita información de la pantalla Temas, luego el sistema se encarga de obtener y mostrar la información solicitada.	
<b>Propósito</b>	Mostrar información del tema seleccionado.	
<b>Referencias</b>	R3, R4, R5.	
<b>CU asociado</b>		
<b>Precondiciones</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario solicita la información de un tema determinado.	1.1 El sistema a partir de la opción seleccionada se encarga de obtener la información. 1.2 El sistema se encarga de mostrar la pantalla con la información obtenida.	
<b>Cursos Alternos</b>		
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>	Se mostrará la información de la pantalla que solicite el usuario.	
<b>Prioridad</b>	Crítico.	

<b>CUS 6</b>	<b>Interactuar palabra caliente &lt;&lt;extend&gt;&gt;</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario selecciona una palabra caliente, el sistema muestra la información correspondiente a la palabra seleccionada.	
<b>Propósito</b>	Mostrar información de palabras calientes.	
<b>Referencias</b>	R6→R11.	
<b>CU asociado</b>	Mostrar Contenido. Visualizar media.	
<b>Precondiciones</b>		
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario selecciona una palabra caliente.	1.1 El sistema obtiene la información de la media que se debe cargar. 1.2 Comienza la ejecución del <b>CUS Visualizar Media</b>	
<b>Cursos Alternos</b>		
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>	.	
<b>Prioridad</b>	Crítico.	

<b>CUS 7</b>	<b>Visualizar media &lt;&lt;include&gt;&gt;</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia sólo si se ha ejecutado el CUS Interactuar con palabra caliente o el CUS Mostrar Galería, en dependencia del resultado de estos CUS, se muestra una imagen o un video.	
<b>Propósito</b>	Mostrar las medias, videos e imágenes existentes en el software.	
<b>Referencias</b>	R8, R12, R13, R14, R16, R17, R18, R20.	
<b>CU asociado</b>	Mostrar Galería Interactuar palabra caliente.	
<b>Precondiciones</b>	La ejecución del CUS Interactuar con palabra caliente o del CUS Mostrar Galería.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
	1 El sistema a partir del resultado del CUS Interactuar con palabra caliente o del CUS Mostrar Galería, determina si visualizar una imagen o un video. En caso de que corresponda visualizar una imagen, ir a <b>Escenario 1</b> , en caso de que corresponda visualizar un video, ir a <b>Escenario 2</b> .	
<b>Escenario 1: Visualizar Imagen</b>		
	1.1 El sistema muestra el visor de imágenes, con la imagen correspondiente. 1.2 El sistema visualiza la descripción de la imagen mostrada.	

<b>Escenario 2: Visualizar Video</b>	
	<p>2.1 El sistema muestra el visor de videos, con el video correspondiente.</p> <p>El sistema muestra la descripción del video mostrado.</p> <p>El sistema muestra opciones de reproducción del video. Estas son:</p> <ul style="list-style-type: none"> <li>- play / pause</li> <li>- stop</li> <li>- barra de progreso.</li> </ul>
<b>Cursos Alternos</b>	
3. El usuario solicita alguna de estas opciones de reproducción.	3.1 El sistema detiene o reproduce el video, en dependencia del estado en que este se encuentre, si se encuentra reproduciendo lo detiene, si se encuentra detenido lo reproduce.
3. El usuario selecciona la opción stop.	3.1 El sistema detiene el video y lo lleva al inicio.
3. El usuario selecciona la opción barra de progreso.	3.1 El sistema desplaza la reproducción del video hasta donde el usuario lo haya solicitado mediante la barra de desplazamiento.
<b>Requisitos no Funcionales</b>	
<b>Poscondiciones</b>	
<b>Prioridad</b>	Crítico.

<b>CUS 8</b>	<b>Mostrar Galería</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario solicita ver algunas de las galerías que brinda el sistema, ya sea la de videos o la de imágenes. Luego, el sistema se encarga de visualizar cada una de estas correspondientemente.	
<b>Propósito</b>	Mostrar de forma organizada todas las imágenes y videos que contiene la aplicación.	
<b>Referencias</b>	R12, R13, R16 y R17.	
<b>CU asociado</b>	Visualizar Media	
<b>Precondiciones</b>	El usuario debe solicitar ir a pantalla Galería Imágenes o Galería Videos.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario selecciona visualizar pantalla Galería Imágenes o Galería Videos.  2. El usuario acciona cualquiera de las imágenes o videos que desee visualizar.	1.1 El sistema obtiene todas las imágenes o videos que existen en la aplicación.  1.2 Muestra la vista diminuta de las imágenes o videos existentes de forma organizada.  2.1 Comienza la ejecución del <b>CUS Visualizar Media</b>	
<b>Cursos Alternos</b>		
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>		
<b>Prioridad</b>	Crítico.	

<b>CUS 9</b>	<b>Imprimir</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario solicita la opción de imprimir. El sistema permite la impresión del texto correspondiente.	
<b>Propósito</b>	Permitir que se imprima el texto del tema que se encuentra visualizando la aplicación en ese momento.	
<b>Referencias</b>	R35	
<b>CU asociado</b>		
<b>Precondiciones</b>	El usuario para poder proceder a esta acción debe estar en una pantalla de contenido o en la pantalla Glosario de Términos.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario solicita imprimir el contenido de una pantalla.	1.1 El sistema muestra la pantalla de propiedades de impresión del sistema permitiendo la impresión del contenido deseado.	
<b>Cursos Alternos</b>		
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>		
<b>Prioridad</b>	Secundario	

<b>CUS 10</b>	<b>Mostrar Glosario.</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario desea buscar el significado de una palabra en la pantalla "Glosario". El sistema muestra las palabras con su significado.	
<b>Propósito</b>	Mostrar el significado de la palabra que el usuario desea.	
<b>Referencias</b>	R22, R23.	
<b>CU asociado</b>		
<b>Precondiciones</b>	El usuario para poder proceder a esta acción debe estar en la pantalla de Glosario.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
1. El usuario selecciona un botón que muestra una letra del abecedario.	1.1 El sistema se encarga de obtener todas las palabras que comienzan con la letra seleccionada.  1.2 El sistema se encarga de mostrar las palabras que comiencen con esa letra, con su significado correspondiente y ordenadas alfabéticamente en orden ascendente.	
<b>Cursos Alternos</b>		
<b>Requisitos no Funcionales</b>		
<b>Poscondiciones</b>		
<b>Prioridad</b>	Secundario.	

<b>CUS 11</b>	<b>Interactuar con ejercicio.</b>	
<b>Actores</b>	Usuario	
<b>Resumen</b>	El caso de uso se inicia cuando el usuario solicita la opción de realizar ejercicio.	
<b>Propósito</b>	Permitir la realización de ejercicios.	
<b>Referencias</b>	R24→R28.	
<b>CU asociado</b>		
<b>Precondiciones</b>	El usuario debe encontrarse en la pantalla Ejercicios.	
<b>Flujo Normal de Eventos</b>		
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>	
<p>1. El usuario escoge la opción realizar algún tipo de ejercicios.</p> <p>2. El usuario tiene la opción de responder o no el ejercicio.</p> <p>3. En caso de que el usuario solicite ver respuesta correcta del ejercicio, ir a <b>escenario 1</b>. En caso de que seleccione la opción realizar más ejercicios, ir a <b>escenario 2</b>.</p>	<p>1.1 El sistema obtiene la información para mostrar los ejercicios.</p> <p>1.2 Muestra los ejercicios del tipo seleccionado.</p> <p>2.1 En cualquier caso el sistema muestra la opción de ver la respuesta correcta del ejercicio, y de realizar otros ejercicios del mismo tipo.</p>	

<b>Escenario 1:</b> Mostrar respuesta correcta del ejercicio.	
	<p>3.1. El sistema obtiene la información de la respuesta correcta del ejercicio actual.</p> <p>3.2. El sistema muestra la respuesta correcta del ejercicio actual.</p>
<b>Escenario 2:</b> Mostrar más ejercicios	
	3.1 Si aun quedan más ejercicios sin resolver, volver al evento <b>1.1</b> . En caso contrario expresa en un mensaje que se han agotado todos los ejercicios de este tipo.
<b>Requisitos no Funcionales</b>	
<b>Poscondiciones</b>	
<b>Prioridad</b>	Crítico

### 3.7 Conclusiones

El desarrollo de este capítulo ha permitido una mejor comprensión de la aplicación a automatizar, teniendo en cuenta que se realizó un modelo de dominio donde se analizaron los conceptos más importantes del entorno en que se utilizará la aplicación, realizando de esta manera la descripción de la solución propuesta y definiendo los requerimientos funcionales y no funcionales que debe cumplir el sistema. Además se muestra el diagrama de navegación que tendrá la aplicación informática. Mediante los diagramas de casos de uso del sistema se representó la acción del actor sobre los casos de uso, además se realizó la descripción de cada caso uno de estos en un formato expandido, en los cuales se evidencia como será la interacción del usuario con el sistema. Gracias a la culminación de este capítulo, se puede empezar a construir el sistema con tecnología multimedia, tratando de que se cumplan todos los requerimientos y las funciones que se han considerado necesarias.

## CAPÍTULO 4

### Construcción de la solución propuesta

#### 4.1 Introducción

El Proceso Unificado Racional o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. (WIKIPEDIA)

En capítulos anteriores se hace referencia a estos conceptos pero es válido hacerle mención ya que la fase de análisis y diseño empleando los artefactos de UML y expandidos con OMMMA-L necesitan de estos.

UML plantea una vista estática y otra dinámica de los objetos que interactúan en el desarrollo de un sistema. Para la vista estática se utilizan los diagramas de clases y para la dinámica los de interacción (secuencia).

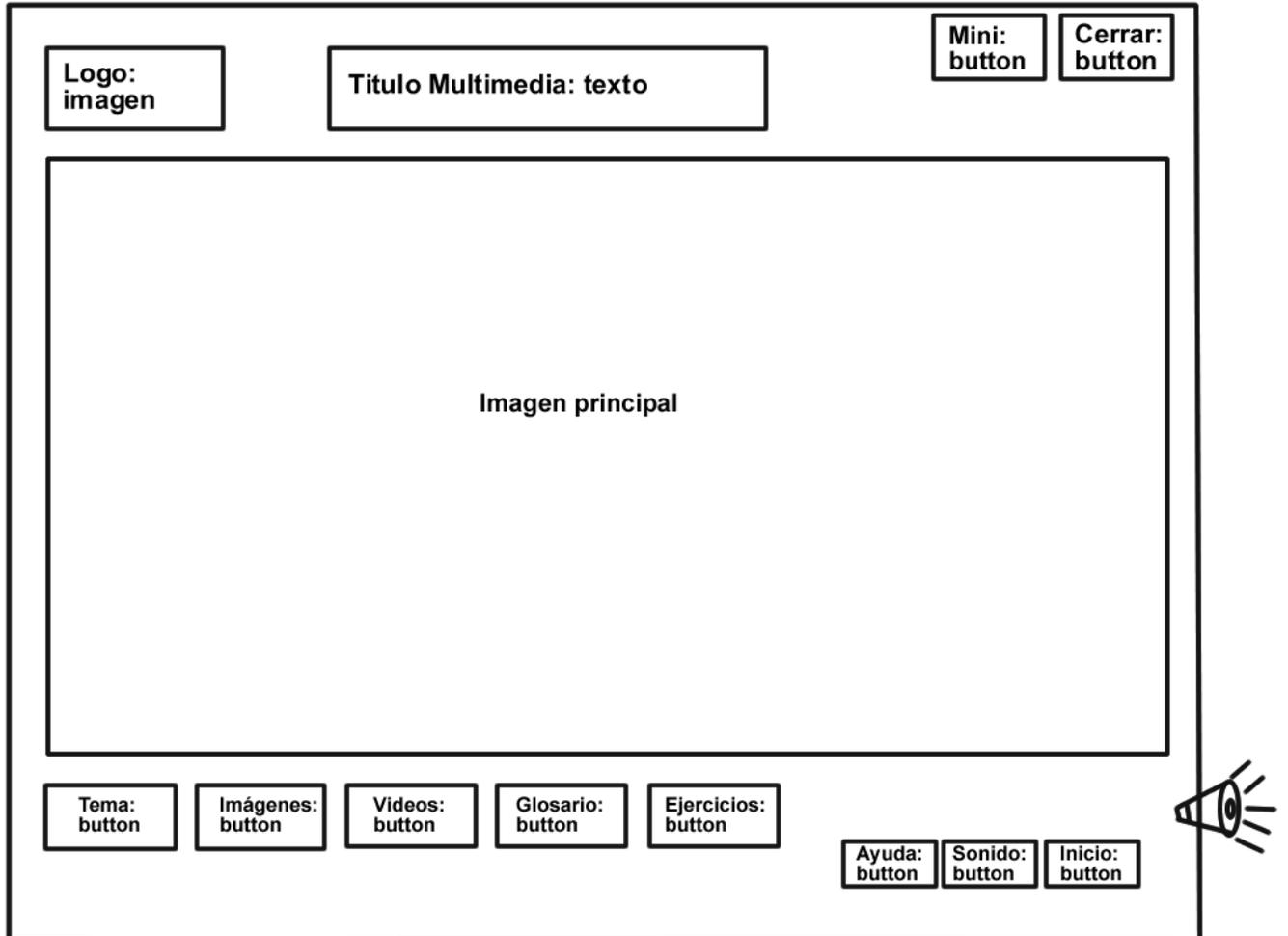
OMMMA-L para una mejor comprensión utiliza los diagramas de presentación y modifica los diagramas de clases, este último se divide en dos áreas: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.

Una vez obtenido el diagrama de presentación para cada entidad escenario y aplicación, OMMMA – L identifica la obtención de los elementos de media principales. Se define la estructura de paquete y los diagramas de clases y objetos específicos. En la primera se sienta la estructura de la organización de los escenarios y aplicaciones y en la segunda la correspondencia de las medias.

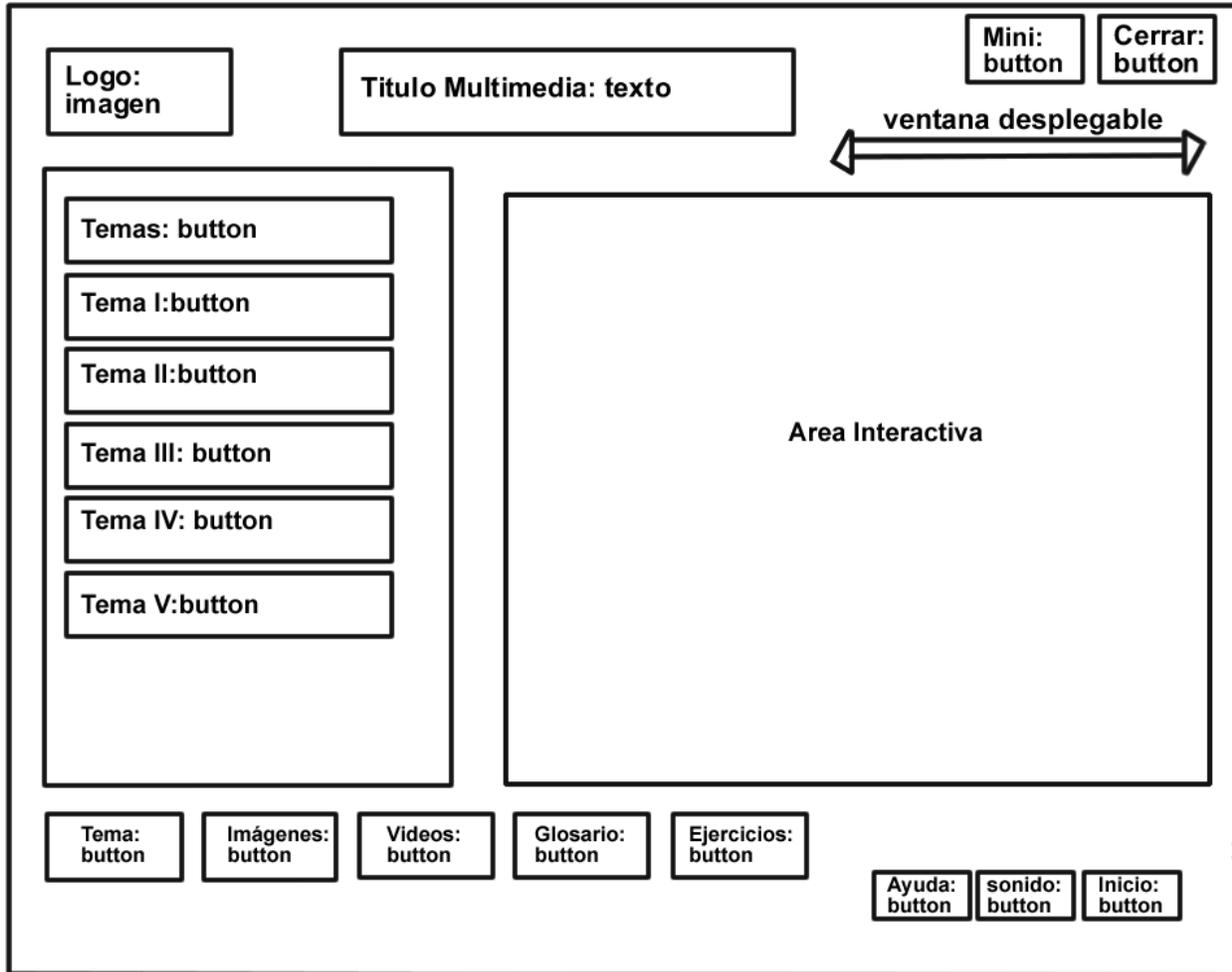
Resumiendo, previamente se presenta el modelo de análisis y diseño y se muestra también el modelo de implementación con los diagramas de componentes definidos.

#### 4.2 Diagramas de presentación del modelo de análisis.

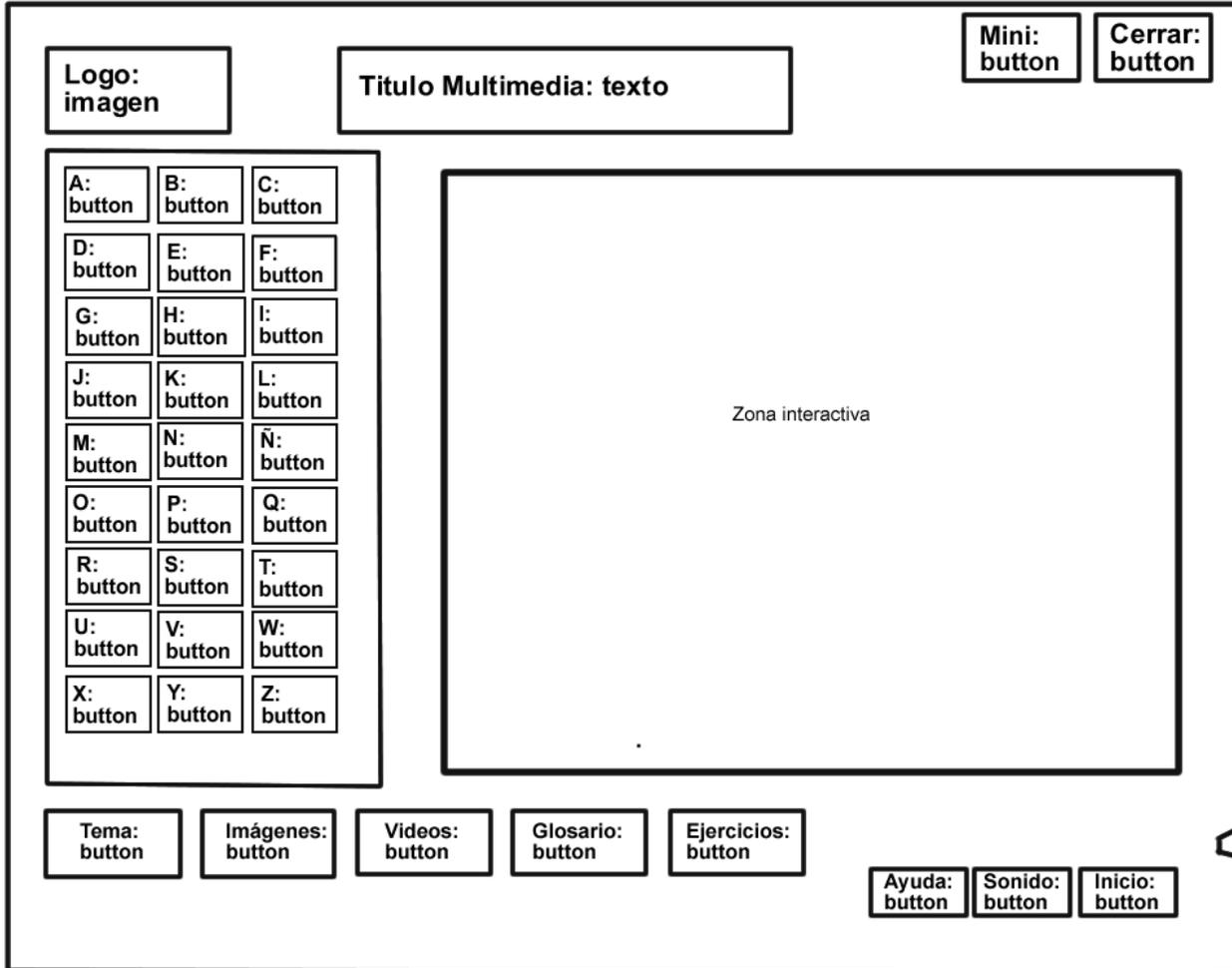
#### 4.2.1 Diagrama de presentación Pantalla Principal



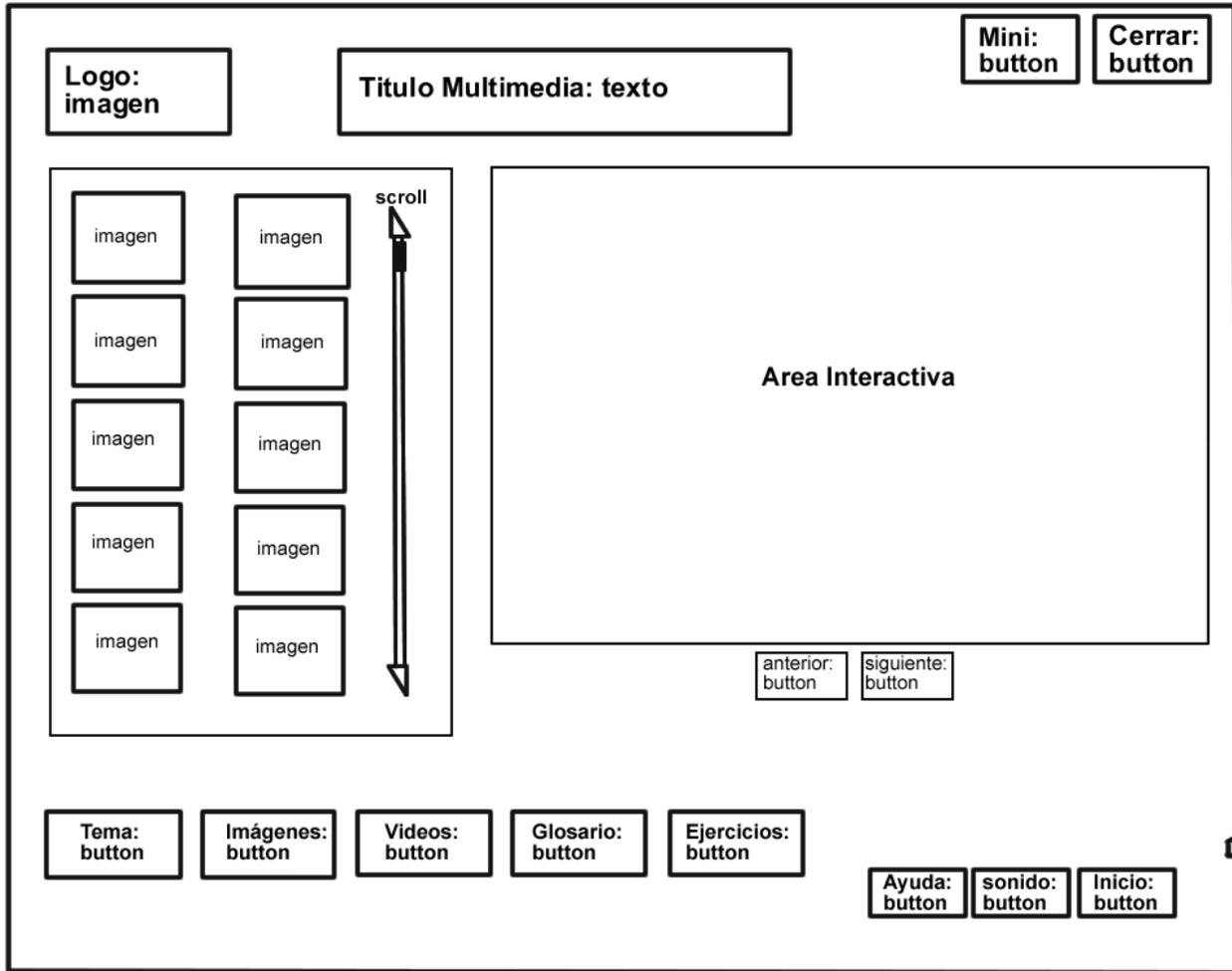
#### 4.2.2 Diagrama de presentación Módulo Temas



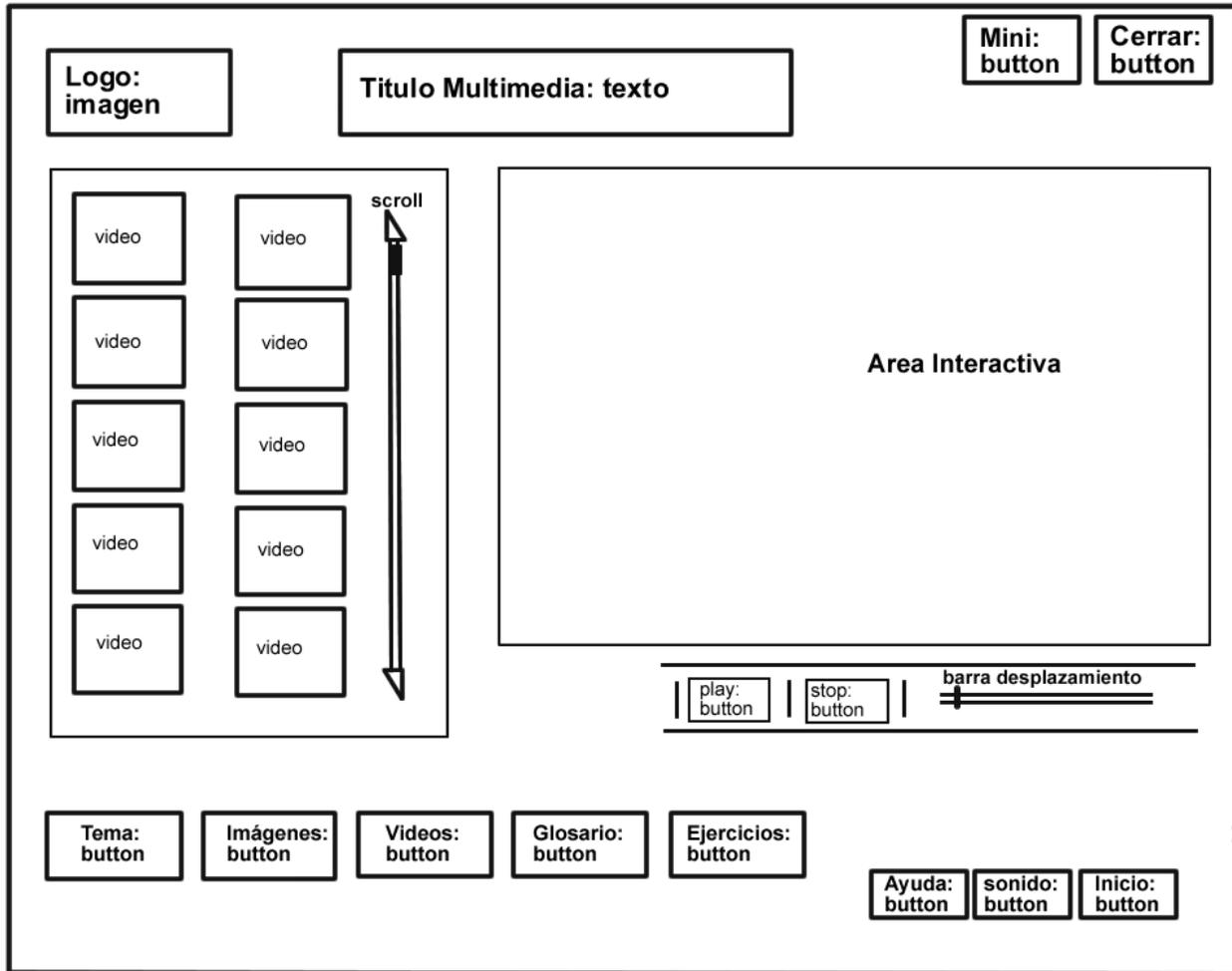
### 4.2.3 Diagrama de presentación del Módulo Glosario



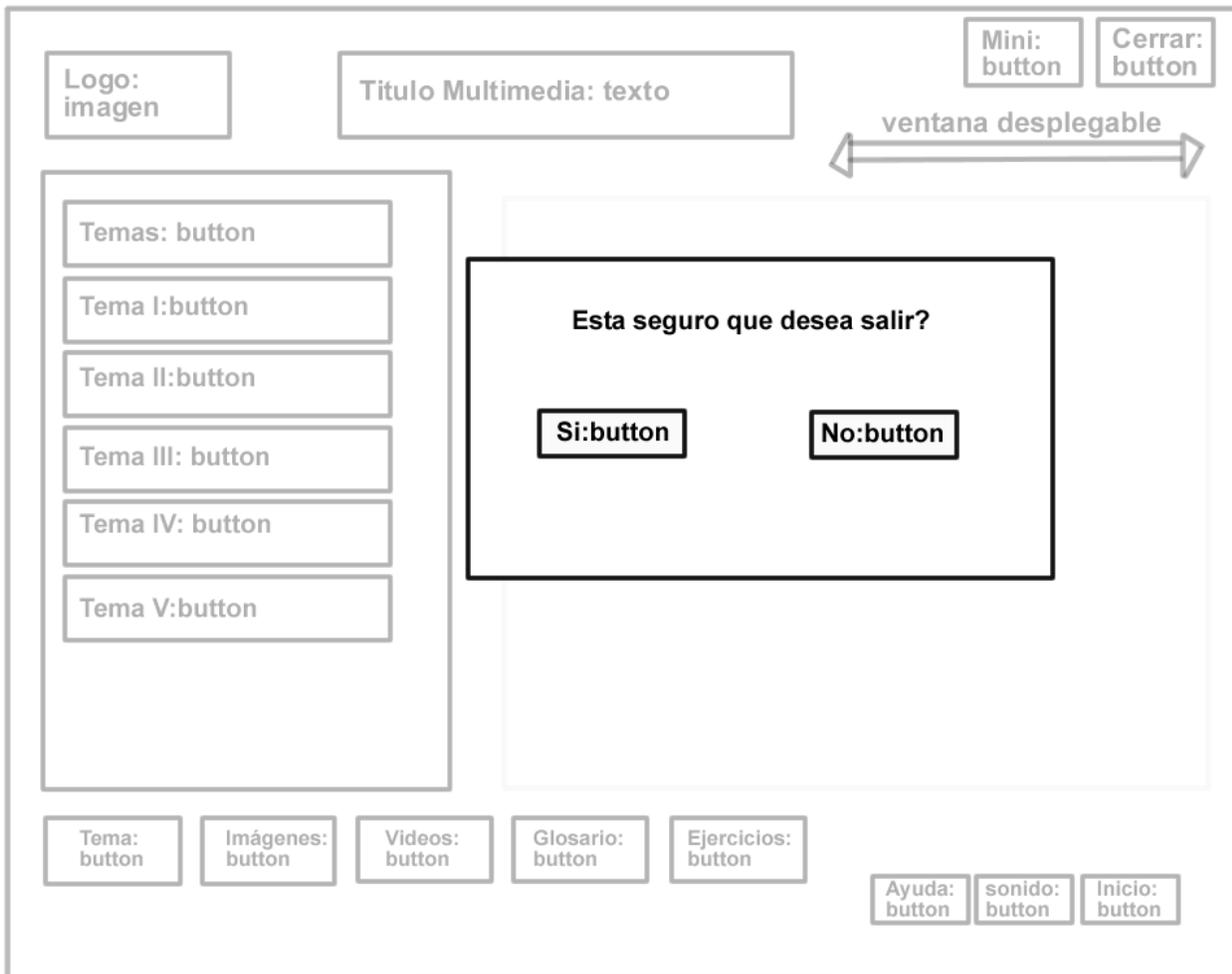
#### 4.2.4 Diagrama de presentación del Módulo Galería de Imágenes



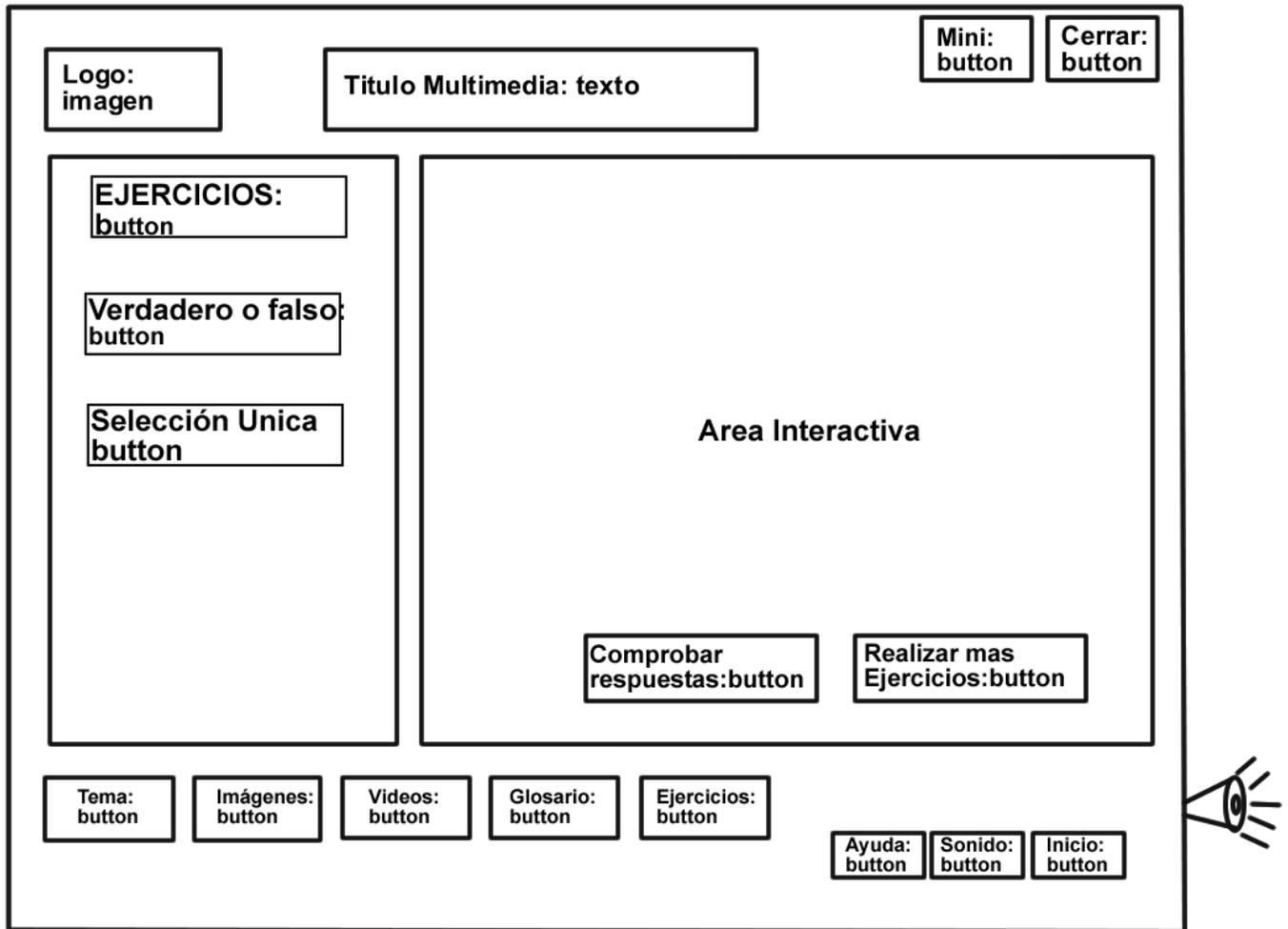
### 4.2.5 Diagrama de presentación Módulo Videos



#### 4.2.6 Diagrama de presentación Pantalla Salir



#### 4.2.7 Diagrama de presentación Pantalla Ejercicios



#### 4.3 Diagrama de clases de diseño.

Una vez obtenido el diagrama de presentación para cada entidad escenario y aplicación, OMMMA – L identifica la obtención de los elementos de media principales. Se definen los diagramas de clases y objetos específicos. En el primero se sienta la estructura de la organización de los escenarios y aplicaciones y en el segundo la correspondencia de las medias.

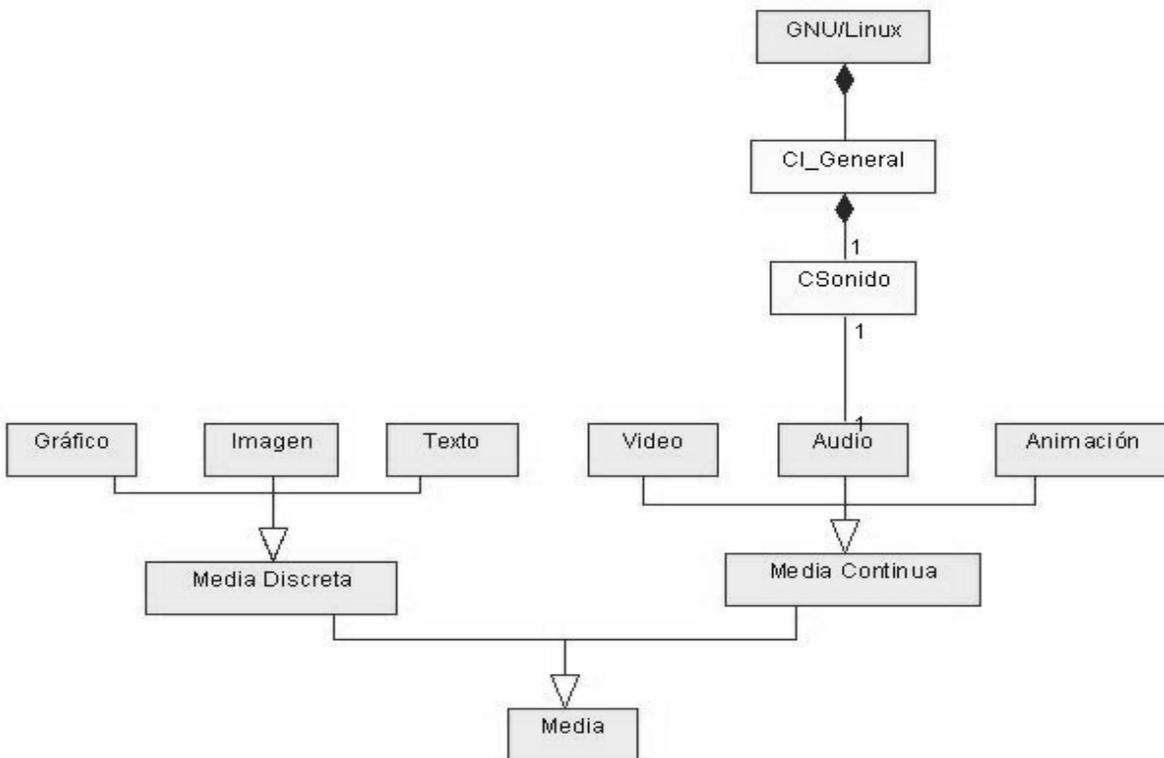
El *Diagrama de Clases de OMMMA-L*, extendido del *Diagrama de Clases de UML*, utiliza las mismas notaciones que este último, pero incorporando las clases correspondientes a las medias; en este caso las

*medias continuas* y las *medias discretas*, generalizadas en una clase *medias*. Al mismo tiempo este diagrama se divide en dos áreas: una para la jerarquía de los *tipos de media* y otra para la *modelación de la estructura lógica* del dominio de la aplicación.

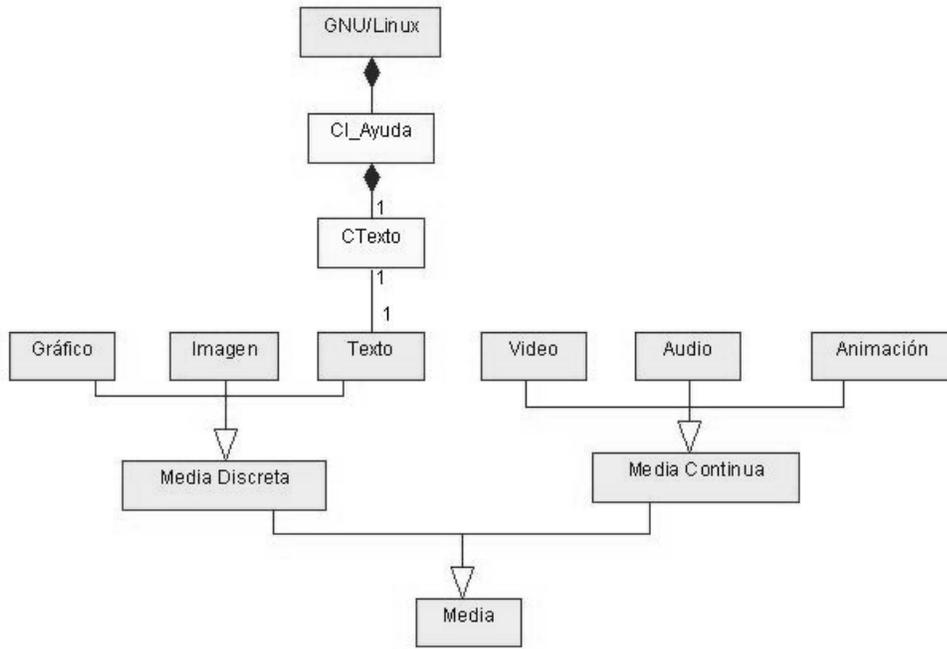
Los diagramas de clases son los más utilizados en el modelado de sistemas orientados a objetos. Un diagrama de clases muestra un conjunto de clases, interfaces y colaboraciones, así como sus relaciones. Los diagramas de clases se utilizan para modelar el vocabulario del sistema, modelar las colaboraciones o modelar esquemas.

### 4.3.1 Mapeo de clases y jerarquía de Medias.

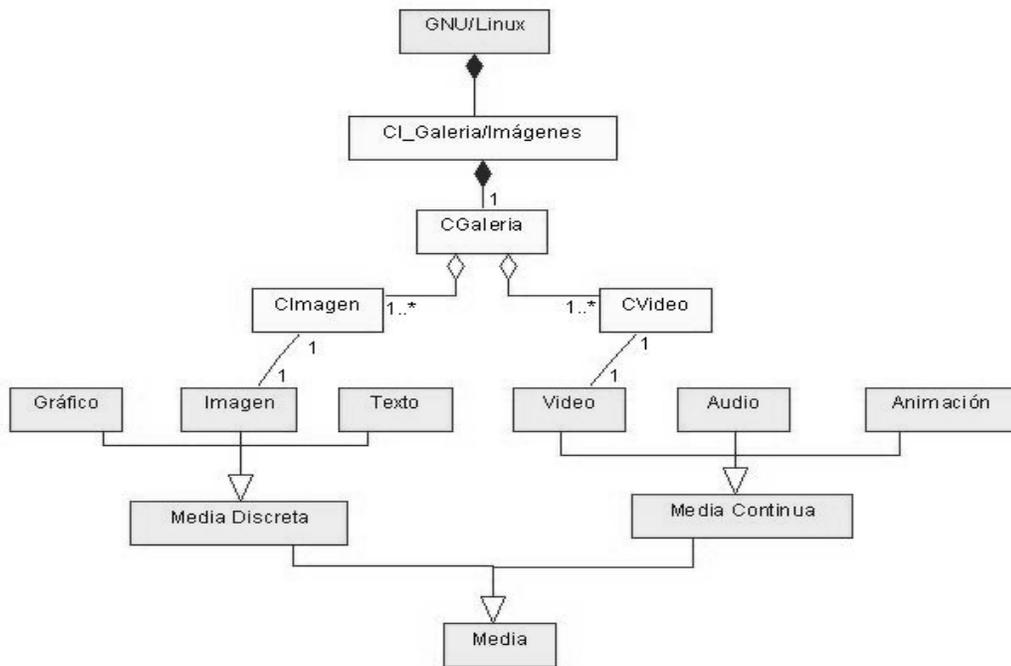
#### Mapeo de clases general.



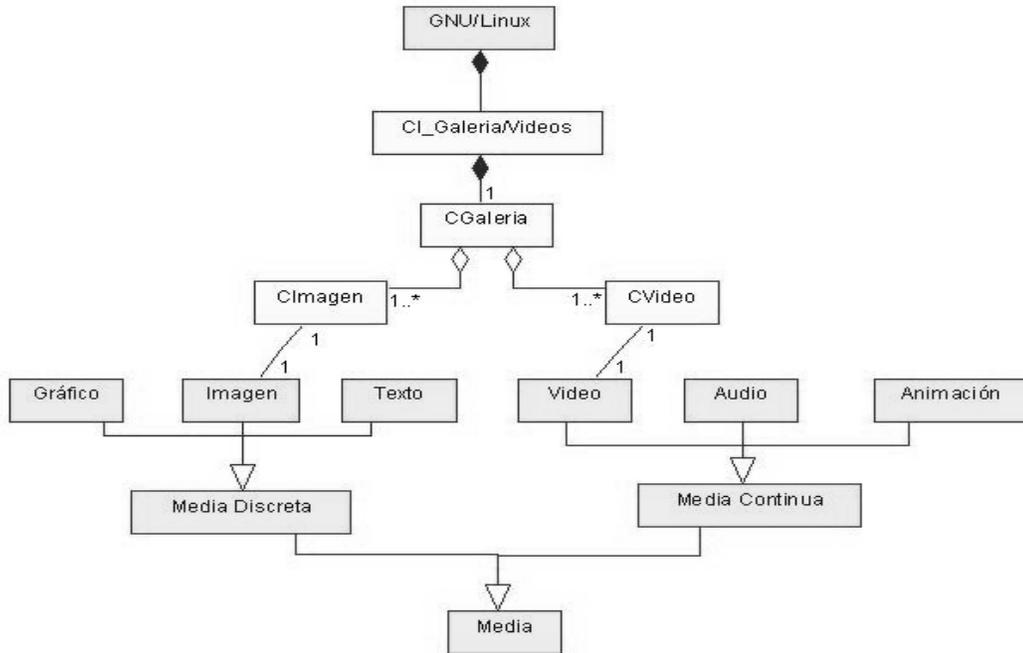
**Mapeo de clases del escenario ayuda**



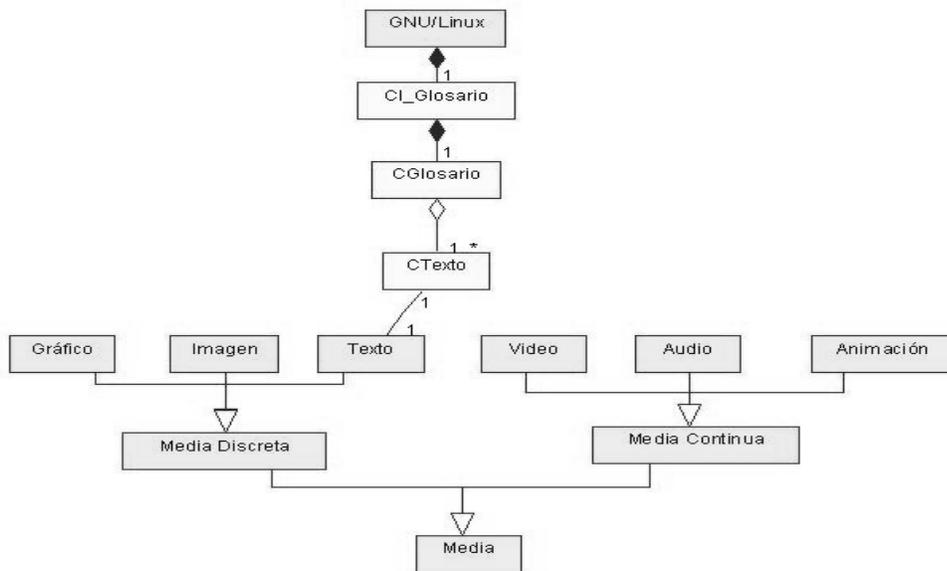
**Mapeo de clases del escenario Galería de imágenes**



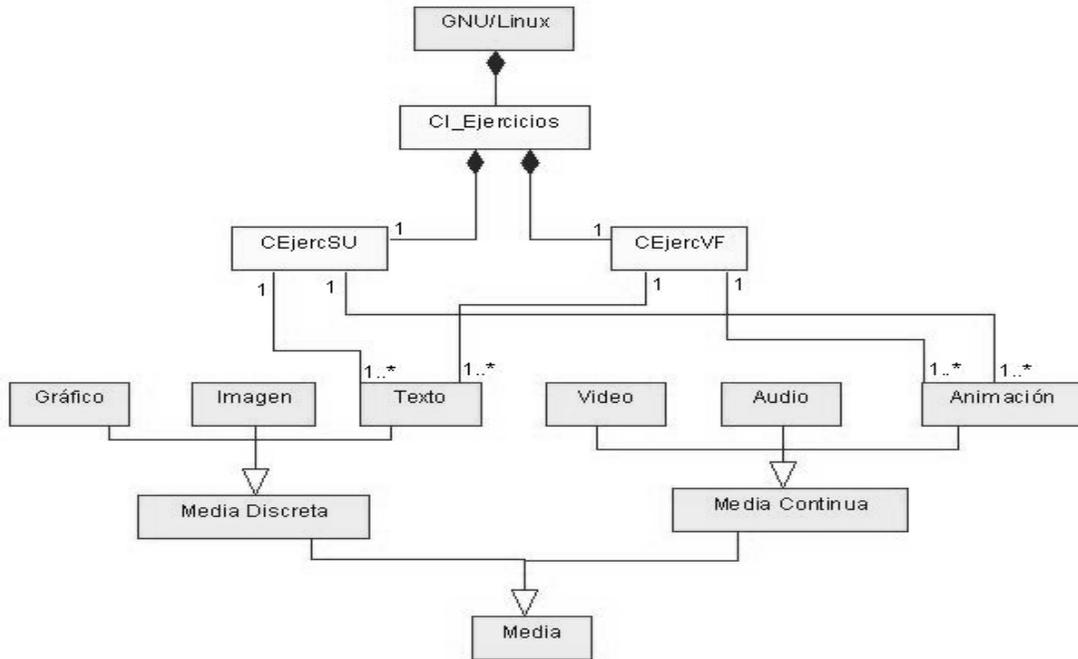
**Mapeo de clases del escenario Galería Videos**



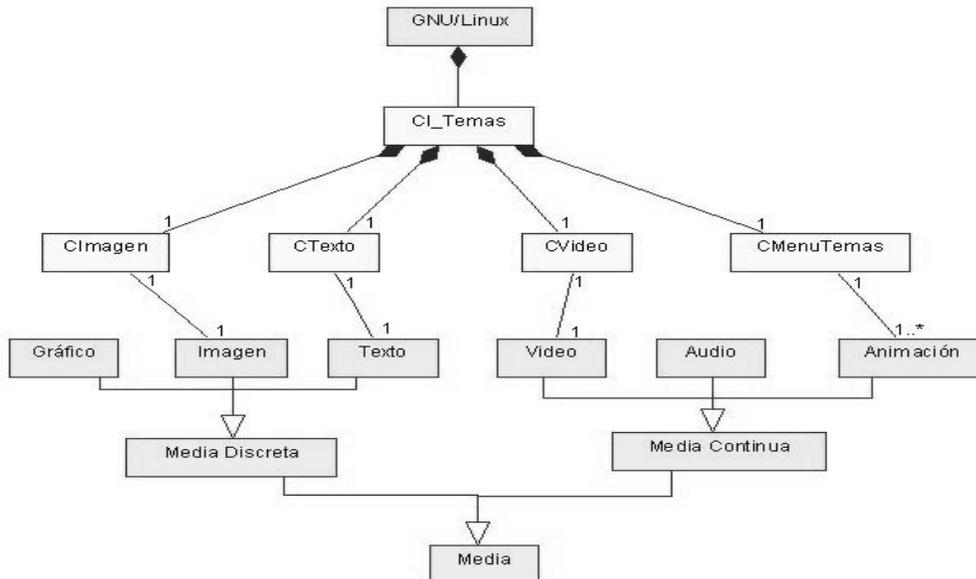
**Mapeo de clases del escenario Glosario**



**Mapeo de clases del escenario Ejercicios**



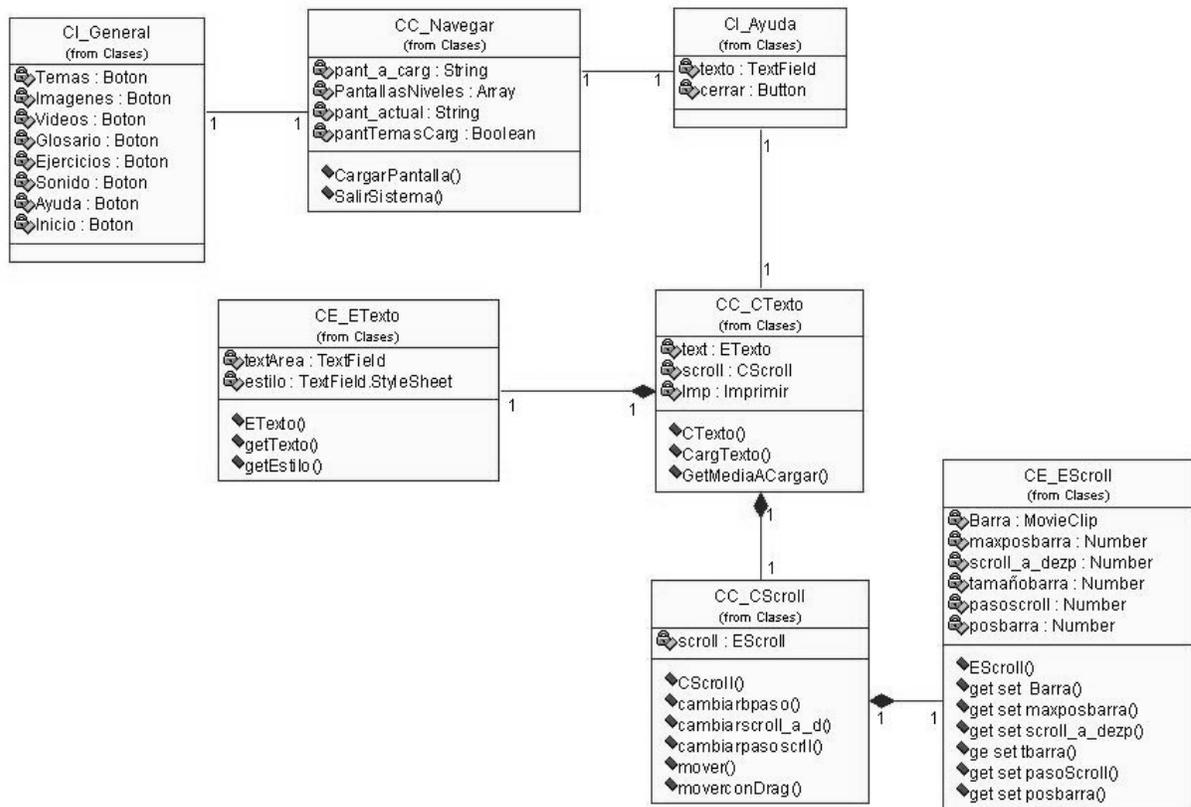
**Mapeo de clases del escenario Temas**



### 4.3.2 Diagrama de clases general

El diagrama de clases general, se puede ver en el Anexo 1, Figura 3.

### 4.3.3 Diagrama de clase de diseño CUS Mostrar Ayuda del Sistema



Diagramas de secuencia de los casos de uso relacionados con Mostrar Ayuda.

Ver Anexo1 figura 4.

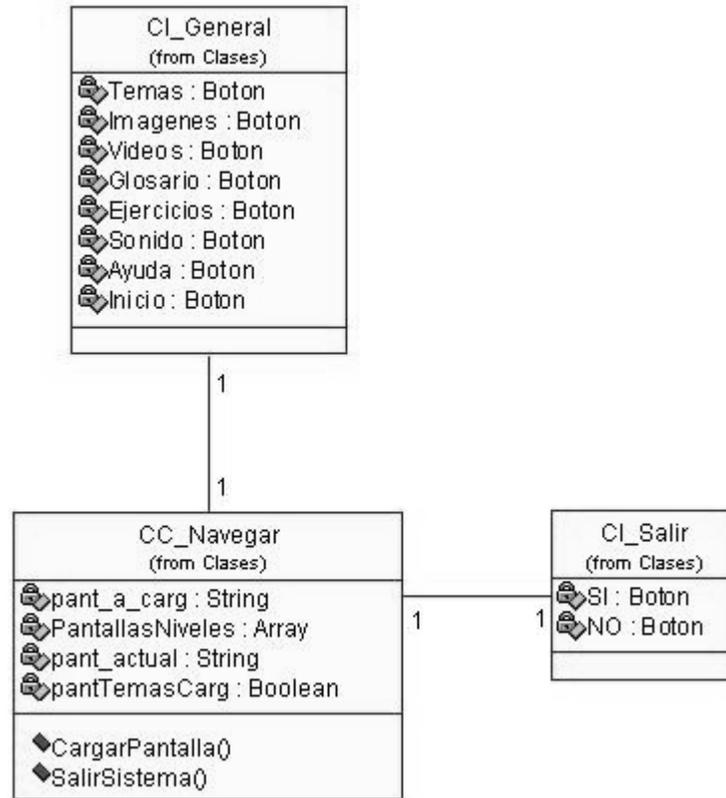
4.3.4 Diagrama de clase de diseño CUS Controlar Sonido



Diagramas de secuencia de los casos de uso relacionados con Controlar Sonido

Ver Anexo1 figura 5.

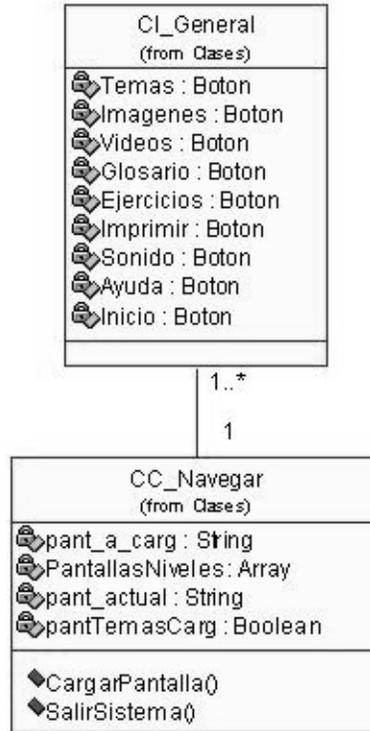
4.3.5 Diagrama de clase de diseño CUS Permitir Salida



Diagramas de secuencia de los casos de uso relacionados con Permitir Salida

**Ver Anexo1 figura 6.**

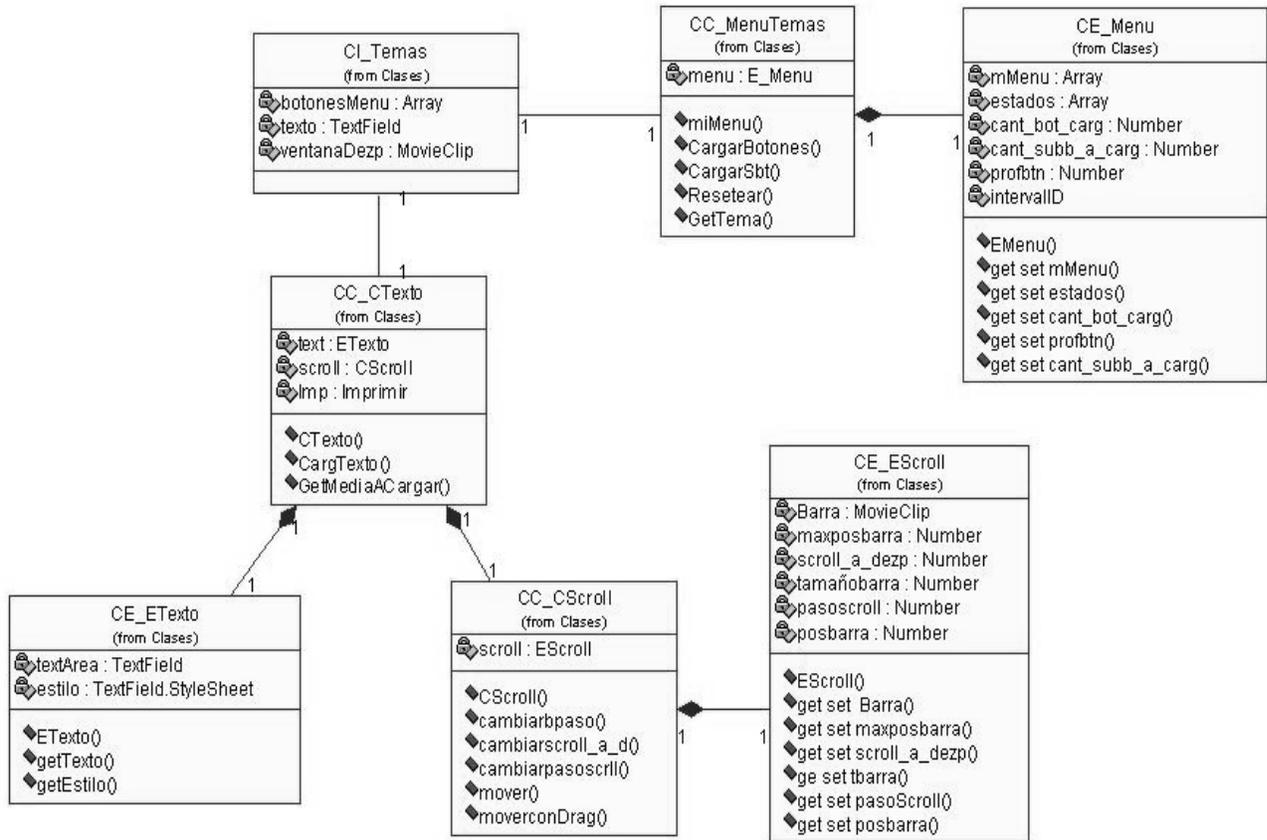
#### 4.3.6 Diagrama de clase de diseño CUS Permitir Navegación



Diagramas de secuencia de los casos de uso relacionados con Permitir Navegación

**Ver Anexo1 figura 7.**

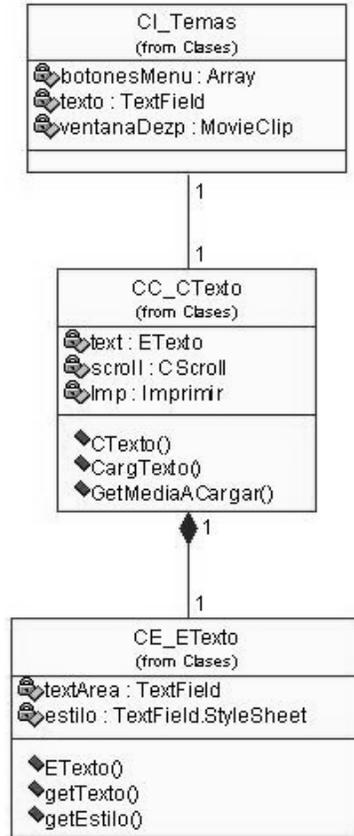
4.3.7 Diagrama de clase de diseño CUS Mostrar Contenido de un tema.



Diagramas de secuencia de los casos de uso relacionados con Mostrar Contenido de un tema.

Ver Anexo1 figura 8.

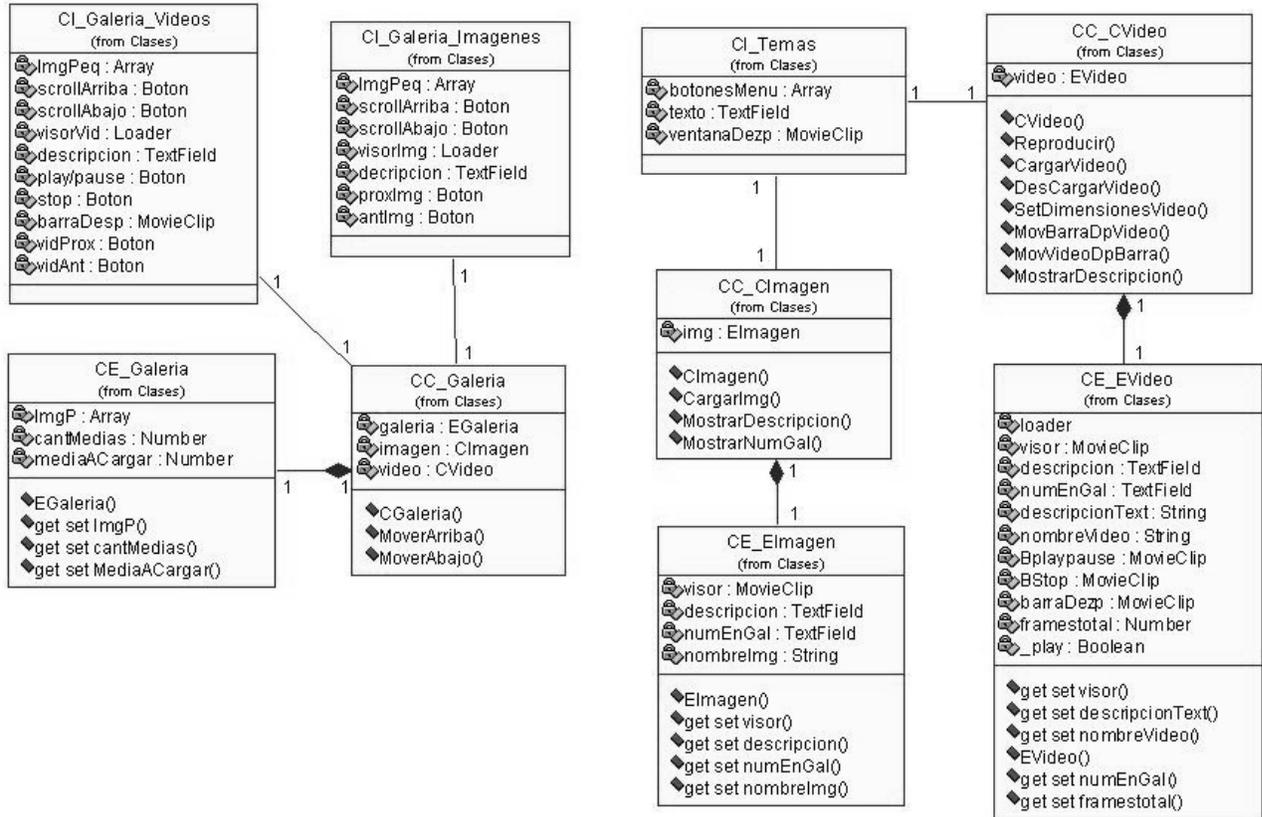
4.3.8 Diagrama de clase de diseño CUS Interactuar con palabras calientes.



Diagramas de secuencia de los casos de uso relacionados con Interactuar con palabras calientes.

Ver Anexo1 figura 9 y figura 9a.

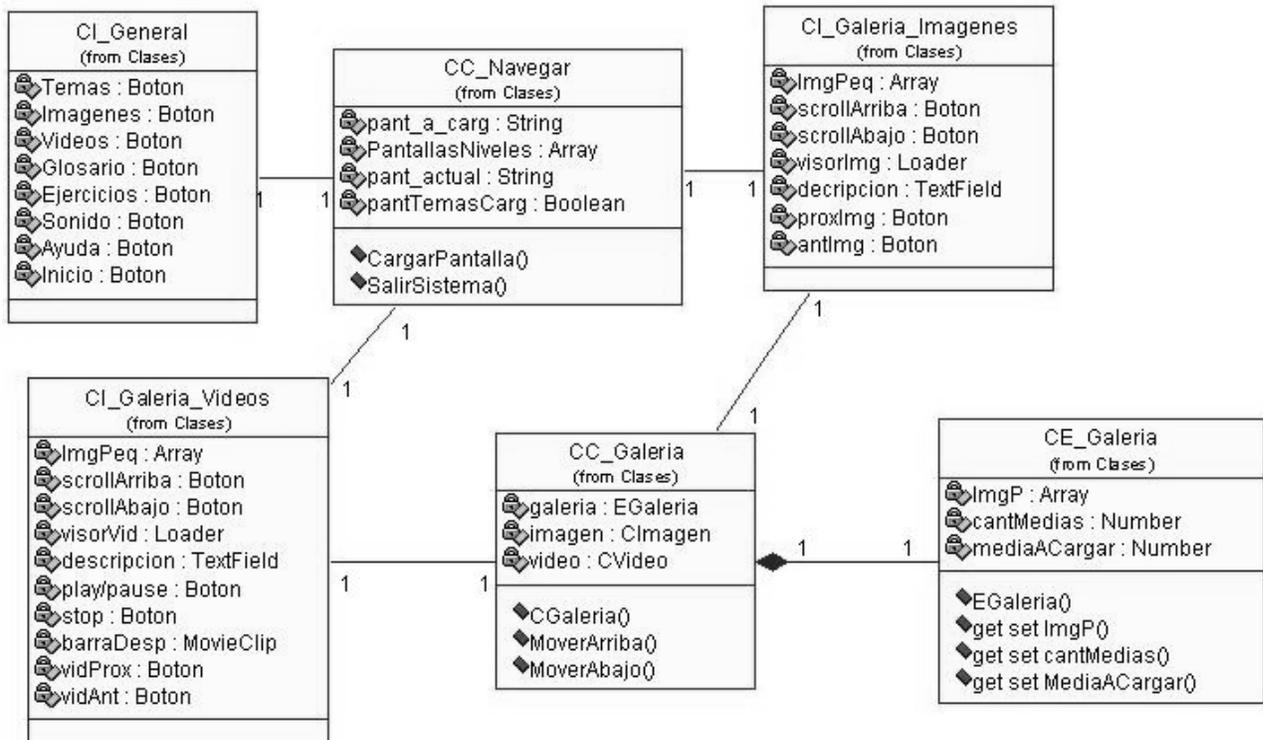
4.3.9 Diagrama de clase de diseño CUS Visualizar media.



Diagramas de secuencia de los casos de uso relacionados con Visualizar Media.

Ver Anexo1 figura 10 y figura 10a.

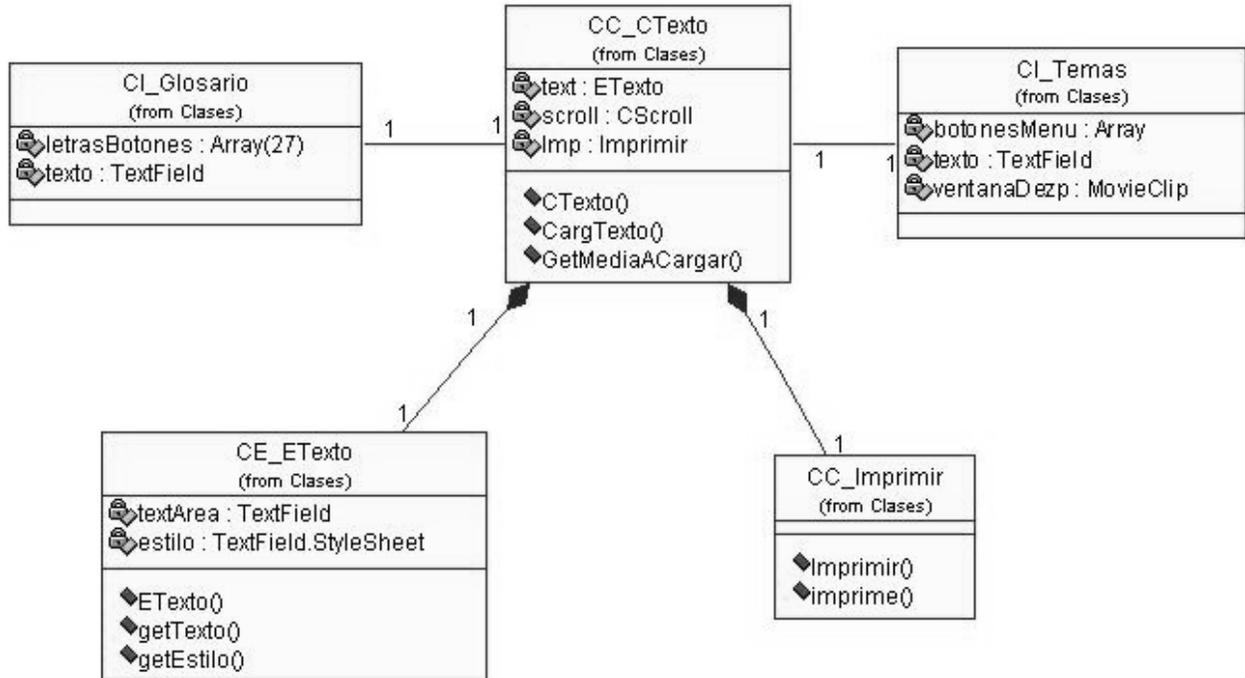
4.3.10 Diagrama de clase de diseño CUS Mostrar Galería.



Diagramas de secuencia de los casos de uso relacionados con Mostrar Galería.

Ver Anexo1 figura 10 y figura 10a.

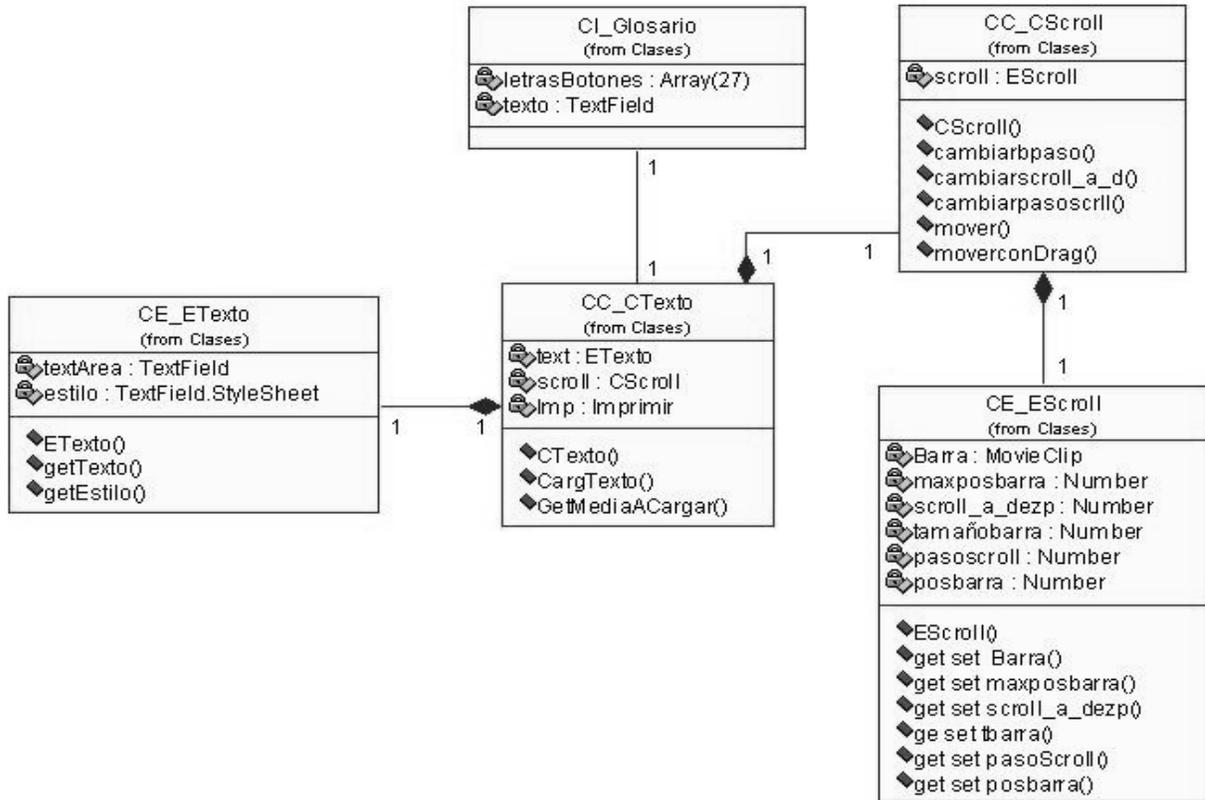
4.3.11 Diagrama de clase de diseño CUS Imprimir.



Diagramas de secuencia de los casos de uso relacionados con Imprimir.

Ver Anexo1 figura 11.

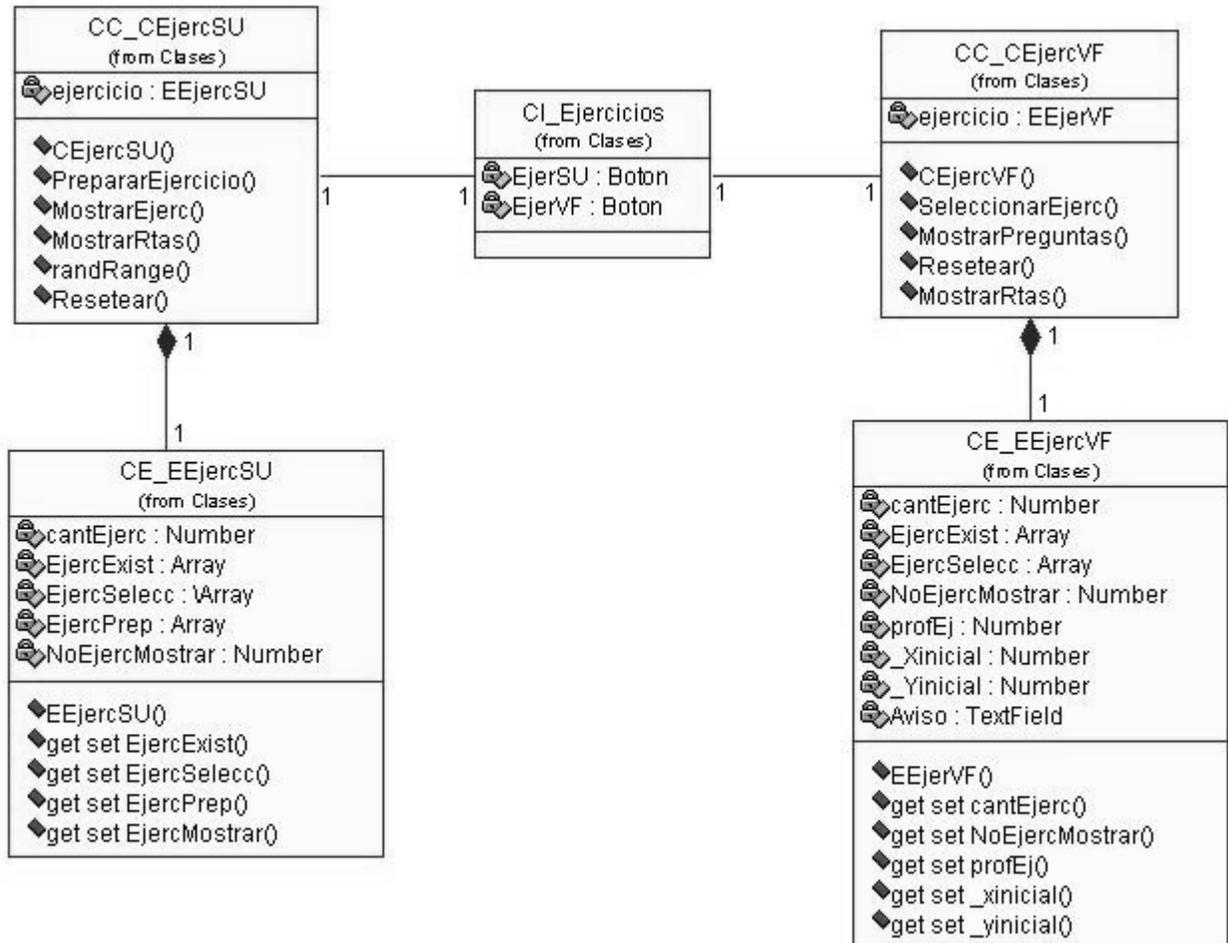
4.3.12 Diagrama de clase de diseño CUS Mostrar Glosario.



Diagramas de secuencia de los casos de uso relacionados con Mostrar Glosario.

Ver Anexo1 figura 12.

**4.3.13 Diagrama de clase de diseño CUS Interactuar con ejercicios.**



Diagramas de secuencia de los casos de uso relacionados con Interactuar con ejercicio.

**Ver Anexo1 figura 13 y figura 13 a.**

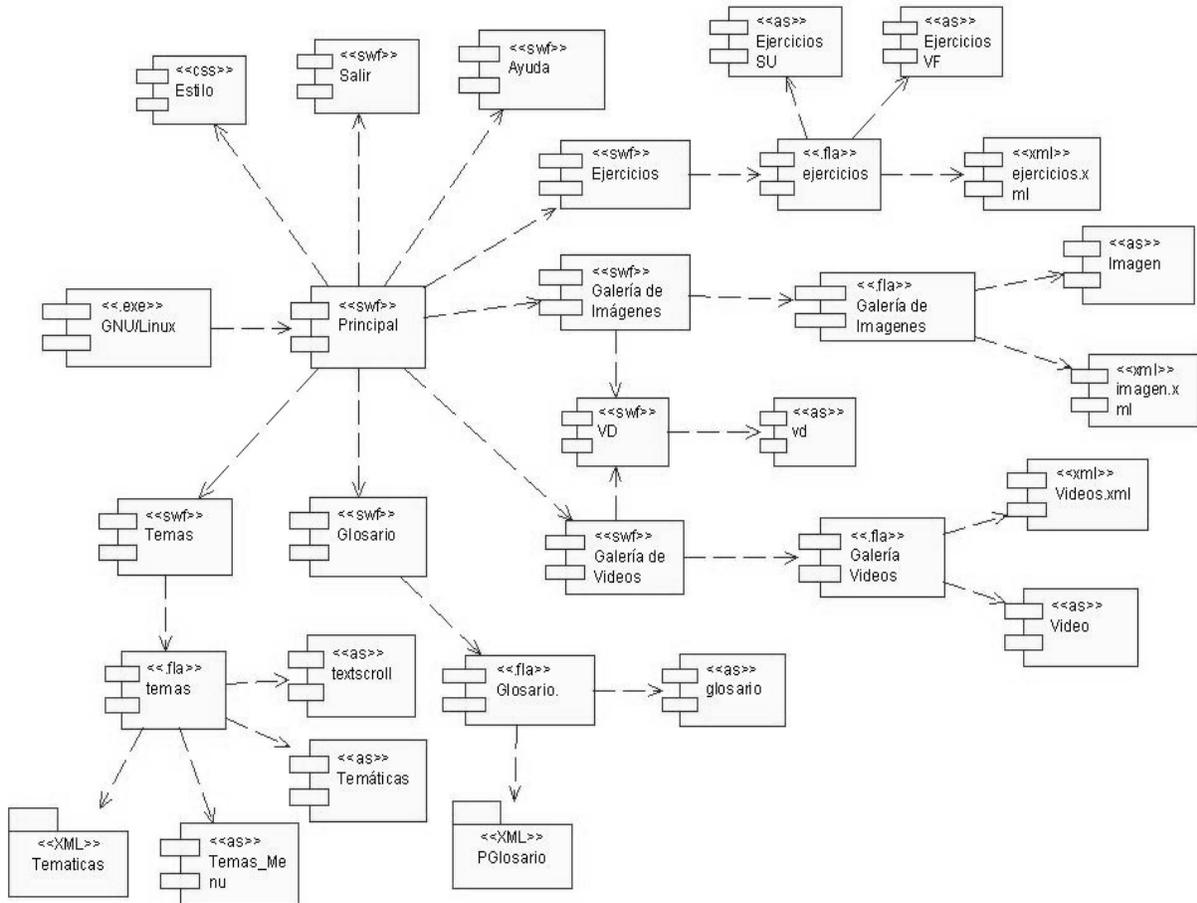
**4.4 Modelo de implementación.**

La vista de implementación modela los componentes de un sistema a partir de los cuales se construye la aplicación así como las dependencias entre los componentes, para poder determinar el impacto de un cambio propuesto. También modela la asignación de clases y de otros elementos del modelo a los componentes.

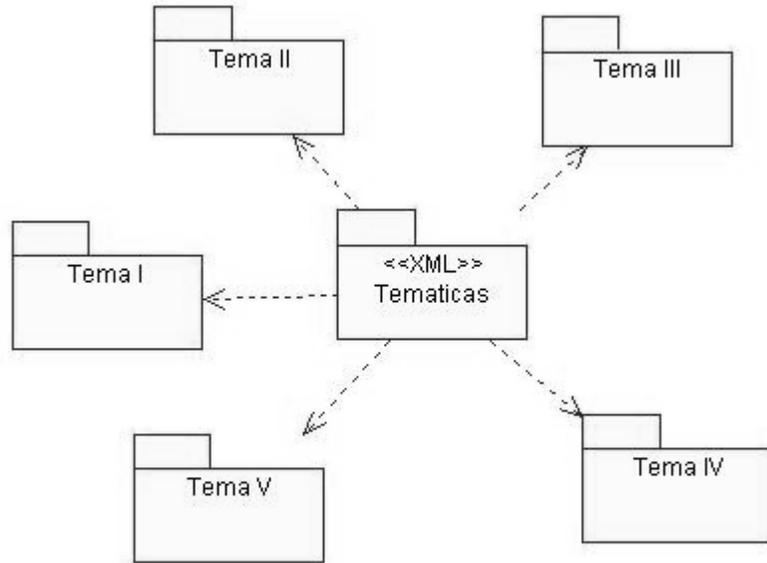
El modelo de implementación describe como se organizan y se relacionan los elementos del modelo del diseño, así como se implementan en términos de componentes, definiendo un componente como el empaquetamiento físico de los elementos de un modelo, como es el caso de las clases del modelo de diseño.

Un Diagrama de Componentes representa la separación de un sistema de software en componentes físicos (por ejemplo archivos, cabeceras, módulos, paquetes). El Diagrama de Componentes se usa para modelar la estructura del software, incluyendo las dependencias entre los componentes de software, los componentes de código binario, y los componentes ejecutables.

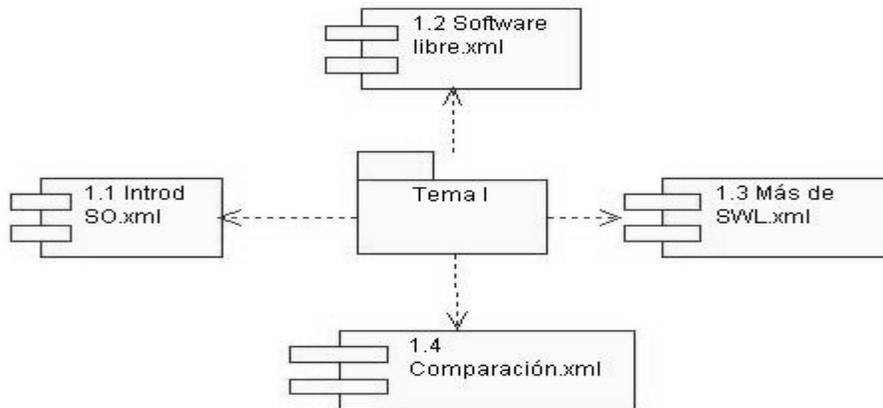
### 4.4.1 Diagrama de componentes



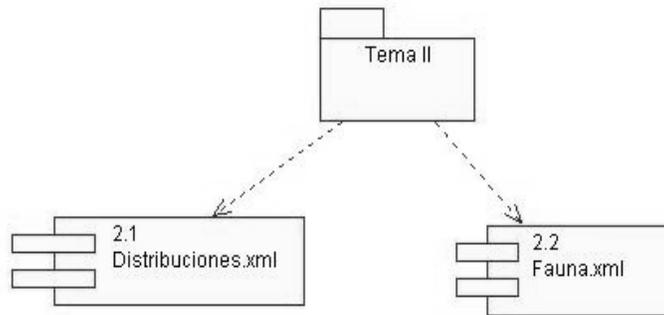
#### 4.4.2 Diagrama de componentes del paquete Temáticas



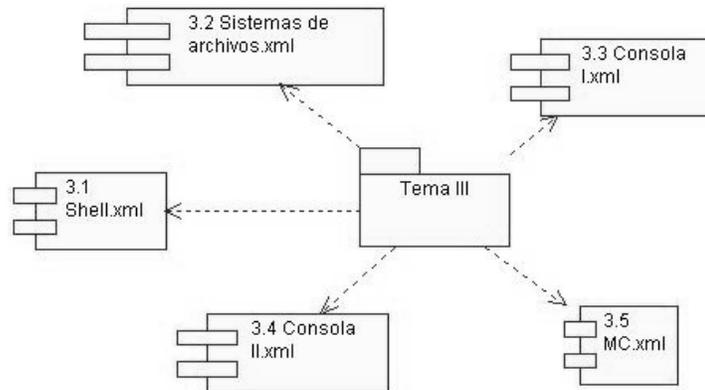
#### 4.4.2.1 Paquete Tema I



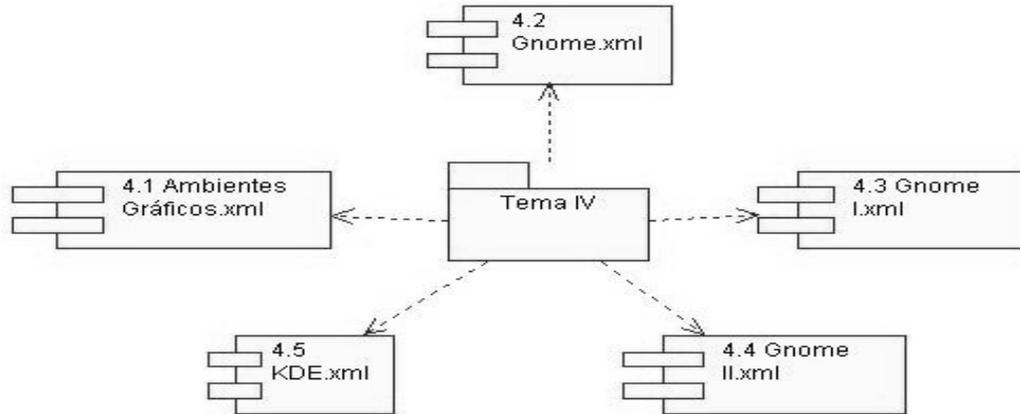
#### 4.4.2.2 Paquete Tema II



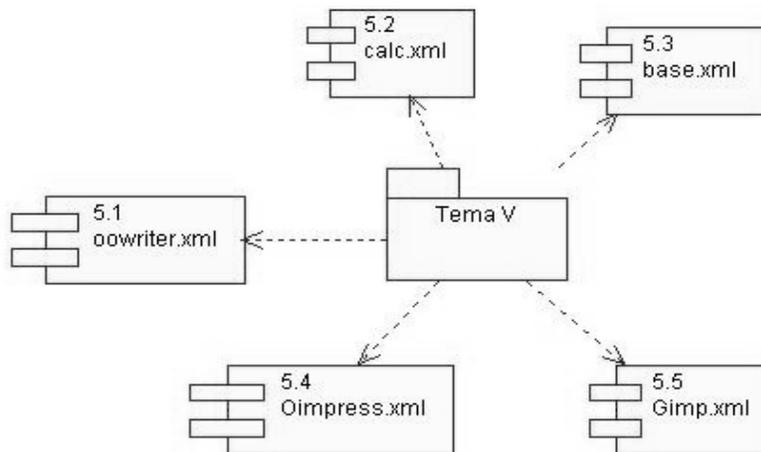
#### 4.4.2.3 Paquete Tema III



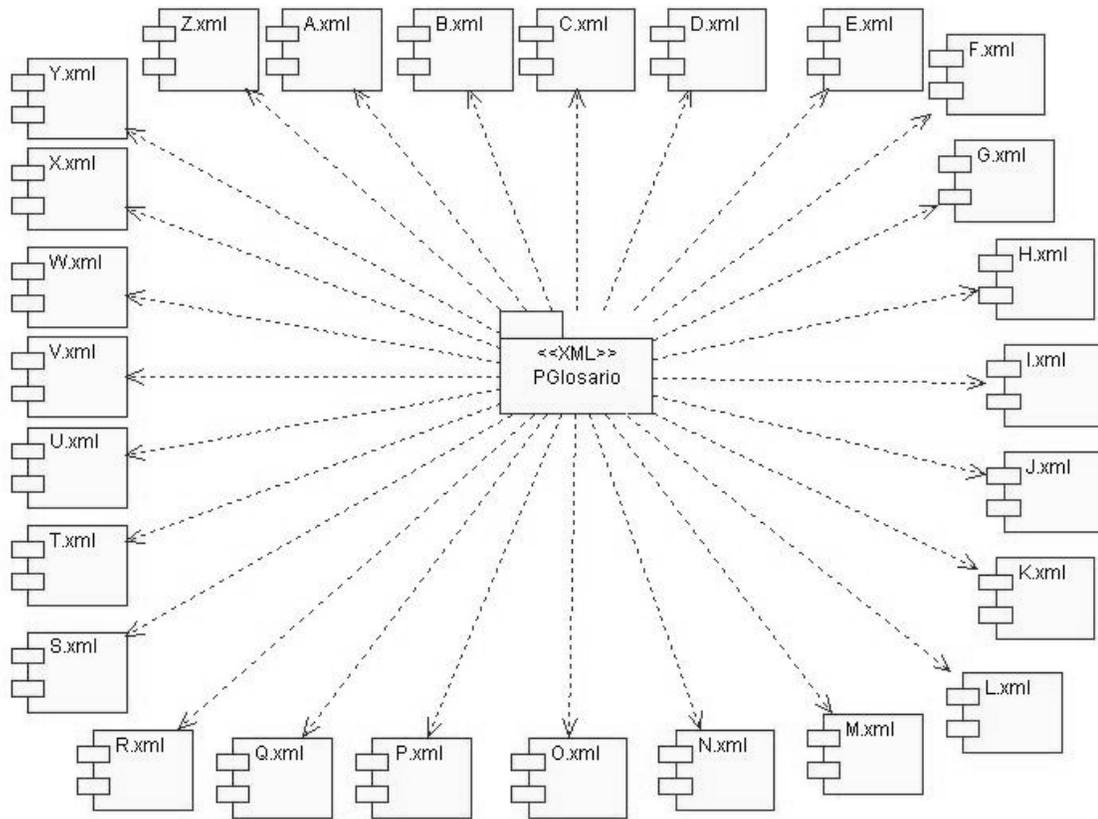
#### 4.4.2.4 Paquete Tema IV



#### 4.4.2.5 Paquete Tema V



#### 4.4.3 Diagrama de componentes del paquete glosario



#### 4.5 Principios del diseño.

- El diseño de interfaces es una labor importante dentro del proceso de un sistema. La calidad de la interfaz de usuario puede ser uno de los motivos que conduzca a un sistema al éxito o al fracaso, es por eso que uno de los aspectos más relevantes de la usabilidad de un sistema es la consistencia de su interfaz de usuario.
- Con el objetivo de desarrollar la interfaz del sistema se analizaron aspectos importantes que posibilitan el bienestar de todo usuario, para ello se tuvo presente la organización, la distribución de la información, el público a quien va dirigido y la temática que se aborda.

- Los usuarios que interactúan con el sistema, lo hacen por medio de una interfaz gráfica por tal razón la misma es amena, sencilla, agradable, presenta facilidad de uso logrando que el usuario se sienta identificado con ella.
- Las pantallas del sistema contienen la información necesaria para evitar la sobrecarga, además de mantener las opciones principales en el mismo lugar de la interfaz para una mejor interacción y adaptabilidad del usuario con la aplicación.
- El sistema manifiesta colores oscuros y pocos llamativos, pues son los que están afín con la temática de GNU/Linux.

### 4.5.1 Estándares de la interfaz de la aplicación

La interfaz de la aplicación fue elaborada teniendo en cuenta algunos de los estándares comunes de las aplicaciones multimedia. La aplicación contiene navegación, ayuda, opción de salir y opción de reproducir sonido.

#### 1 Navegación.

- En primer lugar, aparece un **menú principal** de la aplicación el cual permite el acceso libre a los contenidos lineales de forma rápida.
- En segundo lugar se considera el **árbol de contenidos**, puesto que al profundizar en las diferentes ramas, mediante diferentes pulsaciones del ratón, permitirá acceder a los diferentes epígrafes de los contenidos y al pulsar sobre uno de ellos acceder a la página de contenidos del mismo.
- En tercer lugar tenemos el **Índice temático** el cual consiste en un directorio de temas.
- Se puede acceder a las opciones principales desde cualquier pantalla.

#### 2 Ayuda:

- Puede ser accedida desde cualquier pantalla pues en cada una de las mismas aparece una breve explicación del tópico que se aborda.
- El usuario puede consultarla cuando desee.

#### 3 Salir:

- La opción de salir es accesible desde cualquier pantalla.
- El usuario puede salir cuando desee.

#### 4 Reproducir sonido:

- La reproducción del sonido se puede efectuar desde cualquier pantalla.
- El sonido tiene dos estados, de activado y desactivado.
- El usuario puede reproducir el sonido cuando desee.

### 5 Tipografía

- Verdana 12pts Negrita: Se utiliza para mostrar los botones que permiten la navegación dentro de la Aplicación.
- Verdana 12 pts: Se utiliza para el índice de temáticas, además de todos los textos de contenido que aparecen.

### 6 Color

- Los colores que predominan, son el negro, el gris y el naranja, debido a que el sistema operativo GNU/Linux se caracteriza por usar colores serios, y poco llamativos.

### 7 Imagen

- Las imágenes usadas son de formato JPG ya que soporta 16,7 millones de colores (24 bits) y es el más empleado (y adecuado) para las fotografías. Las imágenes son usadas exclusivamente en las pantallas donde el texto mostrado necesite de una información visual y en la galería.

### 8 Video

- El formato de los videos son .swf, debido a que son animaciones, pues se usó la herramienta Wink para capturar el ambiente de GNU/Linux y elaborar videos de captura de pantalla con él.

#### 4.5.2 Estándares de codificación

Los estándares de codificación son reglas específicas a una lengua que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran.

Teniendo en cuenta que el desarrollo de aplicaciones no termina con la entrega del producto al usuario final, sino que siempre está sujeta a posibles mejoras, es necesario facilitar elementos para que otros desarrolladores entiendan el código fuente. Por tanto el código debe ser comentado de manera clara y se deben fijar algunos estándares en el nombre de identificadores, de variables, de clases, entre otros.

La nomenclatura utilizada fue la siguiente:

- Para el nombre de las clases controladoras se utilizó el prefijo **C\_**nombclase delante de cada clase.
- Para el nombre de las clases entidad se utilizó el prefijo **E\_**nombclase delante de cada clase.
- Para el nombre de clips de películas se utilizó el sufijo **\_mc**.
- Para el nombre de botones se utilizó el sufijo **\_btn**.

### **4.6 Descripción de archivos XML**

Es conocido por todos, que las bases de datos son una parte fundamental de todas las organizaciones, pues en ellas se almacena información crucial para el buen desempeño de las mismas.

Es conocido que XML (eXtensible Language Markup) es el presente y futuro de la administración de datos, pues este lenguaje ha permitido romper barreras y crear una manera estándar de procesar la información.

En la actualidad, XML está provocando la aparición de nuevas tecnologías, entre ellas, la aparición de una nueva generación de bases de datos, que si bien se encuentran en una fase de investigación y desarrollo, en un futuro pueden ser una buena alternativa a las ya conocidas bases de datos relacionales. Estos son aquellos basados netamente en XML o "native XML database".

Este tipo de bases de datos son completamente distintas a las relacionales, las cuales en la actualidad tienen soporte para XML, pero aún siguen almacenando toda la información de manera relacional, es decir en forma tabular (tablas, registros y columnas) o caso contrario almacenan todo el documento en formato Binary Large Object (BLOB), pero la principal característica que brindan estas bases de datos es la capacidad de obtener los resultados de las consultas en formato XML; es por ello que dichas bases de datos pertenecen a la categoría de "XML-enabled database".

En estos momentos, este tipo de base de datos es perfecta para las organizaciones que mantienen su información en diferente formatos, debido a que les permitiría rápidamente pasar su información a un formato XML, utilizando la aplicación de su preferencia, y de esta manera almacenarla en una de estas bases de datos y evitarse el proceso de conversión de un formato a otro. Además, teniendo en cuenta que por ahora aún no existe un lenguaje estándar para el procesamiento de los datos, estas bases de datos son adecuadas para realizar búsquedas sobre los documentos almacenados.

Conforme todas las falencias que tienen estas bases de datos sean superadas, podrán ser una muy buena alternativa para el almacenamiento de la información no solo porque la información estará

almacenada en un formato estándar, XML, sino porque facilitará el desarrollo de aplicaciones basadas en XML, como por ejemplo los Web Services.

Los archivos XML que posee la aplicación, se utilizan con el objetivo de identificar línea a línea el contenido de la misma, además para garantizar independencia de los datos respecto de la aplicación, haciendo lectura del contenido que se necesita mostrar en cada momento.

La estructura que presentan los XML, son las siguientes:

#### 4.6.1 Descripción del archivo XML Menú

Este archivo tiene como objetivo almacenar la información del Menú temáticas, donde aparecerán todas las temáticas con las subtemáticas a presentar en la aplicación, y con el cual el usuario estará interactuando continuamente.

```
<menu>
  <botones>
    <boton id = "1">
      <nombreboton> </nombreboton>
      <accion> </accion>
      <cant>4</cant>
      <subtn1>
        <nombre> </nombre>
        <accion> subbotones </accion>
      </subtn1>
    <boton id = "2">
      <nombreboton></nombreboton>
      <accion>subbotones</accion>
    </subtn1>
      :
      :
    </botones>
  </menu>
```

#### 4.6.2 Descripción del archivo XML Contenido

Este archivo tiene como objetivo almacenar la información de cada una de las temáticas que se abordan en la aplicación. La estructura es la misma para todos los archivos.

```
<texto>
```

```
    ..aquí aparece el contenido de los tópicos que se abordan..
```

Ejemplo: Si se estuviera hablando de los Sistemas Operativos, aparecería el texto de forma íntegra que aborde este tema.

```
</texto>
```

#### 4.6.3 Descripción del archivo XML Ayuda

Este archivo tiene como objetivo almacenar la información correspondiente a la ayuda de la aplicación, presenta una estructura bastante sencilla. La aplicación posee una ayuda para cada pantalla o escenario, por lo que existe un archivo XML diferente para cada uno de estos que siguen la misma estructura.

```
<ayuda>
```

```
    ..aquí aparece el contenido de Ayuda..
```

```
</ayuda>
```

#### 4.6.4 Descripción del archivo XML Glosario

Este archivo tiene como objetivo almacenar todas las palabras y su correspondiente significado que van a ser mostrados en la pantalla Glosario. Para cada letra del abecedario existe un archivo XML con la misma estructura que el resto de los XML.

Cada término tiene un id que identifica una palabra, la cual tiene un significado.

Ejemplo:

```
<terminos>
```

```
  <termino id = "1">
```

```
    <palabra> </palabra>
```

```
    <significado> </significado>
```

```
  </termino>
```

```
  <termino id = "2">
```

```

    <palabra> </palabra>
    <significado> </significado>
  </termino>
  :
  :
</terminos>

```

#### 4.6.5 Descripción del archivo XML Ejercicios SU

Este archivo guarda el contenido de los ejercicios de tipo selección única. El mismo contiene una cantidad de preguntas, un id para cada pregunta, las preguntas, con una única posible respuesta correcta (RC), y varias opciones de respuestas incorrectas (R1, R2....Rn).

Ejemplo:

```

<ejerciciosSU>
  <cantPreg></cantPreg>
  <pregunta id = "1">
    <preg>preg>
    <RC></RC>
    <cantR></cantR>
    <R1></R1>
    :
    :
  </pregunta>
  <pregunta id = "2">
    <preg>preg>
    <RC></RC>
    <cantR></cantR>
    <R1></R1>
    :
    :
  </pregunta>
</ejerciciosSU>

```

#### 4.6.6 Descripción del archivo XML Ejercicios VF

Este archivo almacena todos los ejercicios de tipo verdadero o falso que se muestran en la aplicación. Su estructura es la siguiente:

```
<ejerciciosVF>
  <cantPreg></cantPreg>
  <pregunta id = "1">
    <preg> preg>
    <rta> </rta>
  </pregunta>
  <pregunta id = "2">
    <preg> preg>
    <rta> </rta>
  </pregunta>
  :
</ejerciciosVF>
```

#### 4.6.7 Descripción del archivo XML Imagen

Este archivo almacena la información correspondiente a cada imagen que aparece en la aplicación, asignándole un número a cada una de ellas a modo de identificador. Cada imagen posee además un nombre y una pequeña descripción. Ejemplo:

```
<imagenes>
  <imagen id = "1">
    <nombre> </nombre>
    <descripcion> </descripcion>
  </imagen>
  <imagen id = "2">
    <nombre> </nombre>
    <descripcion> </descripcion>
  </imagen>
  :
```

</imagenes>

#### 4.6.8 Descripción del archivo XML Video

Este archivo almacena la información correspondiente a cada video que aparece en la aplicación, donde permite identificar cada video con id, y que además del nombre permite mostrar una pequeña descripción del mismo. Sería una estructura similar a la de las imágenes. Ejemplo:

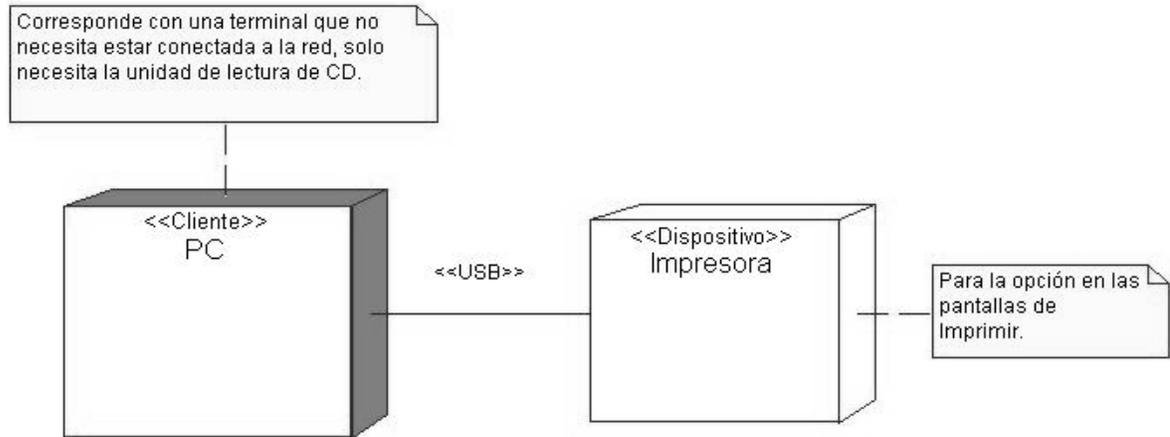
```
<videos>
  <video id="...">
    <nombre> </nombre>
    <descripcion> descripcion>
  </video>
  <video id="...">
    <nombre> </nombre>
    <descripcion> descripcion>
  </video>
  :
  :
</videos>
```

En cada uno de los archivos XML, se utilizan etiquetas HTML y CSS, para dar estilo a la información al mostrarla.

#### 4.7 Diagrama de Despliegue

Un Modelo de Despliegue modela la arquitectura en tiempo de ejecución de un sistema. Forma parte de la vista de despliegue de UML, la cual representa la disposición de las instancias de componentes de ejecución en instancias de nodos conectados por enlaces de comunicación. Un nodo es un recurso de ejecución tal como un computador, un dispositivo o memoria.

El Modelo de Despliegue planteado está compuesto por dos nodos, el primero representa la computadora donde se ejecuta la aplicación y el segundo impresora como dispositivo usado por el sistema una vez que el usuario selecciona la opción de imprimir un determinado contenido.



## 4.8 Conclusiones

En este capítulo se le ha dado cumplimiento a los flujos de trabajo de diseño e implementación, se han generado una serie de artefactos, como son: el diagrama de presentación en el cual se muestran los escenarios de la aplicación, el diagrama de clases de diseño, el cual muestra la interacción entre las clases de objetos, el diagrama de secuencia donde se muestran las diferentes acciones del sistema, el diagrama de componentes donde se pueden ver los seis tipos de componentes utilizados en la aplicación (.exe, .swf, .fla, .xml, .css y .as) y las relaciones que se establecen entre ellos. Además se modela la arquitectura en tiempo de ejecución del sistema mediante el modelo de despliegue presentando de este modo los nodos fundamentales necesarios para la ejecución de la aplicación. Teniendo en cuenta los principios de diseño utilizados se logró obtener un software de interfaz sencilla y fácil de utilizar, además se realizó la descripción de los archivos XML, con el objetivo de facilitar a futuros programadores el entendimiento del sistema y resulte más fácil la adición de nuevos contenidos y funcionalidades, quedando confeccionada de esta manera la propuesta de solución.

# CAPÍTULO 5

## Estudio de la Factibilidad.

### 5.1 Introducción

Para la realización de un proyecto es importante: estimar el esfuerzo humano, el tiempo de desarrollo que se requiere para la ejecución del mismo y también su costo. En este capítulo se realizará el estudio de factibilidad del sistema utilizando el modelo del análisis de Puntos de Casos de Uso, que es un método propuesto originalmente por Gustav Karner de Objectory AB, y posteriormente refinado por muchos otros autores.

### 5.2 Estimación de Tiempo de Desarrollo, Costo y Esfuerzo

#### ➤ Cálculo de Puntos de Casos de Uso sin ajustar

El primer paso para la estimación consiste en el cálculo de los Puntos de Casos de Uso sin ajustar. Este valor, se calcula a partir de la siguiente ecuación:

$$\text{UUCP} = \text{UAW} + \text{UUCW}$$

Donde:

- ❖ **UUCP**: Puntos de Casos de Uso sin ajustar
- ❖ **UAW**: Factor de Peso de los Actores sin ajustar
- ❖ **UUCW**: Factor de Peso de los Casos de Uso sin ajustar

#### Factor de Peso de los Actores sin ajustar (UAW)

El usuario constituye un actor de tipo complejo, ya que se trata de una persona utilizando el sistema mediante una interfaz gráfica, al cual se le asigna un peso 3. Luego, el factor de peso de los actores sin ajustar resulta:

$$\text{UAW} = 1 \times 3 = 3$$

#### Factor de Peso de los Casos de Uso sin ajustar (UUCW)

Cada uno de los Casos de Uso Mostrar ayuda, Controlar sonido, Mostrar Contenido, Imprimir y Mostrar glosario tienen una única transacción. Permitir salida, Interactuar con palabras calientes y Mostrar galería

tiene 2 transacciones. Permitir navegación y Visualizar media tienen 3 transacciones e interactuar con ejercicio tiene 4 transacciones, para un total de 11 casos de usos, de ellos 10 de tipo simples con un (peso 5) y 1 caso de uso de tipo complejo con un (peso 10) con lo cual el factor de peso de los casos de uso sin ajustar resulta:

$$\mathbf{UUCW = (10 \times 5) + (1 \times 10) = 60.}$$

Finalmente, los Puntos de Casos de Uso sin ajustar resultan:

$$\mathbf{UUCP = UAW + UUCW = 3 + 60 = 63.}$$

➤ **Cálculo de Puntos de Casos de Uso ajustados**

Una vez que se tienen los Puntos de Casos de Uso sin ajustar, se debe ajustar éste valor mediante la siguiente ecuación:

$$\mathbf{UCP = UUCP \times TCF \times EF}$$

Donde:

- ❖ **UCP:** Puntos de Casos de Uso ajustados
- ❖ **UUCP:** Puntos de Casos de Uso sin ajustar
- ❖ **TCF:** Factor de complejidad técnica
- ❖ **EF:** Factor de ambiente

➤ **Factor de complejidad técnica (TCF)**

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante.

El Factor de complejidad técnica se calcula mediante la siguiente ecuación:

$$\mathbf{TCF = 0.6 + 0.01 \times \Sigma (\text{Peso}(i) \times \text{Valor asignado}(i))}$$

➤ **Factor de ambiente (EF)**

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El

cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5.

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 \times \sum (\text{Peso}(i) \times \text{Valor asignado}(i))$$

**Factor de complejidad técnica (TCF)**

Factor	Descripción	Peso	Valor asignado	Comentario
T1	Sistema distribuido.	2	0	El sistema es centralizado.
T2	Objetivos de performance o tiempo de respuesta.	1	0	La velocidad de respuesta es de apenas segundos de tiempo.
T3	Eficiencia del usuario final.	1	1	Escasas restricciones de eficiencia.
T4	Procesamiento interno complejo.	1	1	No hay cálculos complejos.
T5	El código debe ser reutilizable.	1	5	Se requiere que el código sea reutilizable.
T6	Facilidad de instalación.	0.5	4	Existen requerimientos de facilidad de instalación.
T7	Facilidad de uso.	0.5	5	Alta
T8	Portabilidad.	2	4	Se requiere que sea portable.
T9	Facilidad de cambio.	1	3	Se requiere un costo moderado de mantenimiento

T10	Concurrencia.	1	0	No hay concurrencia
T11	Incluye objetivos especiales de seguridad.	1	3	Seguridad normal.
T12	Provee acceso directo a terceras partes.	1	3	Los usuarios tienen accesos directos.
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	1	1	Sistema es fácil de usar por cualquier usuario.

**El Factor de complejidad técnica resulta:**

$$TCF = 0.6 + 0.01 \times 29.5 \approx 0.90$$

**Factor de ambiente EF**

Factor	Descripción	Peso	Valor asignado	Comentario
E1	Familiaridad con el modelo de proyecto utilizado.	1.5	3	El grupo no está muy bien familiarizado con el modelo.
E2	Experiencia en la aplicación.	0.5	4	La mayoría del grupo ha trabajado algún tiempo en este tipo de aplicación.
E3	Experiencia en orientación a objetos.	1	4	La mayoría del grupo programa orientado a objetos.
E4	Capacidad del analista líder.	0.5	2	No existe un especialista, aunque se tienen conocimientos bastos.

E5	Motivación.	1	5	El grupo está altamente motivado
E6	Estabilidad de los requerimientos.	2	4	No se esperan muchos cambios.
E7	Personal part-time.	-1	2	Parte del grupo es part-time.
E8	Dificultad del lenguaje de programación.	-1	2	Se usará el lenguaje de programación actionscript 2.0.

**El Factor de ambiente resulta:**

$$EF = 1.4 - 0.03 \times 20.5 \approx 0.79$$

➤ **Finalmente, los Puntos de Casos de Uso ajustados resultan:**

$$UCP = 60 * 0.90 * 0.79 = 42,66$$

➤ **De los Puntos de Casos de Uso a la estimación del esfuerzo**

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

Donde,

- ❖ **E:** esfuerzo estimado en horas-hombre
- ❖ **UCP:** Puntos de Casos de Uso ajustados
- ❖ **CF:** factor de conversión

Se debe tener en cuenta que éste método proporciona una estimación del esfuerzo en horas-hombre contemplando sólo el desarrollo de la funcionalidad especificada en los casos de uso.

Para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello se puede tener en cuenta el siguiente

criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto.

Con éste criterio, y tomando como entrada la estimación de tiempo calculada a partir de los Puntos de Casos de Uso, se pueden calcular las demás estimaciones para obtener la duración total del proyecto.

Aplicando éstos criterios, se obtiene el **esfuerzo necesario** para el desarrollo de los casos de uso como:

$$E = 42,66 * 20 = 853,2 \text{ Horas-Hombre}$$

Actividad	Porcentaje	Horas-Hombres
Análisis	10.00%	213,3
Diseño	20.00%	426,6
Programación	40.00%	853,2
Pruebas	15.00%	319,95
Sobrecarga (otras actividades)	15.00%	319,95
Total	100 %	2133

**E = 2133 Horas-Hombre**

El esfuerzo calculado está dado en horas / hombre pero se necesita convertirlo en hombres / mes, por lo que si se tiene en cuenta que la jornada laboral de un día de trabajo es de 8 horas, y en un mes se trabajan aproximadamente 24 días de este, entonces el esfuerzo sería 11.11 hombres / mes.

Para calcular el tiempo de desarrollo del proyecto se aplican las fórmulas de Bohem siguientes:

$$TDEV = C \times (PM)^F \text{ donde:}$$

$$F = D + 0.2 \times 0.01 \times \sum SF_j \text{ (j=1 hasta j=5)}$$

TDEV: Tiempo de Desarrollo.

$$C = 3.67$$

$$D = 0.28$$

SFj: Factores de escala

El significado de los factores de escala y el valor asignado se muestran en la siguiente tabla:

Factor de Escala	Descripción	Clasificación	Valor
PREC	Variable de precedencia u orden secuencial del desarrollo	nominal	3.72
FLEX	Variable de flexibilidad del desarrollo	alto	2.03
RESL	Indica la fortaleza de la arquitectura y métodos de estimación y reducción de riesgos.	bajo	5.65
TEAM	Esta variable refleja la cohesión y madurez del equipo de trabajo.	alto	2.19
PMAT	Relaciona el proceso de madurez del software.	alto	3.12
		Total	16.71

$$F = 0.28 + 0.2 \times 0.01 \times (3.72 + 2.03 + 5.65 + 2.19 + 3.12)$$

$$F = 0.28 + 0.2 \times 0.01 \times 16.71$$

$$F = 0.28 + 0.03342$$

$$F = 0.31342$$

$$TDEV = 3.67 \times (11.11)^{0.31342}$$

$$TDEV = 3.67 \times 2.12$$

$$\mathbf{TDEV = 7.7804}$$

A partir del tiempo de desarrollo estimado y el esfuerzo en hombres / mes se procede a calcular la cantidad de hombres para el desarrollo del proyecto mediante la ecuación:

$$\mathbf{CH = Pm / TDEV}$$
 donde:

CH = Cantidad de hombres

$$CH = 11.11 / 7.7804$$

**CH = 1.42**

Como la cantidad de hombres calculada es distinta de la cantidad de hombres real que desarrolla el proyecto, se calcula nuevamente el tiempo de desarrollo donde se este se ajusta a la cantidad real de hombres, que en este caso son 2.

TDEV Real =  $P_m / CH$  Real donde:

TDEV Real = tiempo de desarrollo real.

CH Real = cantidad de hombres real

TDEV Real =  $11.11 / 2$

**TDEV = 5.5 meses**

Para calcular los costos se tienen en cuenta el tiempo de desarrollo, la cantidad de hombres y el salario básico de un adiestrado que seria \$225 MN

Costo = TDEV x Salario x CH

Costo =  $5.5 \times \$225 \times 2$

**Costo = \$ 2475**

### **5.3 Beneficios tangibles e intangibles.**

#### **5.3.1 Tangibles**

La aplicación que se desarrolla forma parte de un proyecto de la facultad 10 con el objetivo de lograr una mejor preparación en el trabajo del sistema operativo GNU/Linux. El costo por desarrollar la aplicación es de **\$2475** pesos (moneda nacional), el cual es perfectamente reparable si en un futuro se comercializara.

#### **5.3.2 Intangibles**

Como beneficios intangibles asociados al desarrollo de Multimedia “Aprendiendo GNU/Linux” se señalan los siguientes:

- Aumento de la cultura informática.

- Centralización de la información disponible sobre el sistema operativo GNU/Linux.
- Aumento en la preparación y estudio de las personas sobre este sistema operativo.
- Aumento de la motivación en el estudio de GNU/Linux en una nueva forma interactiva.
- Aumento de la calidad de los resultados que se pueden alcanzar en el estudio de este sistema operativo.
- Mayor y mejor aprovechamiento de las tecnologías de la información.

#### 5.4 Análisis de costo-beneficio.

El desarrollo de este sistema no supone grandes gastos de recursos, ni tampoco de tiempo; los datos que contiene toda la información referente al sistema se encuentran en archivos XML, y pueden ser alojados sin problema alguno, ya que los mismos tienen buenas prestaciones y acceso rápido y no necesita de ningún gestor de base de datos.

La fácil utilización y navegabilidad del software no genera daño físico alguno a los manipuladores de este, ni a los dispositivos utilizados para su uso. La interfaz está diseñada de forma sencilla y resulta agradable al entorno del usuario.

- Es factible desarrollar una aplicación para centralizar la información existente sobre el sistema operativo GNU/Linux, brindando un medio de apoyo al estudio del mismo.

#### 5.5 Conclusiones.

Una vez terminado el estudio de factibilidad del sistema, se puede brindar el siguiente estimado:

Parámetros	Valores
Esfuerzo	2133 Horas-Hombre
Tiempo de desarrollo	5.5 Meses
Cantidad de hombres	2 Hombres
Salario min	\$225
Costo	\$2475

Con el desarrollo de este capítulo se expresa claramente los aspectos que de alguna manera influyen en el costo de un proyecto con tecnología multimedia y particularmente de este software informativo. Se puntualiza de una forma clara los costos a incurrir, el tiempo de desarrollo, esfuerzo y los beneficios que aporta el software multimedia. Entre los beneficios que aporta la construcción de este producto se encuentra la ventaja que implica ahorrar recursos humanos, tiempo de desarrollo, así como la utilización del producto.

## Conclusiones

Luego del estudio realizado y analizado las tendencias actuales, los lenguajes y metodologías a utilizar y efectuado el diseño correspondiente, se implementó un software con tecnología multimedia de carácter informativo sobre GNU/Linux como material de apoyo para el estudio de este sistema operativo, logrando

El desarrollo del producto está basado en la metodología RUP, haciendo uso del lenguaje UML y de OMMMA-L como extensión del anterior. La aplicación posee un ambiente sencillo y es fácil de manejar, cumpliendo con estándares de diseño y aplicando la programación orientada a objetos, resulta además de fácil mantenimiento, logrando que sea dinámica, permitiendo con esto, la incorporación de nuevos contenidos en caso de que se necesite.

Fue desarrollado por dos personas estimándose un costo total de \$2475 MN, considerándose de esta manera y de acuerdo a los beneficios que reporta, factible la elaboración de este producto.

Por todos los planteamientos descritos anteriormente, se puede concluir que se han cumplido satisfactoriamente los objetivos descritos al inicio del presente trabajo.

## Recomendaciones

De manera general se puede plantear que el software con tecnología multimedia implementado cumple con los objetivos descritos al inicio del trabajo, sin embargo, hacemos las siguientes recomendaciones:

- Continuar el desarrollo de esta aplicación, adicionándole nuevas funcionalidades.
- Recopilar más información sobre el sistema operativo GNU/Linux e incorporar nuevos contenidos al producto final.
- Hacer extensible a los IPI del país, la utilización del producto como material de apoyo para el estudio del sistema operativo GNU/Linux.

## Referencias bibliográficas

- AGUSTIN, M. *Herramienta Revolution* 2007. [Disponible en: <http://www.disca.upv.es/magustim/mmmultiplataforma/c2361.htm#SECCIOPRES-ENTACIODELAFE>]
- BERROCO, J. V. *El guión Multimedia. Principios* 2007. [Disponible en: <http://www.unex.es/didactica/Tecnologia-Educativa/quion11.htm>]
- CIBERAULA. *Macromedia FreeHand*, 2007. [Disponible en: <http://www.ciberaula.com/curso/freehandmx/>]
- CORPORATION., R. *Lo nuevo de Rational Rose 2000*, 2004. [Disponible en: <http://www.abists.com.mf/Fabs/Rational/notasTK>].
- DÍAZ, C. C. *LA TECNOLOGIA MULTIMEDIA: Una Nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones.*, 1994. [Disponible en: <http://www.monografias.com/trabajos7/mult/mult2.shtml>]
- EDUCAR, P. *Concepto de Interactividad*, 2007. [Disponible en: <http://portal.educar.org/multimediam/blog/queesmultimedaiinteractiva>]
- EDUTEKA. *TOOLBOOK. Diseño y desarrollo de aplicaciones multimedia educativas.* Disponible en: <http://www.uv.es/bellochc/pwedu6.htm>
- ENCICLOPEDIA. *ACTION SCRIPT.* Disponible en: <http://enciclopedia.us.es/index.php/ActionScript>
- FREDDIE. *ACTION SCRIPT 2.0*, 2004. [Disponible en: <http://www.cristalab.com/tutoriales/21/tutorial-de-actionscript-2.0-en-flash>]
- HISTCHFELD, P. S. C. Y. N. *Tutorial de UML.*, 2000. p.
- ISO/TC, C. T. *Ergonomía del software para interfaces de usuario multimedia.*, 2002. [Disponible en: <http://www.ni4.org/modules/Downloads/archivos/legislacion/iso14915-1.pdf>]
- JACOBSON, I.; G. BOOCH, et al. *El proceso unificado de desarrollo de software.* 2000. p.
- MÁRQUEZ, D. G. P. *Ventajas e inconvenientes del software Multimedia.*, 2000. [Disponible en: <http://dewey.uab.es/pmarques/ventajas.htm>]
- MENDOZA, M. F. *Introduccion al XML.* Disponible en: <http://www.monografias.com/trabajos6/ixml/ixml.shtml#xml>
- ORTIZ, J. S. *Aspectos básicos de Neobook*, 2007.
- RODRIGUEZ, D. F. Z. *Características de la Multimedia.*, 2006. [Disponible en: <http://www.monografias.com/trabajos10/mmedia/mmedia.shtml>]

- SALINAS, D. J. M. *Multimedia en los procesos de enseñanza - aprendizaje: Elementos de discusión.*, 1996. [Disponible en: <http://www.uib.es/depart/qte/multimedia.html>]
- SAUER, S. *Extending UML for Modeling of Multimedia Applications.*
- SAUER., S. *Xtending UML for Modeling of Multimedia Applications.*, 2004. [Disponible en: <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>]
- SERGIO MARTÍNEZ MAILLO , P. A. J. L. F. A., PROF INSTRUCTOR ALEXEI CABRALES LEYVA , PROF INSTRUCTOR MARCOS LEÓN FONSECA. *Mediator*, 2007. [Disponible en: <http://www.monografias.com/trabajos23/mediator-seis/mediator-seis.shtml>].
- STALLMAN, R. M. *GNU/Linux y la libertad.*, 2002. [Disponible en: <http://www.gnu.org/home.es.html>]
- WIKIPEDIA. *Adobe Illustrator.* Disponible en: [http://es.wikipedia.org/wiki/Adobe\\_Illustrator](http://es.wikipedia.org/wiki/Adobe_Illustrator)
- . *HTML.* Disponible en: <http://es.wikipedia.org/wiki/HTML>
- . *Lenguaje de Mercado.* Disponible en: [http://es.wikipedia.org/wiki/Lenguaje\\_de\\_mercado](http://es.wikipedia.org/wiki/Lenguaje_de_mercado)
- . *Lenguajes de programación.* Disponible en: [http://es.wikipedia.org/wiki/Lenguaje\\_de\\_programaci%C3%B3n](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n)
- . *Macromedia Director,* 2007. [Disponible en: [http://es.wikipedia.org/wiki/Macromedia\\_Director](http://es.wikipedia.org/wiki/Macromedia_Director)]
- . *Photoshop.* Disponible en: [http://es.wikipedia.org/wiki/Adobe\\_Photoshop](http://es.wikipedia.org/wiki/Adobe_Photoshop)

**Bibliografía**

1. ACADEMIA, D. D. L. R. Diccionario electrónico, 2007.
2. BARROS, D. R. Conceptos generales sobre sistemas hipermediales., 2004. p.
3. BERROCO, J. V. El guión Multimedia. Principios 2007. [Disponible en: [http://www.unex.es/didactica/Tecnologia\\_Educativa/quion11.htm](http://www.unex.es/didactica/Tecnologia_Educativa/quion11.htm)]
4. CIBERAULA. Macromedia FreeHand, 2007. [Disponible en: <http://www.ciberaula.com/curso/freehandmx/>]
5. CORPORATION., R. Lo nuevo de Rational Rose 2000, 2004. [Disponible en: <http://www.abists.com.mf/Fabs/Rational/notasTK>].
6. DEBIAN, C. Sitio de Debian, 2007. [Disponible en: <http://www.debiancuba.org/>]
7. DÍAZ, C. C. LA TECNOLOGIA MULTIMEDIA: Una Nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones., 1994. [Disponible en: <http://www.monografias.com/trabajos7/mult/mult2.shtml>]
8. EDUCAR, P. Concepto de Interactividad, 2007. [Disponible en: <http://portal.educar.org/multimediam/blog/queesmultimediaminteractiva>]
9. EDUTEKA. TOOLBOOK. Diseño y desarrollo de aplicaciones multimedia educativas., 2007. [Disponible en: <http://www.uv.es/bellochc/pwedu6.htm>]
10. ENCICLOPEDIA. ACTION SCRIPT 2007. [Disponible en: <http://enciclopedia.us.es/index.php/ActionScript>]
11. ENGELS, G. Specification of Interactive Multimedia Applications. 2004. [Disponible en: <http://wwwcs.upb.de/cs/ag-engels/Papers/2001/SauerHCC01.pdf>].
12. FEBE ÁNGEL CIUDAD RICARDO, S. M. EMBRIOCIM – Enciclopedia de Embriología Médica – Colección GALENOMEDIA. Cuba, INSTITUTO SUPERIOR POLITÉCNICO “JOSÉ ANTONIO ECHEVARRÍA”. FACULTAD DE INGENIERÍA INDUSTRIAL, 2004. p.
13. FERNÁNDEZ., G. S. and S. D. CATALÁ. MULTIMEDIA AUTO-APRENDE. Cuba, INSTITUTO SUPERIOR POLITÉCNICO “JOSÉ ANTONIO ECHEVARRÍA”. FACULTAD DE INGENIERÍA INDUSTRIAL 2006. p.
14. FREDDIE. ACTION SCRIPT 2.0, 2004. [Disponible en: <http://www.cristalab.com/tutoriales/21/tutorial-de-actionscript-2.0-en-flash>]
15. HISTCHFELD, P. S. C. Y. N. Tutorial de UML., 2000. p.
16. ISO/TC, C. T. Ergonomía del software para interfaces de usuario multimedia., 2002. [Disponible en: <http://www.ni4.org/modules/Downloads/archivos/legislacion/iso14915-1.pdf>]

17. IVARJACOBSON, G. B., JAMES RUMBAUGH. El lenguaje Unificado de Modelado. Manual de referencia. 2000. p.
18. JACOBSON, I.; G. BOOCH, et al. El proceso unificado de desarrollo de software. 2000. p.
19. JIMÉNEZ, S. V. Constitución Volumen II. Multimedia Educativa. Cuba, Universidad de las Ciencias Informáticas., 2006. p.
20. LARMAN, C. Introducción al análisis y diseño orientado a objeto., 1999. p.
21. LEÓN, R. R. La Arquitectura de la Información y las Ciencias de la Información, 2005. [Disponible en: [http://www.nosolousabilidad.com/articulos/ai\\_cc\\_informacion.htm#rosenfeld](http://www.nosolousabilidad.com/articulos/ai_cc_informacion.htm#rosenfeld)
22. MÁRQUEZ, D. G. P. Ventajas e inconvenientes del software Multimedia., 2000. [Disponible en: <http://dewey.uab.es/pmarques/ventajas.htm>
23. MENDOZA, M. F. Introducción al XML, 2007. [Disponible en: <http://www.monografias.com/trabajos6/ixml/ixml.shtml#xml>
24. MONOGRAFIAS. Mediator, 2007. [Disponible en: <http://www.monografias.com/trabajos23/mediator-seis/mediator-seis.shtml>
25. Multimedia interactiva, 2007. [Disponible en: <http://www.monografias.com/trabajos33/multimedia-interactiva/multimedia-interactiva2.shtml>
26. ORTIZ, J. S. Aspectos básicos de Neobook, 2007.
27. PEÑA, R. Artículo de Linux., 2004.
28. PRESSMAN, R. S. Ingeniería del Software. Un enfoque práctico. Quinta Edición., 1998. p.
29. ROBERLÁN RODRÍGUEZ SÁNCHEZ, E. A. Q. C. POO Interactivo: Multimedia para el aprendizaje de la programación orientada a objetos. Cuba, Universidad de las Ciencias Informáticas, 2006. p.
30. RODRIGUEZ, D. F. Z. Características de la Multimedia., 2006. [Disponible en: <http://www.monografias.com/trabajos10/mmedia/mmedia.shtml>
31. SALINAS, D. J. M. Multimedia en los procesos de enseñanza - aprendizaje: Elementos de discusión., 1996. [Disponible en: <http://www.uib.es/depart/gte/multimedia.html>
32. SAUER , S. Xtending UML for Modeling of Multimedia Applications., 2004. [Disponible en: <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>
33. SERGIO MARTÍNEZ MAILLO , P. A. J. L. F. A., PROF INSTRUCTOR ALEXEI CABRALES LEYVA , PROF INSTRUCTOR MARCOS LEÓN FONSECA.

- 
- Mediator, 2007. [Disponible en: <http://www.monografias.com/trabajos23/mediator-seis/mediator-seis.shtml>]
34. STALLMAN, R. M. GNU/Linux y la libertad., 2002. [Disponible en: <http://www.gnu.org/home.es.html>]
35. STEFAN, S. Extending UML for Modeling of Multimedia Applications, 2007.
36. ---. Extending UML for Modeling of Multimedia Applications, 2004 [Disponible en: <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>]
37. WAUGH, J. Debate sobre GNU/Linux en Cuba, 2002. [Disponible en: <http://umeet.uninet.edu/umeet2002/talk/2002-12-12-linux3.txt.html>]
38. WIKIPEDIA. Adobe Illustrator, 2007. [Disponible en: [http://es.wikipedia.org/wiki/Adobe\\_Illustrator](http://es.wikipedia.org/wiki/Adobe_Illustrator)]
39. ---. HTML, 2007. [Disponible en: <http://es.wikipedia.org/wiki/HTML>]
40. ---. Lenguaje de Marcado., 2007. [Disponible en: [http://es.wikipedia.org/wiki/Lenguaje\\_de\\_marcado](http://es.wikipedia.org/wiki/Lenguaje_de_marcado)]
41. ---. Lenguajes de programación, 2007. [Disponible en: [http://es.wikipedia.org/wiki/Lenguaje\\_de\\_programaci%C3%B3n](http://es.wikipedia.org/wiki/Lenguaje_de_programaci%C3%B3n)]
42. ---. Macromedia Director, 2007. [Disponible en: [http://es.wikipedia.org/wiki/Macromedia\\_Director](http://es.wikipedia.org/wiki/Macromedia_Director)]
43. ---. Photoshop, 2007. [Disponible en: [http://es.wikipedia.org/wiki/Adobe\\_Photoshop](http://es.wikipedia.org/wiki/Adobe_Photoshop)]
44. ---. RUP, 2007. [Disponible en: <http://es.wikipedia.org/wiki/RUP>]

## Glosario

- **Temas:** se le llama Temas a los contenidos que se abordan, basados en el sistema operativo GNU/Linux.
- **Capítulo:** Se le llama capítulo a la primera división que tiene el contenido. El contenido esta agrupado en capítulos y estos a su vez en epígrafes. Un capítulo esta formado por 1 o varios epígrafes.
- **Pantalla:** Es la agrupación visual de elementos de medias contenidas en una vista determinada.
- **RUP:** El Proceso Unificado Racional o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso).
- **OMMMA-L:** El Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia es una extensión de UML especializado en aplicaciones multimedia.
- **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, vídeo y sonido.
- **Hipermedia:** Es un término usado como lógica extensión del término Hipertexto, en el cual audio, video, texto e hipervínculos generalmente no secuenciales, se entrelazan para formar un continuo de información, que puede considerarse como virtualmente infinito desde la perspectiva de Internet.
- **MVC:** Es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos.
- **Macromedia Flash:** Programa para la edición y creación de animaciones.
- **CI:** estereotipo para identificar las clases interfaz, encargadas de mostrar la información solicitada

- **CC:** son las clases controladoras correspondiente al patrón de diseño MVC que se encargan de dirigir y controlar el funcionamiento de una petición, decidiendo quien procesa y quien muestra.
- **CE:** son las clases entidades correspondientes al MVC, que contienen los atributos, según el tópico.