

Universidad de las Ciencias Informáticas

FACULTAD 8

Análisis de un IDE para múltiples plataformas con tecnologías y herramientas libres para desarrollar software educativo en formato multimedia. Análisis, diseño e implementación de componentes, ejercicios y clases reutilizables.

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

Autores

Geyser Zamora Sánchez

Yosber Rodríguez Tamayo

Tutor

Ing. Abel Ernesto Lorente Rodríguez

Ciudad de La Habana

Junio del 2008

Agradecimientos:

De Geysler:

A mi mamá Margarita y mi papa Félix por haber estado siempre a mi lado, confiar en mí y apoyarme en todo.

A mi novia Aylin por haber estado siempre a mi lado sin importar el momento o las circunstancias, por haber sido tan paciente, comprensiva y haberme dado tanta felicidad.

A mi hermana Anita por quererme tanto y ser tan incondicional para conmigo.

A mi primo Yudanis y a sus padres por haber sido la principal fuerza que inspiró a tomar esta carrera.

A mi familia, en especial a mi abuela y tío Jesús por haberme apoyado en todas mis decisiones y haberme dado tantos consejos para que fuera alguien en la vida una persona de bien.

A mi tutor Abel por haber sido mi guía y amigo en esta universidad desde mi segundo año.

A mis suegros por haber sido espejos en los cuales siempre he querido verme reflejado.

A mis amistades y compañeros de trabajo Yonnys, Joan Pablo, Yosber, el pequeño Roberto, Anisley, Diana pero en especial a mi amigo y co-tutor Michel Miranda por haber sido tan paciente conmigo, su ayuda fue de vital importancia en el desarrollo de este trabajo de diploma, a él le debo todo lo que pude lograr. Gracias a todos.

De Yosber:

A mi familia por brindarme su apoyo incondicional durante mis años de estudiantes.

A mi abuela por preocuparse por mí.

A mi hermana y a mi hermano por todo su apoyo y por cuidar de mi madre, mi abuela y la niña.

A mis amistades en general, y en especial a los más cercanos: Reinaldo, Roberto, Michel, Marlon, Geysler, Yonnys, Joan Pablo, Aylin, Diana y otros que aunque no menciono saben que le agradezco de todo corazón.

A todos mis compañeros de grupo, compañeros del apto.

Especialmente agradezco a la persona más importante para mí, la persona que sin ella no sería posible ser lo que soy, mis más grandes agradecimientos para ese ser que sacrifica todo sin esperar nada a cambio, un agradecimiento eterno para mi madre.

Dedicatoria:

A nuestros padres, familiares, a nuestro Comandante en jefe por ser el principal precursor de la Universidad de las Ciencias Informáticas y a todos los amigos que siempre han estado presentes para brindarnos su ayuda incondicional.

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 firmo la presente a los ____ días del mes de _____ del año 2008

Autores:

Yosber Rodríguez Tamayo

Geyser Zamora Sánchez

Tutor:

Ing. Abel Ernesto Lorente Rodríguez

Resumen:

Actualmente existen en la Universidad de las Ciencias Informática un gran auge en el campo de las alternativas libres principalmente en el mundo del software educativo en formato multimedia debido a esto han salido una gran gama de soluciones y estrategias para posibilitar esta migración. En ocasiones en los proyectos productivos que se dedican a realizar este tipo de software presentan problemas principalmente con la rapidez de entrega de dichos producto. Esto es debido a que los desarrolladores no poseen de herramientas, clases y componentes ya implementados que les facilite este trabajo y agilice la producción de los mismos. Este trabajo de diploma ofrece una solución a la hora de realizar un producto multimedia, proporciona diversas clases, ejercicios y componentes reutilizables para darle solución con rapidez y eficacia a dichos productos. Por lo anteriormente expuesto esta solución contribuye al mejoramiento de la actividad productiva pues brinda la posibilidad a todos los implicados en el proyecto a realizar software educativo en formato multimedia con la mayor brevedad posible.

Palabras clave: IDE, Herramientas Libres, Multimedia, Software Libre.

Índice:

Introducción:	1
Capítulo 1 Fundamentación Teórica.....	7
1.1 Introducción.....	7
1.2 Informática Educativa en el Mundo	8
1.3 Informática Educativa en Cuba	9
1.4 Análisis de soluciones existentes.....	10
1.5 Estrategia de investigación:	13
1.6 Metodología y lenguaje de modelado	14
1.6.1 Rational Unified Process (RUP)	15
1.6.2 Extreme Programming (XP).....	16
1.6.3 Microsoft Solution Framework (MSF)	17
1.7 ¿Por qué UML como lenguaje de modelado?.....	17
1.8 Herramienta case: Visual Paradigm.....	18
1.9 Definición del lenguaje de codificación.	19
1.10 Herramientas de desarrollo.	19
1.11 Librería para el desarrollo AsWing.....	20
1.12 Conceptos asociados al problema	20
1.13 Conclusiones del Capítulo.....	20
Capítulo 2 Descripción de la Solución Propuesta.....	21
2.1 Introducción.....	21
2.2 Descripción del Modelo de Dominio.....	21
2.3 Solución Propuesta	22
2.3.1 Requisitos Funcionales.....	22
2.3.2 Requerimientos no funcionales de Restricciones en la implementación.....	28
2.3.3 Requerimientos no funcionales de Usabilidad.....	28
2.3.4 Modelo de Casos de Uso del Sistema.....	29
2.3.5 Descripción y expansión de los casos de uso.	30

2.4 Conclusiones del capítulo	43
Capítulo 3 Diseño e Implementación	45
3.1 Introducción.....	45
3.2 Diseño de la aplicación	45
3.2.1 Diagrama de Clases del Diseño:	45
3.3 Arquitectura del Sistema	51
3.4 Patrones GRASP	53
3.5 Modelo de Implementación.	56
3.5.1 Diagrama de componentes de código fuente:	56
3.6 Conclusiones del Capítulo.....	60
Capítulo 4 Factibilidad	61
4.1 Introducción.....	61
4.2 Método de Estimación por Puntos de Casos de Uso.....	61
4.2.1 Cálculo de Puntos de Casos de Uso sin ajustar.....	61
4.2.2 Factor de Peso de los Actores sin ajustar (UAW)	61
4.2.3 Factor de Peso de los Casos de Uso sin ajustar (UUCW)	62
4.2.4 Cálculo de Puntos de Casos de Uso ajustados.....	63
4.2.5 Factor de complejidad técnica (TCF).....	63
4.2.6 Factor de ambiente (EF).....	64
4.2.7 Estimación del esfuerzo.....	65
4.3 Beneficios tangibles	66
4.4 Beneficios intangibles.....	66
4.5 Conclusiones del Capítulo.....	66
Conclusiones Generales.....	67
Recomendaciones	68

Introducción:

El mundo avanza cada día a una velocidad mayor, las nuevas tecnologías quedan obsoletas en breves periodos de tiempo. En la actualidad las nuevas tecnologías de la información y las comunicaciones se han extendido a todos los rincones del mundo, causando un impacto revolucionario en todas las esferas sociales, políticas y económicas, y particularmente sobre los medios de enseñanza tradicionales.

Las computadoras y sus métodos cada día más interactivos y variados de aprendizaje, la posibilidad del acceso al conocimiento global sin restricciones de tiempo, ni espacio; la dependencia tecnológica de la sociedad y las facilidades que esta brinda, así como todas las ventajas asociadas hacen que cada día el desarrollo de estos tipos de productos dirigidos a este sector en particular sean cada vez más cotizados y necesarios. (Ana García-Valcárcel Muñoz-Repiso)

A nivel mundial han surgido nuevos conceptos asociados a la Informática y la Educación. La creación de Aulas Virtuales para fomentar la educación a distancia y el desarrollo de plataformas de tele formación han generado un despunte acelerado de la función de ambas esferas. En los centros de enseñanza se va haciendo énfasis cada vez más en el uso de las plataformas virtuales de aprendizaje. Los estudiantes aprenden y hacen suyos los contenidos de las diferentes materias mediante software educativo destinados a estos fines. Ya no es necesaria la presencia de un profesor impartiendo una materia, guiado por el enfoque tradicional, donde el aprendizaje era de muchos a uno: un grupo de estudiantes atedian un mismo profesor. Las plataformas de aprendizaje han traído un nuevo enfoque, un aprendizaje de muchos a muchos, incrementando la colaboración y las habilidades autodidactas de los estudiantes.

Para lograr un mejor aprendizaje haciendo uso de las computadoras es necesario acercarse algo más a la manera habitual en que los seres humanos nos comunicamos cuando empleamos varios sentidos para comprender un mismo objeto o concepto. Es necesaria una correcta combinación de las diferentes medias con las que contamos, imágenes, sonido, texto, para lograr así una mejora notable en la atención, la comprensión y el aprendizaje. (Ana García-Valcárcel Muñoz-Repiso)

Desde los inicios de la Revolución la educación ha sido una prioridad. Hace ya algún tiempo nuestro país se dio a la tarea de revolucionar y mejorar el sistema de aprendizaje de los niños y jóvenes cubanos, para lograr esto se han realizado una serie de software que han ayudado en la preparación y la motivación de estos así como el desarrollo de diferentes programas que contribuyan a esta preparación.

El Embargo económico y político del Gobierno de los Estados Unidos de América hacia nuestro país también nos ha afectado seriamente en el avance del desarrollo de soluciones informáticas para la educación. La mayor parte de las grandes compañías productoras de herramientas y materiales para este sector son de origen norteamericano; y las trabas que nos impone el gobierno norteamericano para el uso libremente de estos productos nos lleva experimentar la necesidad de incentivar la producción nacional a esta rama de la producción de software basándonos en nuevas soluciones. Pudiendo así solventar nuestras necesidades, y porque no, también abrimos un poco más a la comercialización o instauración de nuestros productos en el exterior. (David Grau Merconchini 2008)

Las tecnologías libres llegaron bastante rápido a Cuba, si se tiene en cuenta que su uso comenzó desde principios de la década de los 90, cuando el proyecto GNU no llegaba aún a los diez años de creado.

Muchos organismos vienen desarrollando en sus instituciones un trabajo constante y meritorio: Salud Pública, la Aduana General de la República, Educación Superior, el Ministerio de Ciencia, Tecnología y Medio Ambiente o el Ministerio de Informática y las Comunicaciones (MIC), entre otros tantos en los que las tecnologías libres muestran un crecimiento sostenido. (David Grau Merconchini 2008)

Hoy se estiman en más de 2 000 los usuarios de distribuciones GNU/Linux, cifra contrastante con unos 200 que se estimaban en 1995.

El gobierno cubano también ha hecho manifiesto su interés en este movimiento,

considerándolo esencial para el desarrollo del país. Al respecto, en el año 2002 fue lanzada una estrategia guiada por el MIC para favorecer la inserción en el país de las tecnologías libres, convicción que reafirmó dos años después. (David Grau Merconchini 2008)

En toda Cuba, se lleva a cabo un plan de migración a software libre y específicamente en la Universidad de las Ciencias Informáticas este proceso de migración repercute negativamente en la producción de software educativo en formatos multimedia.

Para lograr establecer satisfactoriamente una migración gradual en este sentido, los desarrolladores en el área requieren de herramientas que ofrezcan soluciones completas con ambientes de desarrollo muy profesionales como el Adobe Flash, o el nuevo proyecto de Microsoft: Expresión; herramientas que actualmente no existen bajo una licencia que nos permita usarlas libremente o no cubren las necesidades requeridas por nuestros desarrolladores.

Para lograr establecer satisfactoriamente una migración gradual al software libre en la Universidad de las Ciencias Informáticas sin que esto repercuta negativamente en la producción de software educativo en formato multimedia la dirección Universidad de las Ciencias Informáticas en conjunto con la dirección de Software Educativo de la misma se han planteado la necesidad de elaborar o construir una herramienta que posibilite crear software educativo con tecnología multimedia en menor tiempo, con gran calidad y lo más importante con perspectiva libre, es decir para múltiples plataformas.

Situación Problemática:

Llevar a cabo el desarrollo de productos educativos multimedia para múltiples plataforma con tecnologías libres es un reto en la actualidad ya que no existen o no se conocen herramientas de desarrollo que nos permitan cumplir con los requerimientos del formato para multimedia, por otro lado no existe conocimiento de cómo trabajar con algunas de estas herramientas en caso de existir, por estos motivos no se cumple con las necesidades de producción de esta importante línea productiva en la Universidad de las Ciencias Informáticas.

Atendiendo a estas particularidades en la Dirección de Producción # 2 de la Universidad de las Ciencias Informáticas se pretende desarrollar un sistema para aquellos desarrolladores que requieran de herramientas libres para el desarrollo de multimedia y haciendo uso de estándares y formatos abiertos puedan desarrollar cualquier tipo de productos educativos con formato multimedia, sin embargo el camino que se debe correr para alcanzar estas metas apenas está comenzando.

Para lograr esto se propone un sistema que cumpla con las características y componentes necesarios de un ambiente de desarrollo integrado con una interfaz gráfica para usuario final, que permita la construcción de proyectos multimedia con todas las exigencias, facilidades, requisitos y rapidez que los mismos necesitan.

Para lograr este trabajo y ganar también en organización se decidió dividir dicho sistema (IDE) en tres subsistemas:

1. Subsistema de gestión visual.
2. Subsistema de gestión de código.
3. Subsistema de análisis, diseño e implementación de componentes, ejercicios y clases reutilizables.

Problema a Resolver:

La escasez de ejercicios y clases reutilizables impide que se agilice el desarrollo de productos multimedia educativos en la Universidad de las Ciencias Informática y cumplir con las necesidades de producción de la misma.

Objetivo General:

Proporcionar a los desarrolladores una serie de componentes, clases y ejercicios reutilizables para un rápido desarrollo de aplicaciones multimedia educativas e incluir estas clases en el IDE a desarrollar.

Objetivos Específicos:

1. Realizar un estudio y documentación de las diferentes tipologías de ejercicios,

los componentes y las clases que se pueden desarrollar en la implementación de software educativo multimedia

2. Realizar el análisis y diseño para la implementación de componentes, clases y ejercicios.
3. Implementar componentes, clases y ejercicios reutilizables para la su utilización en el IDE a desarrollar.

Para dar cumplimiento a los objetivos de este trabajo se concibieron las siguientes **tareas de la investigación:**

- Revisar estado del arte.
 1. Revisión bibliográfica del tema.
 2. Existencia de otras aplicaciones o soluciones similares.
- Elaborar el diseño teórico de la investigación.
- Elaborar la propuesta de solución.
 1. Levantamiento de requisitos.
 2. Análisis y diseño de la aplicación.
 3. Elaboración de modelos

Objeto de Estudio:

Análisis, diseño e implementación de componentes, clases y ejercicios reutilizables que permita y facilite el desarrollo de productos multimedia en el Ambiente de Desarrollo Integrado (IDE) propuesto.

Campo de Acción:

Desarrollo de productos multimedia y software educativo en la UCI.

A continuación se dará una breve descripción sobre los capítulos que se trataran en el trabajo:

Capítulo 1: Se realiza una investigación detallada del estado de arte de las herramientas para la creación de multimedia, así como una introducción de los detalles y contenidos de las multimedia. Además se concretan las herramientas y metodologías a utilizar en nuestra propuesta a desarrollar.

Capítulo 2: Se realiza un bosquejo del entorno de desarrollo de la propuesta del sistema, dando al traste con sus requisitos funcionales así como la descripción de los diferentes casos de uso generados durante su análisis.

Capítulo 3: Se desarrolla el diseño de la propuesta partiendo de los patrones que guían su arquitectura. Además se promueve el uso de un manual del desarrollador para futuras mejoras. Se describen las fases de implementación.

Capítulo 4: Se hace un análisis minucioso de la factibilidad que posee la solución propuesta entre los datos que se destacan están el esfuerzo realizado, tiempo de desarrollo y el costo de producción.

Capítulo 1

Fundamentación Teórica

1.1 Introducción

Las Tecnologías de la Información y las Comunicaciones emergen como instrumentos novedosos, generándose una amplia gama de utilidades para éstas, siendo una de sus principales condiciones la de funcionar como eficaces herramientas para el desarrollo. La educación no ha sido ajena al impacto de las nuevas tecnologías de la información y las comunicaciones, en el campo de la educación se han venido desarrollando nuevas estrategias y formas para lograr un eficaz aprendizaje haciendo uso de las diferentes tecnologías. Las computadoras y sus métodos cada día más interactivos y variados de aprendizaje, la posibilidad del acceso al conocimiento global sin restricciones de tiempo, ni espacio; la dependencia tecnológica de la sociedad y las facilidades que esta brinda, así como todas las ventajas asociadas hacen que cada día el desarrollo de estos tipos de productos dirigidos a este sector en particular sean cada vez más cotizados y necesarios. Se han desarrollado herramientas para facilitar el aprendizaje del estudiantado, así como lograr un intercambio fluido de conocimientos ente alumnos y profesores, la incorporación de la informática en el campo educativo ha provocado el surgimiento de una nueva rama de la educación denominada Informática Educativa.

En este capítulo se tratarán aspectos de la informática educativa a nivel mundial así como su impacto e influencia en la educación cubana, el desarrollo de herramientas para fomentar y facilitar el uso de la informática en la educación, permitiendo el desarrollo favorable de la informática educativa. Se analizará además la informática en la educación como una herramienta, un material de apoyo para aprender con las computadoras, pero manteniendo siempre los principios básicos de la educación para el aprendizaje de los estudiantes. Los fines en la educación deben fundirse indisolublemente a la dimensión universal del hombre, tanto en lo individual como en su acción y vinculación con la sociedad. La informática en la educación, así como sus herramientas deben contribuir a la formación tanto profesional, como humanos de los estudiantes.

1.2 Informática Educativa en el Mundo

Las Tecnologías de la Informática y las Comunicaciones (TIC) juegan en la actual revolución científica técnica que se opera en el mundo un papel de primera importancia. Gracias a las máquinas computadoras se ha podido realizar un número considerable de actividades, que antes eran imposibles ejecutar. Internet juega dentro de los recursos informáticos un rol de primera línea, siendo hoy en día el medio por excelencia para la búsqueda de información sobre cualquier elemento del saber que se necesite.

Desde las pasadas décadas se ha ido intensificando notablemente la aplicación de la Informática en las diferentes esferas, lo que ha estado influenciado por el marcado desarrollo de la computación y más recientemente de las nuevas tecnologías de la información y la comunicación. El impacto de estas alcanza también a la educación, y es especialmente en este terreno es donde más deben emplearse los medios técnicos actualizados y capaces de mejorar la calidad de la enseñanza.

A la educación se le ha designado socialmente la función de transmitir y generar los conocimientos. Esta acción fomenta, a través de la investigación, al avance de la ciencia y el desarrollo tecnológico. Actualmente, la educación ha estado empleando las nuevas tecnologías de la información y la comunicación para apoyar la labor docente.

La informática se ha incorporado a los estudiantes y docentes con la finalidad de apoyar y mejorar los procesos de enseñanza y el aprendizaje a la cual le hemos denominado informática educativa.

La informática educativa consiste en el uso de estas tecnologías para educar a los alumnos de las instituciones educativas, para los programas de educación a distancia y de auto aprendizaje y para el entretenimiento personal de las empresas e instituciones que lo requieran.

La informática educativa puede emplearse como un apoyo para la enseñanza con la finalidad de estimular varios sentidos del sujeto que posibilite el aprendizaje. También puede ser el medio que permita la comunicación inmediata con el alumno. Es el vehículo a través del cual puede acercarse la acción docente a los estudiantes sin importar, fronteras, distancias o barrera. Nunca se sustituirá la labor del profesor porque la informática es una herramienta con la cual se puede operar el proceso educativo pero el diseño previo, la planificación y la estrategia a utilizar estará a cargo del experto educativo y del profesional. (Sergio Martínez Dunstan 2001)

1.3 Informática Educativa en Cuba

Con el triunfo de la revolución en 1959, nuestro Comandante en Jefe Fidel Castro junto con la máxima dirección del naciente estado, se dio a la tarea de dar un sentido más sólido al sistema social cubano. En la esfera de la educación, los enormes esfuerzos del Gobierno Revolucionario tuvieron como contenido esencial dar solución a los grandes problemas del pasado neocolonial, la reorganización y tecnificación del Ministerio de Educación y la toma de medidas inmediatas para eliminar el analfabetismo y garantizar la extensión de los servicios educacionales. Con el transcurso de los años la educación cubana ha ido tomando mucha más fuerza y solides se han obtenido grandes logros como son:

- La erradicación total del analfabetismo en la isla.
- El carácter gratuito y democrático de la educación en Cuba, lo que significa que todos, independientemente de su raza, sexo, credo religioso o procedencia social, tienen acceso a la educación.
- Garantizar la escolarización de todos los niños y jóvenes en edad escolar, y brindar facilidades a los jóvenes y adultos que están en disposición de continuar estudios especializados y superiores.
- Brindar ayuda a otras naciones mediante acuerdos gubernamentales en materia educacional, y en acciones específicas directas suscritas entre instituciones educacionales cubanas y homólogas extranjeras. (Luis Ignacio Gómez 2004)

La inserción de las Tecnologías Informáticas en la enseñanza tiene como objetivo general “ elevar la calidad de la Educación y garantizar la necesaria formación en las Tecnologías de la Información y las Comunicaciones (T.I.C.) de los ciudadanos cubanos mediante un proceso de educación continua” (Miguel Rodríguez Cuervo 2006), plasmado en el Programa de Informática Educativa del MINED de alcance nacional, con objetivos esenciales y una estructura para el programa que responda a las necesidades de la escuela cubana actual y permitir así que la computación llegue a todos los alumnos y profesores sin excepción, en el 100% de los centros educacionales de Cuba logrando una mayor formación informática del escolar cubano y una mejor aplicación de los software educativos.

El Centro de Informática y Comunicaciones, es la institución del Ministerio de Educación de la República de Cuba, encargada de diseñar, orientar y controlar las políticas nacionales

sobre el uso de las tecnologías de redes y comunicaciones, los servicios sobre ellas, así como el diseño, desarrollo y mantenimiento de sistemas de gestión y dirección para el sector de la educación.

Dicho centro tiene como funciones principales:

- Implantar, asesorar y controlar la aplicación de los principios por los cuales se rige la informatización de la Sociedad Cubana.
- Propiciar la utilización de las tecnologías de la información y las comunicaciones en el Sistema Nacional de Educación, ofreciendo diversos servicios que permiten la optimización y automatización del trabajo educacional mediante el empleo de las Nuevas Tecnologías de la Información y las Comunicaciones.
- Organiza la información generada por las distintas instituciones y universidades para su distribución virtual, desarrollando herramientas que facilitan el acceso a la misma.
- Garantiza el buen funcionamiento de recursos educativos como la Educación a Distancia, Bibliotecas Virtuales, Revistas Pedagógicas, entre otros. Velando además por la integridad y la seguridad de la Red de Servicios Telemáticos del Ministerio de Educación (RIMED).

Es por eso que se han llevado a cabo transformaciones para lograr una mayor utilización y dominio por la población de estas tecnologías y propiciar la ampliación de su cultura general integral, llevando esta ciencia a los distintos niveles de enseñanzas como una asignatura más y aprovechando sus posibilidades como medio y recurso para potenciar el aprendizaje.

1.4 Análisis de soluciones existentes

Cuadernia: Cuadernia es una herramienta de la Consejería de Educación y Ciencia de Castilla en España.

Es un desarrollo innovador generado en Castilla-La Mancha con grandes expectativas de futuro al cumplir con todos los estándares europeos y nacionales de la creación de contenidos educativos digitales.

Se trata de una herramienta fácil y funcional que nos permite crear de forma dinámica y visual "cuadernos digitales" que pueden contener información y actividades multimedia distribuibles a través de un navegador de Internet, por lo tanto independientes del Sistema Operativo. (Consejería de Educación y Ciencia 2008)

EdiLim: Este software o herramienta de autor para docentes está creado en Galicia por Fran Macías. Tiene un sitio web bastante completo (ofrece el software, tutorial de uso, ejemplos, está en varios idiomas,...). En dicha web se define como "entorno para la creación de materiales educativos, formado por un editor de actividades (EdiLim), un visualizador (LIM) y un archivo en formato XML (libro) que define las propiedades del libro y las páginas que lo componen".

Presenta un archivo ejecutable que no precisa instalación. Opcionalmente puede incluir una página HTML de ayuda con sus imágenes correspondientes. Posee un pequeño tamaño ya que permite utilizarlo desde un disco flexible. (Fran Macías 2006)

Entre sus ventajas se destacan:

- Entorno atractivo.
- Basado en la técnica de "arrastrar y soltar".
- Incluye editor de imágenes.
- Prepara los libros para su publicación en internet o distribución.

Requisitos:

- Microsoft Windows 98 o superior.

Atenex: Es una plataforma para la creación y gestión de materiales multimedia interactivas y para el seguimiento y evaluación del proceso de aprendizaje de nuestros alumnos caracterizada por:

Estructura **modular** que permite su crecimiento en cuanto a posibilidades. Se trata de una herramienta que acaba de nacer y está abierta a las sugerencias y demandas de la comunidad educativa para mejorar en facilidad de uso y funcionalidades.

- Es una herramienta para compartir, cuyos materiales son fácilmente **editables**, y preparada para su inclusión en otras plataformas de Aprendizaje Electrónico, ya que es compatible con los estándares **SCORM e IMS**.
- Es una herramienta multiplataforma, preparada para trabajar tanto en local como en línea.
- Su objetivo fundamental es la facilidad de uso: se pueden elaborar materiales por el simple procedimiento de arrastrar y soltar y e incorpora un completo plantillero de

actividades programadas de fácil configuración.

- Existen distintos perfiles de usuarios con permisos diferenciados, desde el nivel de administrador o coordinador hasta el de alumno o profesor autor.
- Adaptada a la diversidad, ya que cuenta con la posibilidad de plantear itinerarios diferenciados para los diversos ritmos de aprendizaje.
- Viene acompañada de un tutorial animado para facilitar los primeros pasos con la herramienta.
- Dispone de foros de ayuda y módulos de Preguntas Frecuentes como ayuda en línea.
- Dispone de un plantillero con 44 plantillas donde se encuentran un sinnúmero de actividades y juegos interactivos de carácter multimedia que pueden cargarse de manera sencilla en los distintos fotogramas.
- Los materiales realizados con la herramienta son completamente editables por parte de cualquier autor y sus archivos multimedia pueden ser aprovechados en otros materiales didácticos interactivos. (Junta de Extremadura 2007)

Existen otras tantas herramientas para el desarrollo de materiales educativos como son: Ardora, HotPotatoes, WebQuestion, Rayuela, Clic, Crossword... etc. Todas estas herramientas tratan la parte relacionada con los ejercicios, juegos interactivos o actividades de forma similar aunque son representadas de diversas formas, pero todas dan solución a una serie de tipologías como son:

1. Selección Simple.
2. Selección Múltiple.
3. Enlazar.
4. Completar.
5. Verdaderos o Falsos.
6. Crucigramas.
7. Ordenar.

Existen innumerables tipologías de ejercicios todos la que una mente humana sea capaz de crear, así como existen también muchas formas de visualizar o mostrar al los usuarios dichas tipologías. Esta tesis se centrara en las anteriormente mencionadas ya que después de un estudio de las diferentes soluciones existentes en el mundo moderno llegamos a la conclusión de son las más utilizadas y aceptadas por los usuarios.

Todas las herramientas que han visto en torno a software para la creación de material multimedia educativo cada vez son más potentes, atractivas, fáciles de usar, y responden a necesidades educativas. Todas las que se han nombrado (Cuadernia, EdiLim, Atenex, y otras también realizadas en todo el mundo representan un importante avance en ofrecer al profesorado instrumentos que les permitan realizar sus propios materiales didácticos.

1.5 Estrategia de investigación:

Se utilizará una *estrategia descriptiva*. Esta estrategia tiene como objetivo principal describir el fenómeno y reflejar lo esencial y más significativo del mismo, sin tener en cuenta las causas que lo originan, para lo que es necesario captar sus relaciones internas y regularidades, así como aquellos aspectos donde se revela lo general. Nuestro objetivo principal es describir el proceso de edición de un proyecto multimedia, reflejando lo esencial y más significativo del mismo en una herramienta libre. Es necesario consultar a profundidad el diseño teórico de la investigación para comprender el valor científico de los resultados obtenidos. (Rolando Alfredo Hernández León and Sayda Coello González 2002)

Métodos de investigación utilizados.

Métodos teóricos

- Histórico lógico
- Dialéctico
- Sistémico

Métodos empíricos

- La observación.
- La experimentación.

En cada caso se planteará el problema como un todo, donde los datos tomados de los proyectos, la propia dinámica de los procesos necesarios para desarrollo de estos proyectos y los roles y la alta interrelación entre todos estos elementos se funden en un sistema sostenible e integral.

Se dará al traste con el origen del problema a partir del método dialéctico, el cual nos provee de las contradicciones existentes y explica los cambios cualitativos que se producen en el sistema llevándonos a la necesidad imperante de crear un nuevo objeto.

Se tomará la problemática de la ingeniería de software en general desde un enfoque histórico lógico, en la primera parte de la investigación se desarrollará un estudio del estado del arte de la problemática analizada; se revisó las bondades y deficiencias de cada uno de los modelos, metodologías, y las tendencias en la resolución de esta problemática.

A través del método sistémico se determinó los componentes del objeto, estructura y jerarquía de cada uno, así como las relaciones entre ellos. Establecemos las leyes del sistema a partir de la dinámica y comportamiento del mismo.

Por otra parte se utilizarán métodos empíricos ya que estos describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional. El conocimiento de las relaciones contradictorias esenciales que caracteriza el comportamiento del objeto permite resolver el problema que dio inicio a la investigación.

Como métodos empíricos se utilizarán:

- Observación
- Experimentación

La observación pues de esta manera se puede obtener la información del comportamiento de nuestro objeto de investigación tal y como este se da en la realidad, es decir, es una forma de obtener información directa e inmediata sobre el fenómeno u objeto que está siendo investigado.

A través del método experimental esclarecimos las propiedades, leyes y relaciones del objeto así como la selección de las tecnologías de integración para la propuesta del nuevo sistema.

1.6 Metodología y lenguaje de modelado

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

Rumbaugh dio la siguiente definición:

“Una metodología de ingeniería de software es un proceso para la producción organizada del software, empleando para ello una colección de técnicas predefinidas

y convencionales en las notaciones. Una metodología se presenta normalmente como una serie de pasos, con técnicas y notaciones asociadas a cada paso. Los pasos de la producción del software se organizan normalmente en un ciclo de vida consistente en varias fases de desarrollo”.

(JACOBSON, IVAR BOOCH, and GRADY RUMBAUGH)

Para dar una idea de qué metodología podemos utilizar y cual se adapta más a nuestro medio, mencionaré tres de ellas de las que considero las más importantes, tal como: RUP, XP y MSF.

1.6.1 Rational Unified Process (RUP)

RUP está dividido en 4 fases.

- **Inicio**, El Objetivo en esta etapa es determinar la visión del proyecto.
- **Elaboración**, En esta etapa el objetivo es determinar la arquitectura óptima.
- **Construcción**, En esta etapa el objetivo es llevar a obtener la capacidad operacional inicial.
- **Transición**, El objetivo es llegar a obtener el release del proyecto.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, la cual consiste en reproducir el ciclo de vida en cascada a menor escala. Los Objetivos de una iteración se establecen en función de la evaluación de las iteraciones precedentes.

Vale mencionar que el ciclo de vida que se desarrolla por cada iteración, es llevada bajo dos disciplinas:

Disciplina de Desarrollo

4. **Ingeniería de Negocios:** Entendiendo las necesidades del negocio.
5. **Requerimientos:** Traslado de las necesidades del negocio a un sistema automatizado.
6. **Análisis y Diseño:** Traslado de los requerimientos dentro de la arquitectura de software.
7. **Implementación:** Creando software que se ajuste a la arquitectura y que tenga el comportamiento deseado.
8. **Pruebas:** Asegurándose que el comportamiento requerido es el correcto y

que todo lo solicitado está presente.

Disciplina de Soporte

- **Configuración y administración del cambio:** Guardando todas las versiones del proyecto.
- **Administrando el proyecto:** Administrando horarios y recursos.
- **Ambiente:** Administrando el ambiente de desarrollo.
- **Distribución:** Hacer todo lo necesario para la salida del proyecto

Esa metodología nos brinda un sin número de potencialidades y garantiza fundamentalmente el desarrollo de un software con calidad así como la organización en el trabajo. Como han podido darse cuenta está basada en iteraciones que van corrigiendo los errores encontrados en iteraciones anteriores, esto hace que el producto final salga con una indudable calidad y sea del agrado de cualquier cliente. (María A. Mendoza Sánchez 2004)

1.6.2 Extreme Programming (XP)

Es una de las metodologías de desarrollo de software muy exitosas en la actualidad utilizada para proyectos de corto plazo, equipos cortos. La metodología consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar al éxito del proyecto.

Características de XP

- **Pruebas Unitarias:** Se basa en las pruebas realizadas a los principales procesos, de tal manera que adelantándonos en algo hacia el futuro, podamos hacer pruebas de las fallas que pudieran ocurrir. Es como si nos adelantáramos a obtener los posibles errores.
- **Refabricación:** Se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio.
- **Programación en pares:** Una particularidad de esta metodología es que propone la programación en pares, la cual consiste en que dos desarrolladores participen en un proyecto en una misma estación de trabajo. Cada miembro lleva a cabo la acción que el otro no está haciendo en ese momento. (María A. Mendoza Sanchez 2004)

1.6.3 Microsoft Solution Framework (MSF)

Esta es una metodología flexible e interrelacionada con una serie de conceptos, modelos y prácticas de uso, que controlan la planificación, el desarrollo y la gestión de proyectos tecnológicos. MSF se centra en los modelos de proceso y de equipo dejando en un segundo plano las elecciones tecnológicas.

MSF tiene las siguientes características:

- **Adaptable:** Es parecido a un compás, usado en cualquier parte como un mapa, del cual su uso es limitado a un específico lugar.
- **Escalable:** Puede organizar equipos tan pequeños entre 3 o 4 personas, así como también, proyectos que requieren 50 personas a más.
- **Flexible:** Es utilizada en el ambiente de desarrollo de cualquier cliente.
- **Tecnología Agnóstica:** Porque puede ser usada para desarrollar soluciones basadas sobre cualquier tecnología.

(María A. Mendoza Sánchez 2004)

Después de haber hecho un estudio de estas metodologías tan exitosas en el mundo se decide utilizar la metodología RUP para el desarrollo, ya que se ajusta más a la hora de modelar el sistema, además de ser la única entre las metodologías estudiadas que garantiza la elaboración de las fases de un producto de software orientado a objetos y brinda nuevos artefactos permitiendo modelar el sistema de forma correcta, también genera la documentación necesaria para futuras actualizaciones o revisiones del producto, aspecto muy importante ya que el presente trabajo forma parte de un proyecto investigativo al cual se le dará seguimiento en el próximo curso.

1.7 ¿Por qué UML como lenguaje de modelado?

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.

Brinda 13 diferentes diagramas que nos dan una mejor visión, organización y entendimiento del software o sistema a desarrollar.

Además UML es un estándar industrial promovido por el grupo OMG al mismo nivel que el estándar CORBA para intercambio de objetos distribuidos. Para la revisión de UML se formaron dos "corrientes" que promovían la aparición de la nueva versión desde distintos puntos de vista. Finalmente se impuso la visión más industrial frente a la académica. Recientemente se ha publicado la versión 2.0 en la que aparecen muchas novedades y cambios que, fundamentalmente, se centran en resolver carencias prácticas. Además, esta versión recibe diversas mejoras que provienen del lenguaje SDL.

Por todo lo antes expuesto se decidió tomar como metodología de desarrollo **RUP** y como lenguaje de modelado **UML**.

1.8 Herramienta case: Visual Paradigm

Visual Paradigm fácil de usar y de soporte multiplataforma, proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Visual Paradigm fue creado para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación, también proporciona características tales como generación del código, ingeniería inversa y generación de informes.

Está diseñado para usuarios interesados en sistemas de software de gran escala con el uso del acercamiento orientado a objeto, además apoya los estándares más recientes de las notaciones de Java y de UML. A pesar de todas estas posibilidades la licencia es muy restringida y las imágenes y reportes generados, no son de muy buena calidad.

Existen otras herramientas case en este gran mundo del software libre como son: Rational Software Architect (RSA) y Umbrello UML Modeller:

El **RSA** es una magnífica herramienta case que permite construir, integrar, ampliar, modernizar y desplegar software. Con amplia ayuda para todos los roles y actividades en el ciclo de vida del software esta la plataforma apoya el desarrollo para Linux. La única dificultad que se le encontró al uso de dicha herramienta es las grandes prestaciones de hardware que requiere, estas son:

Procesador - Mínimo: Pentium™ 3, 800 Mhz; Recomendado: Pentium™ 4, 1.4 GHz

Sistema Operativo: Windows 2000, XP y Linux.

Memoria mínima: 768 MB; 1 GB de memoria RAM recomendada.

Espacio del disco requerido: 3 GB; 6 GB se requiere al instalar cuando se descarga.

Otra herramienta para implementar es el **Umbrello UML Modeller** ayuda en el proceso de desarrollo de software usando el estándar UML (Unified Modelling Language), su propósito principal es el de ayudar en el análisis y diseño de los sistemas, pero Umbrello no es lo suficientemente robusta para un desarrollo de productos educativos en formato multimedia.

1.9 Definición del lenguaje de codificación.

Como lenguaje de codificación fue escogido ActionScript 2.0. La principal función de ActionScript 2.0 es un modelo común para crear programas orientados a objetos. También implementa varios nuevos conceptos y palabras clave de programación orientada a objetos, como por ejemplo *clase*, *interfaz* y *paquete*, con los que estará familiarizado si ha programado alguna vez en Java u otro lenguaje con esta filosofía.

Pero la principal razón por la que fue escogido es porque el IDE a desarrollar utiliza como compiladores principales **MTASC** y **HAXE**, estos compiladores están orientados principalmente a la creación de películas flash cuya utilización principal es en la creación de multimedia.

1.10 Herramientas de desarrollo.

En estos momentos tanto Cuba como el resto de los países está experimentando este gran mundo del software libre pero aun nos falta mucho por ganar a tal punto que no existe o no se conoce ninguna herramienta libre que permita el desarrollo en ActionScript. Para darle una solución a este problema se asumió como herramienta de desarrollo FlashDevelop y como compilador MTASC o HAXE.

1.10.1 FlashDevelop: Es un entorno de desarrollo para Flash ActionScript 2 (AS2) y ActionScript 3 (AS3). Está orientado a la programación en ActionScript, pero que no se limita a este lenguaje. También soporta JavaScript, HTML, XML y CSS. Es un entorno de desarrollo completo con todas sus funcionalidades: buscar y reemplazar, buscar en varios ficheros, debug, validación del código, etc. Además FlashDevelop soporta multitud de plugins que lo hacen aún más funcional.

Es una aplicación de código abierto y gratis. Posiblemente sea la mejor aplicación de código abierto que sirve para programar en AS2. Incluso es superior a otros programas comerciales del mismo tipo. (2007)

1.11 Librería para el desarrollo AsWing

AsWing es un framework de código abierto para Flash ActionScript 2.0 y permite a los programadores realizar aplicaciones utilizando una gran gama de clases y componentes con gran variedad de estilos y formas. (Asociación javaHispano 2007)

1.12 Conceptos asociados al problema

SCORM (Sharable Content Object Reference Model): Es un modelo conceptual que describe cómo administrar, empaquetar y entregar información de aprendizaje para que sea fácilmente distribuibles en Internet. (Juan David Soto 2007)

IMS: Sistema de Gestión Instruccional. Consorcio de aprendizaje global / Coalición de organizaciones gubernamentales dedicadas a definir y distribuir especificaciones de interoperabilidad de arquitectura abierta para productos de tele formación.

Tipologías: Tipo específico de determinado elemento en este caso una tipología de ejercicio es un tipo de ejercicio específico como Verdadero o Falso, Crucigrama, etc.

OMG: Consorcio dedicado al cuidado y el establecimiento de diversos estándares de tecnologías orientadas a objetos.

CORBA: Estándar que establece una plataforma de desarrollo de sistemas distribuidos facilitando la invocación de métodos remotos bajo un paradigma orientado a objetos.

1.13 Conclusiones del Capítulo

En este capítulo se ha realizado un análisis general del diseño teórico del trabajo de diploma, esclareciendo conceptos asociados al dominio del problema. Además se ha ofrecido un análisis de las soluciones existentes para el problema en cuestión, siendo estas ineficientes y dejando fundamentada la necesidad de una nueva solución. Queda así fundamentada la finalidad de nuestro subsistema. Estudiado este capítulo queda orientada además la guía de desarrollo de nuestro trabajo de diploma y sentadas las bases para un posterior análisis de la propuesta del sistema y específicamente del subsistema de edición visual para los recursos multimedia.

Capítulo 2

Descripción de la Solución Propuesta

2.1 Introducción.

En el presente capítulo se realiza la descripción de la solución propuesta a través del Lenguaje Orientado a Objetos para el Modelado de Aplicaciones (UML) como extensión del Lenguaje Unificado de Modelado. Haciendo énfasis en los conceptos asociados, además de los requisitos funcionales. Se realiza la vista de casos de uso del sistema, justificándose el actor y describiéndose detalladamente cada caso de uso.

2.2 Descripción del Modelo de Dominio.

El Proceso Unificado de Desarrollo está basado en componentes y tiene como características principales: guiado por casos de uso, centrado en la arquitectura e iterativo e incremental. Utiliza el lenguaje UML (lenguaje unificado de modelado) para preparar todos los esquemas de un sistema de software. De hecho, UML es una parte esencial de RUP, ya que sus desarrollos fueron paralelos.

En este caso debido a la poca estructuración de los procesos de negocio se plantea un modelo de dominio para lograr una mejor comprensión de los conceptos del sistema. Para esto se realiza la descripción del modelo del dominio a través de un diagrama de clases UML, en el cual se definen las principales clases conceptuales que intervienen en el sistema.

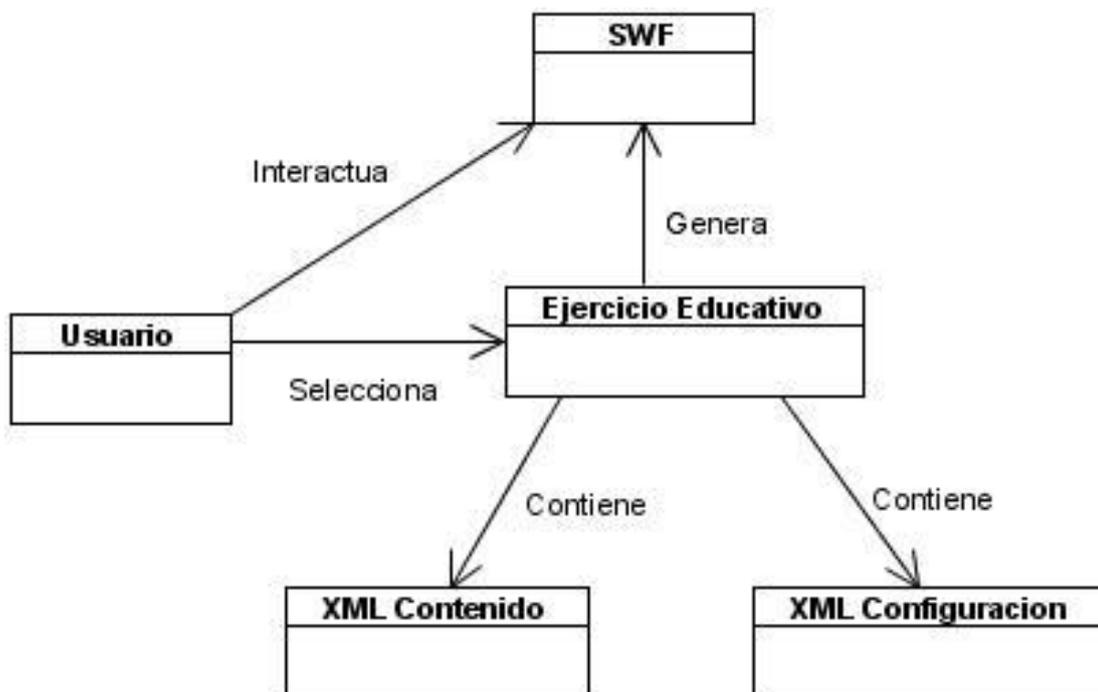


Figura2.1: Modelo de dominio

2.3 Solución Propuesta

La solución propuesta es la elaboración de un conjunto de clases, componentes o ejercicios reutilizables, todo esto representan un modulo del IDE para múltiples plataformas con tecnologías y herramientas libres para desarrollar software educativo en formato multimedia que es el objetivo principal del trabajo en cuestión.

2.3.1 Requisitos Funcionales

R1. Crear Ejercicio

R1.1 El sistema le brindará la posibilidad de indicar el tipo de ejercicio a crear.

R1.2 El sistema tendrá definido previamente el camino del XML de contenido del ejercicio y del XML de configuración.

R1.3 El sistema generará el ejercicio según la selección del usuario, los ejercicios a crear pueden ser:

- Selección Múltiple con Componentes
- Selección Simple con Selección
- Selección Simple con Componentes
- Verdadero o Falso

- Ordenar Palabras
- Ordenar Frases
- Enlazar con Selección
- Enlazar con Arrastre
- Sopa de Letras
- Crucigrama
- Acróstico

R1.4 El sistema creará un ID para cada ejercicio creado.

R2. Configurar Ejercicios

R2.1 El sistema permitirá definir la posición y tamaño del enunciado en el escenario.

R2.2 El sistema permitirá definir la ruta al CSS con los estilos que se aplicara a todo el contenido del ejercicio.

R2.3 El sistema permitirá definir si el ejercicio es Evaluativo o no.

R2.4 El sistema permitirá definir el color para la ventana que mostrará el mensaje en caso de que el ejercicio no sea evaluativo o el resultado en caso de que sea evaluativo.

R2.5 El sistema permitirá definir las dimensiones de la ventana.

R2.6 El sistema permitirá definir el color del borde de la ventana

R3. Cargar Contenido

R3.1 El sistema cargara y visualizara los siguientes elementos.

- Enunciado del ejercicio.
- Pregunta del ejercicio.
- Respuesta del ejercicio.
- Mensaje a mostrar en caso de que el ejercicio no sea evaluativo.

R4. Configurar Ejercicio Selección Múltiple

R4.1 El sistema permitirá definir sub-tipologías de ejercicios Selección Múltiple.

R4.1.1 El sistema permitirá definir la sub-tipología Selección Múltiple con Componentes (ComboBox, CheckBox).

- Tamaño del componente a utilizar (CheckBox).
- Tipo de media a utilizar, en caso de que sea una imagen debe especificarse :
 - Dimensiones de la imagen (alto y ancho).
 - URL de donde se encuentra la misma.
 - Ubicación de la media en el escenario.

R4.1.2 El sistema permitirá definir la sub-tipología Selección Múltiple con Selección.

- Tamaño y posición del enunciado dentro del ejercicio.
- Estado o estilo de los elementos (palabras, frases, letras e imágenes) a seleccionar.

R4.2 El sistema permitirá seleccionar más de un elementos del escenario.

R5. Configurar Ejercicio Selección Simple

R5.1 El sistema permitirá definir sub-tipologías de ejercicios Selección Simple.

R5.1.1 El sistema permitirá definir la sub-tipología Selección Simple con Componentes.

- Tamaño del componente a utilizar (RadioButton).
- Tipo de media a utilizar, en caso de que sea una imagen debe especificarse :
 - . Dimensiones de la imagen (alto y ancho).
 - . URL de donde se encuentra la misma.
 - Ubicación de la media en el escenario.

R5.1.2 El sistema permitirá definir la sub-tipología Selección Simple con Selección.

- Tamaño y posición del enunciado dentro del ejercicio.
- Estado o estilo de los elementos (palabras, frases, letras e imágenes) a seleccionar.

R5.2 El sistema permitirá seleccionar un solo elementos del escenario.

R6. Configurar Ejercicio Enlazar

R6.1 El sistema permitirá definir sub-tipologías de ejercicios Enlazar.

R6.1.1 El sistema permitirá definir la sub-tipología Enlazar con Componentes.

- El sistema permitirá definir el tamaño del componente a utilizar (Combobox).

- El sistema permitirá definir la ubicación del componente en el escenario.
- El sistema permitirá definir la ubicación de la pregunta asociada al componente.
- El sistema permitirá definir el tipo de media a utilizar, en caso de que sea una imagen debe especificarse :
 - Dimensiones de la imagen (alto y ancho).
 - URL de donde se encuentra la misma

R6.1.2 El sistema permitirá definir la sub-tipología Enlazar con Selección.

- El sistema permitirá definir el tipo de media a utilizar, en caso de que sea una imagen debe especificarse :
 - Dimensiones de la imagen (alto y ancho).
 - URL de donde se encuentra la misma
- El sistema permitirá definir la ubicación de la media en el escenario.
- El sistema permitirá definir el formato que se aplicará cuando la media se seleccione, en caso de las imágenes se indicara el alpha que tomará, y en el caso de los textos el color del mismo.
- El sistema permitirá definir el estado del elemento cuando esta seleccionado (tipo de letra, tamaño, color).

R6.2.1 El sistema permitirá definir la sub-tipología Enlazar con Arrastre.

- El sistema permitirá definir el tipo de media a utilizar, en caso de que sea una imagen debe especificarse :
 - Dimensiones de la imagen (alto y ancho).
 - URL de donde se encuentra la misma
- El sistema permitirá definir la ubicación de la media en el escenario.
- El sistema permitirá definir el color de la flecha a utilizar para la unión.
- El sistema permitirá definir el grosor de la flecha.

R7. Configurar Completar

R7.1 El sistema debe permitir definir sub-tipologías de ejercicios Completar.

R7.1.1 El sistema debe permitir definir la sub-tipología Completar por Componente.

R7.1.1.1 El sistema deberá permitir definir el tamaño del componente a utilizar (Combobox).

R7.1.1.2 El sistema deberá permitir definir la ubicación del componente en el escenario.

R7.1.1.3 El sistema deberá permitir definir el tamaño y la posición del enunciado del ejercicio.

R7.1.1.4 El sistema deberá permitir definir el tamaño y el color de la ventana de resultados.

R7.1.2 El sistema debe permitir definir la sub-tipología Completar por Selección.

R7.1.2.1 El sistema debe permitir definir el tamaño y posición del enunciado del ejercicio.

R7.1.2.2 El sistema debe permitir definir el estado o estilo y posición de las palabras antes de ser seleccionadas.

R7.1.2.3 El sistema debe permitir definir el estado o estilo y posición de las palabras después de ser seleccionadas.

R7.1.2.4 El sistema debe permitir definir el tamaño y color de la ventana de resultados.

R8. Configurar Verdadero o Falso

R8.1 El sistema permitirá que el ejercicio incluya dichas configuraciones:

- Tamaño del componente a utilizar (Combobox).
- Ubicación del componente en el escenario.
- Ubicación de la pregunta asociada al componente.
- Tamaño y posición del enunciado del ejercicio.
- Tamaño y posición del botón Revisar del ejercicio.
- En caso de que sea una imagen se guardará el tamaño y posición.

R9. Configurar Crucigrama

R9.1 El sistema permitirá definir el tamaño de las celdas (alto y ancho)

R9.2 El sistema permitirá definir el color de las celdas y del cursor.

R9.3 El sistema permitirá definir las palabras que conformarán la respuesta del ejercicio

R9.4 El sistema permitirá definir de cada palabra: dirección (vertical (V) u horizontal (H)), fila, columna, cantidad de celdas, posición inicial de la primera celda (x,y).

R10. Configurar Ordenar

R10.1 El sistema mostrará los elementos aleatoriamente.

R10.2 El sistema permitirá definir el orden de los elementos.

R10.3 El sistema permitirá definir sub-tipologías de ejercicios Ordenar.

R10.3.1 El sistema permitirá definir la sub-tipología Ordenar con Componentes.

- El sistema permitirá definir el tamaño del componente a utilizar (Combobox).
- El sistema permitirá definir la ubicación del componente en el escenario.
- El sistema permitirá definir la ubicación de la pregunta asociada al componente.
- El sistema permitirá definir el tamaño y posición del enunciado del ejercicio.
- El sistema permitirá definir el tamaño y posición del botón Revisar del ejercicio.
- El sistema permitirá definir en caso de que sea una imagen se guardara la tamaño y posición que ocupara dentro del escenario.
- El sistema permitirá definir el tamaño y color de la ventana de resultados.

R10.3.2 El sistema permitirá definir la sub-tipología Ordenar Secuencias de Frases.

- El sistema permitirá definir el tamaño y posición del enunciado del ejercicio.
- El sistema permitirá definir la posición de las frases a organizar dentro del escenario.
- El sistema permitirá definir el estado o estilo de las frases seleccionadas para ordenar.
- El sistema permitirá definir el tamaño y color de la ventana de resultados.

R10.3.3 El sistema permitirá definir la sub-tipología Ordenar Palabras en una Frase.

- El sistema permitirá definir el tamaño y posición del enunciado del ejercicio.
- El sistema permitirá definir la posición de las palabras a organizar dentro del escenario.
- El sistema permitirá definir el estado o estilo de las palabras seleccionadas para ordenar.
- El sistema permitirá definir el tamaño y color de la ventana de resultados.

R11 Configurar Sopa de Letras

R11.1 El sistema permitirá definir el tamaño de las celdas (alto y ancho)

R11.2 El sistema permitirá definir el color de las celdas antes de encontrar una palabra.

R11.3 El sistema permitirá definir el color de las celdas que conforman una palabra encontrada.

R11.4 El sistema permitirá definir las palabras que conformarán la respuesta del ejercicio

R11.5 El sistema permitirá definir de cada palabra: dirección (vertical (V), horizontal (H), diagonal (D)), fila, columna, cantidad de celdas, posición inicial de la primera celda (x,y).

R12 Verificar Ejercicio

R12.1 El sistema permitirá al usuario introducir una posible respuesta.

R12.2 En dependencia de la tipología de ejercicio que sea el sistema verificara si la respuesta es correcta o no.

R12.3 En caso de ser correcta el sistema verificara el tipo de ejercicio que se está trabajando (Evaluativo o de Retroalimentación).

R12.4 En caso de ser Evaluativo se emitirá una respuesta con la calificación obtenida sino un mensaje especificado por el usuario.

2.3.2 Requerimientos no funcionales de Restricciones en la implementación.

- El lenguaje de programación que se utilizará va a ser el ActionScript 2.0.

2.3.3 Requerimientos no funcionales de Usabilidad.

- Los usuarios a utilizar el sistema deberán tener algún conocimiento previo en trabajo con sistemas operativos visuales y la utilización de los dispositivos externos a la computadora personal que sustentan este modo de trabajo.
- También deberán tener algún conocimiento previo sobre el trabajo con ficheros XML y del lenguaje ActionScript.

2.3.4 Modelo de Casos de Uso del Sistema

El modelado de **casos de uso** es una técnica para la captura de requisitos potenciales de un nuevo sistema o una actualización software. Cada caso de uso proporciona uno o más escenarios que indican cómo debería interactuar el sistema con el usuario o con otro sistema para conseguir un objetivo específico. Utilizando las facilidades que nos brinda el lenguaje 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.

Determinación y justificación de los actores del sistema.

Actor / Usuario – representa a una persona que interactúa con al multimedia directamente como un usuario más de la misma.

DIAGRAMA DE CASOS DE USO DEL SISTEMA.

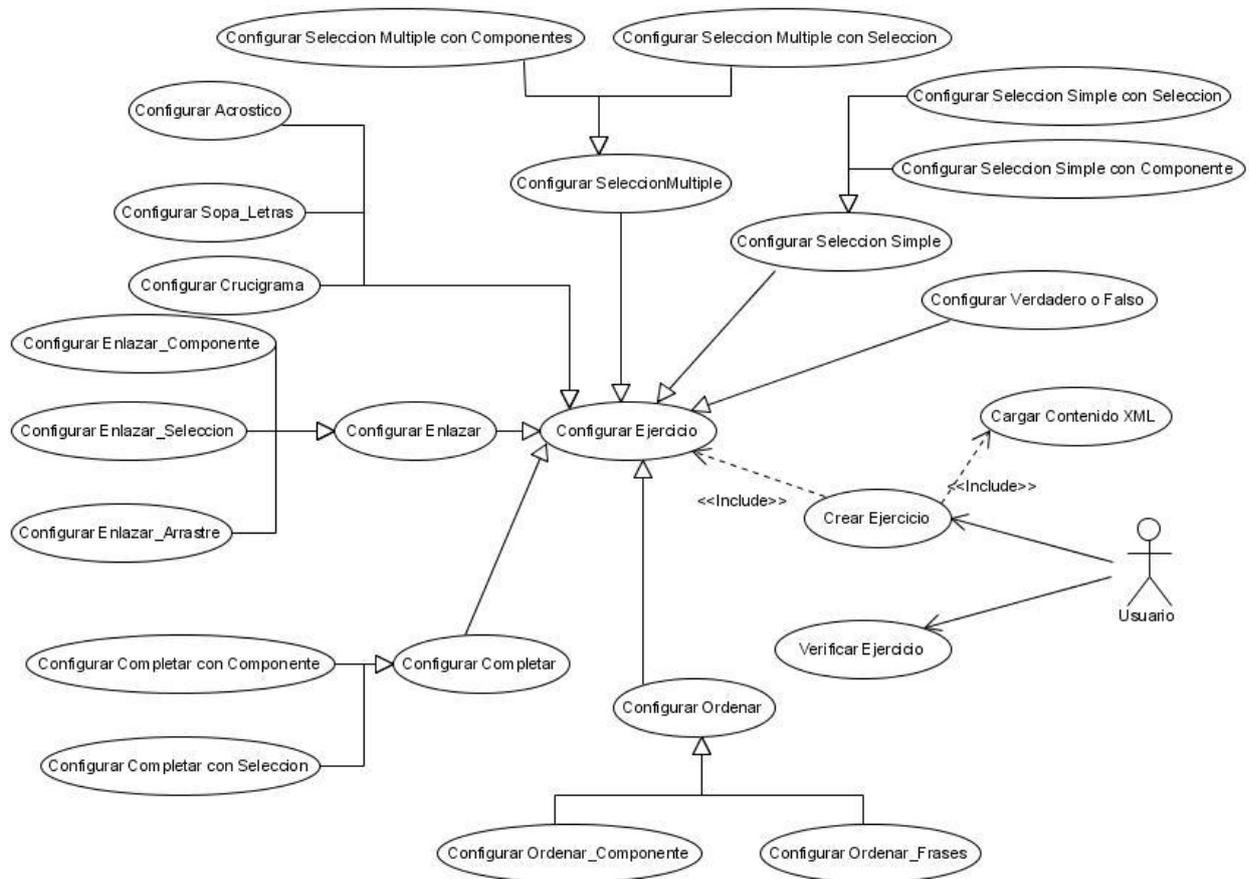


Figura2.2: Diagrama de Casos de Uso del Sistema

2.3.5 Descripción y expansión de los casos de uso.

Casos de Uso <Crear Ejercicios>

Caso de Uso: Crear Ejercicios	
Actores:	Usuario
Resumen:	<p>El actor selecciona la opción Crear Ejercicios, el sistema le brindará la posibilidad de indicar el tipo de ejercicio a crear, el camino del XML de contenido del ejercicio y del XML de configuración. Se invoca el caso de uso Configurar Ejercicio y el caso de uso Cargar Contenido. El sistema generará el ejercicio según la selección del usuario, los ejercicios a crear pueden ser:</p> <ul style="list-style-type: none"> ● Selección Múltiple con Componentes ● Selección Simple con Selección ● Selección Simple con Componentes ● Verdadero o Falso ● Ordenar Palabras ● Ordenar Frases ● Enlazar con Selección ● Enlazar con Arrastre ● Sopa de Letras ● Crucigrama ● Acróstico <p>El sistema creará un ID para cada ejercicio creado.</p>
Referencia:	R1
CU asociados:	
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio y debe existir un XML con la configuración del mismo.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Configurar Ejercicio>

Caso de Uso: Configurar Ejercicios	
Actores:	Usuario
Resumen:	<p>El actor selecciona configurar ejercicio, para ello debe crear, en caso de que no esté creado, un archivo XML, o editar uno ya creado. En este archivo XML se configurara el ejercicio. El XML debe tener una determinada estructura para un correcto funcionamiento del ejercicio y debe ser guardado con el nombre configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Posición del enunciado en el escenario. ● Ruta a la CSS con los estilos que se aplicara a todo el contenido del ejercicio. ● Si el ejercicio es Evaluativo o no. ● Color para la ventana que mostrará el mensaje en caso de que el ejercicio no sea evaluativo o el resultado en caso de que sea evaluativo. ● Dimensiones de la ventana. ● Color del borde de la ventana ● Color de la ventana
Referencia:	R2
CU asociados:	Crear Ejercicio
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Cargar Contenido>

Caso de Uso: Cargar Contenido	
Actores:	Usuario
Resumen:	<p>Este Caso de Uso es invocado por el Caso de Uso <i>Crear Ejercicios</i>, para ello el actor debe crear, en caso de que no este creado, un archivo XML, o editar uno ya creado. Este archivo XML contendrá el contenido del ejercicio. El XML debe tener una determinada estructura para un correcto funcionamiento del ejercicio y debe ser guardado con el nombre Contenido.</p> <ul style="list-style-type: none"> ● Pregunta del ejercicio. ● Respuesta del ejercicio. ● Enunciado del ejercicio. ● Mensaje a mostrar en caso de que el ejercicio no sea evaluativo
Referencia:	R3
CU asociados:	Crear Ejercicio
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Prioridad:	Crítico

Casos de Uso <Verificar Ejercicio>

Caso de Uso: Verificar Ejercicio	
Actores:	Usuario
Resumen:	<p>El actor selecciona la opción Verificar Ejercicio.</p> <p>El actor introduce una posible respuesta para la solución del ejercicio en dependencia de la tipología de ejercicio que el seleccione. El sistema procede a la verificación de la misma y emite un resultado, en caso de que el ejercicio no sea evaluativo se mostrará un mensaje educativo al usuario; en caso que sea evaluativo se le mostrará el</p>

	resultado obtenido en dicho ejercicio.
Referencia:	R12
CU asociados:	
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Prioridad:	Crítico

Casos de Uso <Configurar Verdadero o Falso>

Caso de Uso: Configurar Verdadero o Falso	
Actores:	Usuario de la Multimedia
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Configurar Verdadero o Falso y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tamaño del componente a utilizar (Combobox). ● Ubicación del componente en el escenario. ● Ubicación de la pregunta asociada al componente. ● Tamaño y posición del botón Revisar del ejercicio. ● En caso de que sea una imagen se guardara la tamaño y posición.
Referencia:	R8
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Configurar Selección Múltiple con Componentes>

Caso de Uso: Configurar Selección Múltiple con Componentes	
Actores:	Usuario

Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Configurar Selección Múltiple con Componentes y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tamaño del componente a utilizar (CheckBox). ● Tipo de media a utilizar, en caso de que sea una imagen debe especificarse : <ul style="list-style-type: none"> . Dimensiones de la imagen (alto y ancho). . URL de donde se encuentra la misma ● Ubicación de la media en el escenario.
Referencia:	R4
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Configurar Selección Simple con Componentes>

Caso de Uso: Configurar Selección Simple con Componentes	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Configurar Selección Simple con Componentes y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tamaño del componente a utilizar (RadioButton). ● Tipo de media a utilizar, en caso de que sea una imagen debe especificarse : <ul style="list-style-type: none"> ● . Dimensiones de la imagen (alto y ancho). ● . URL de donde se encuentra la misma. ● Ubicación de la media en el escenario.

Referencia:	R5
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Configurar Selección Simple con Selección>

Caso de Uso: Configurar Selección Simple con Selección	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Configurar Selección Simple con Selección y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tamaño y posición del enunciado dentro del ejercicio. ● Estado o estilo de los elementos (palabras, frases, letras e imágenes) a seleccionar. ● Seleccionar un solo elementos del escenario.
Referencia:	R5
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Configurar Selección Múltiple con Selección>

Caso de Uso: Configurar Selección Múltiple con Selección	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Configurar Selección Múltiple con Selección y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> • Tipo de media a utilizar, en caso de que sea una imagen debe especificarse : <ul style="list-style-type: none"> . Dimensiones de la imagen (alto y ancho). . URL de donde se encuentra la misma • Ubicación de la media en el escenario. • Formato que se aplicará cuando la (s) media (s) se seleccione (n), en caso de las imágenes se indicara el alpha que tomará, y en el caso de los textos el color del mismo.
Referencia:	R5
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Ordenar Frases>

Caso de Uso: Ordenar Frases	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Ordenar Frases y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> • Tamaño y color de la ventana de resultados. • Tamaño y posición del enunciado del ejercicio. • Posición de las frases a organizar dentro del escenario. • Estado o estilo de las frases seleccionadas para ordenar.
Referencia:	R10
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Ordenar por Componente>

Caso de Uso: Ordenar por Componente	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Ordenar por Componentes y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> • Tamaño del componente a utilizar (Combobox). • Ubicación del componente en el escenario. • Ubicación de la pregunta asociada al componente.

	<ul style="list-style-type: none"> • Tamaño y posición del botón Revisar del ejercicio. • En caso de que sea una imagen se guardara el tamaño y posición que ocupara dentro del escenario. • Tamaño y color de la ventana de resultados.
Referencia:	R10
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Completar por Componente>

Caso de Uso: Completar por Componente	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Completar por Componente y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> • Tamaño del componente a utilizar (Combobox). • Ubicación del componente en el escenario.
Referencia:	R7
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Completar con Selección>

Caso de Uso: Completar con Selección	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Completar con Selección y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> • Tamaño y posición del enunciado del ejercicio. • Estado o estilo y posición de las palabras antes de ser seleccionadas. • Estado o estilo y posición de las palabras después de ser seleccionadas. • Tamaño y color de la ventana de resultados.
Referencia:	R7
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Enlazar con Componentes>

Caso de Uso: Enlazar con componentes	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Enlazar con Componentes y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> • Tamaño del componente a utilizar (Combobox). • Ubicación del componente en el escenario. • Ubicación de la pregunta asociada al componente.

	<ul style="list-style-type: none"> ● Tipo de media a utilizar, en caso de que sea una imagen debe especificarse : <ul style="list-style-type: none"> . Dimensiones de la imagen (alto y ancho). . URL de donde se encuentra la misma
Referencia:	R6
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Enlazar con Arrastre>

Caso de Uso: Enlazar con Arrastre	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Enlazar con Arrastre y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tipo de media a utilizar, en caso de que sea una imagen debe especificarse : <ul style="list-style-type: none"> . Dimensiones de la imagen (alto y ancho). . URL de donde se encuentra la misma ● Ubicación de la media en el escenario. ● Color de la flecha a utilizar para la unión. ● Grosor de la flecha.

Referencia:	R6
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Enlazar con Selección>

Caso de Uso: Enlazar con Selección	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de Enlazar con Selección y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tipo de media a utilizar, en caso de que sea una imagen debe especificarse : <ul style="list-style-type: none"> . Dimensiones de la imagen (alto y ancho). . URL de donde se encuentra la misma ● Ubicación de la media en el escenario. ● Formato que se aplicará cuando la media se seleccione, en caso de las imágenes se indicara el alpha que tomará, y en el caso de los textos el color del mismo.
Referencia:	R6
CU asociados:	Configurar Ejercicios (Include)

Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Sopa de letra>

Caso de Uso: Configurar Sopa de letra	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de ejercicio Sopa de Letras y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tamaño de las celdas (alto y ancho) ● Color de las celdas antes de encontrar una palabra. ● Color de las celdas que conforman una palabra encontrada. ● Palabras que conformarán la respuesta del ejercicio ● De las palabras incluirá: dirección (vertical (V), horizontal (H), diagonal (D)), fila, columna. ● Cantidad de celdas. ● Posición inicial de la primera celda (x; y).
Referencia:	R11
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

Casos de Uso <Crucigrama>

Caso de Uso: Crucigrama	
Actores:	Usuario
Resumen:	<p>Este caso de uso comienza cuando el desarrollador selecciona la tipología de ejercicio Crucigrama y define el camino al XML de configuración. Las configuraciones del ejercicio incluirán:</p> <ul style="list-style-type: none"> ● Tamaño de las celdas (alto y ancho) ● Color de las celdas ● Color del cursor ● Palabras que conformarán la respuesta del ejercicio ● De las palabras incluirá: dirección (vertical (V) u horizontal (H)), fila, columna. ● Cantidad de celdas. ● Posición inicial de la primera celda (x; y).
Referencia:	R9
CU asociados:	Configurar Ejercicios (Include)
Precondiciones:	Debe existir un XML con el contenido de dicho ejercicio.
Poscondiciones:	
Prioridad:	Crítico

2.4 Conclusiones del capítulo

En el reciente capítulo se presentó toda la información referida a la solución propuesta del sistema, así como el modelo del dominio del entorno perteneciente al problema que da solución el conjunto de clases, componentes o ejercicios en cuestión, los requerimientos del sistema como base para la modelación de los casos de usos del mismo. Además se realizó la descripción de cada caso de uso.

Construidos todos los artefactos antes mencionados se culmina con los flujos de trabajo: modelamiento del dominio y requerimientos, concluyéndose así la fase de inicio comprendida dentro del ciclo de vida del desarrollo del software y tratando de que se cumplan todos los requisitos y las funciones que se han considerado necesarias en este capítulo, para luego continuar con el análisis y diseño.

Capítulo 3

Diseño e Implementación

3.1 Introducción

En este capítulo se modelan los principales artefactos que ayudan a manejar las dificultades que implican la construcción de la aplicación. También daremos un acercamiento más detallado a nuestro sistema, así como a sus clases y librerías que se utilizan. También se abordarán brevemente los patrones de diseño y arquitectura que se utilizarán, los principios del diseño y finalizando se presentarán el modelo de implementación mediante el diagrama de despliegue que resultó del diseño realizado de cada uno de los casos de uso del sistema.

3.2 Diseño de la aplicación

El diseño es el centro de atención de la propuesta de solución, el mismo contribuye a una arquitectura estable y sólida. En el diseño se modela el sistema y de forma incluida la arquitectura para que soporte todos los requisitos.

3.2.1 Diagrama de Clases del Diseño:

Es una visión mucho mas detalla de las clases que intervienen en el sistema así como sus atributos, métodos, tipos de datos y las relaciones que entre ellas existe.

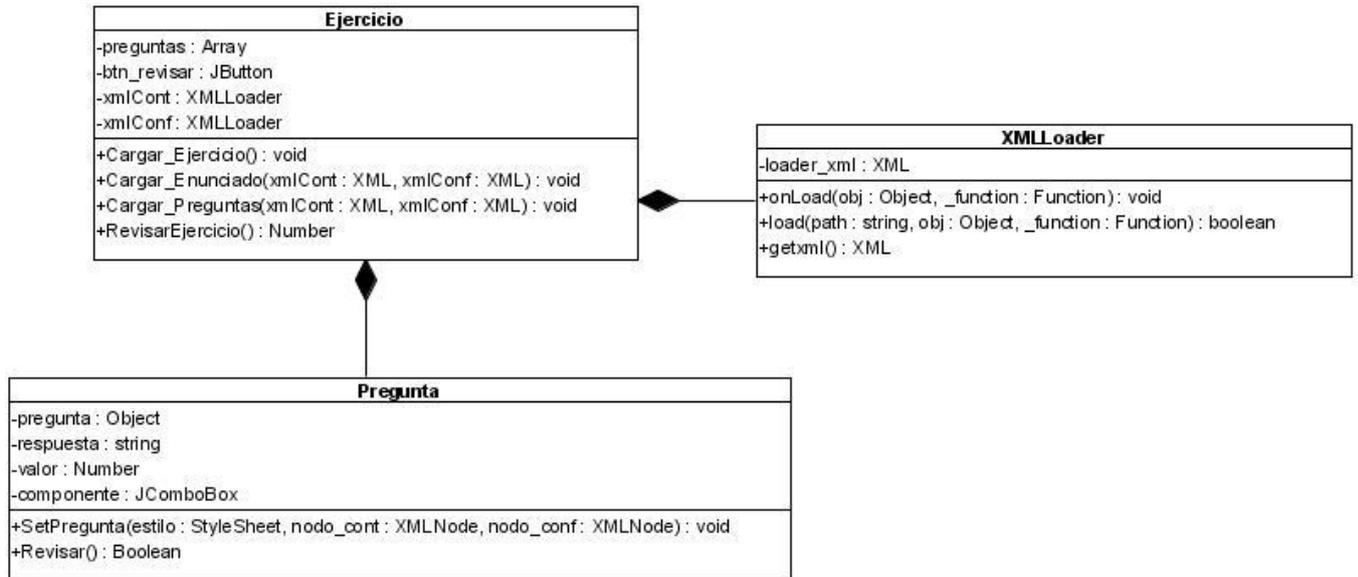


Figura 3.1: Diagrama de clases del Diseño “Verdadero o Falso”

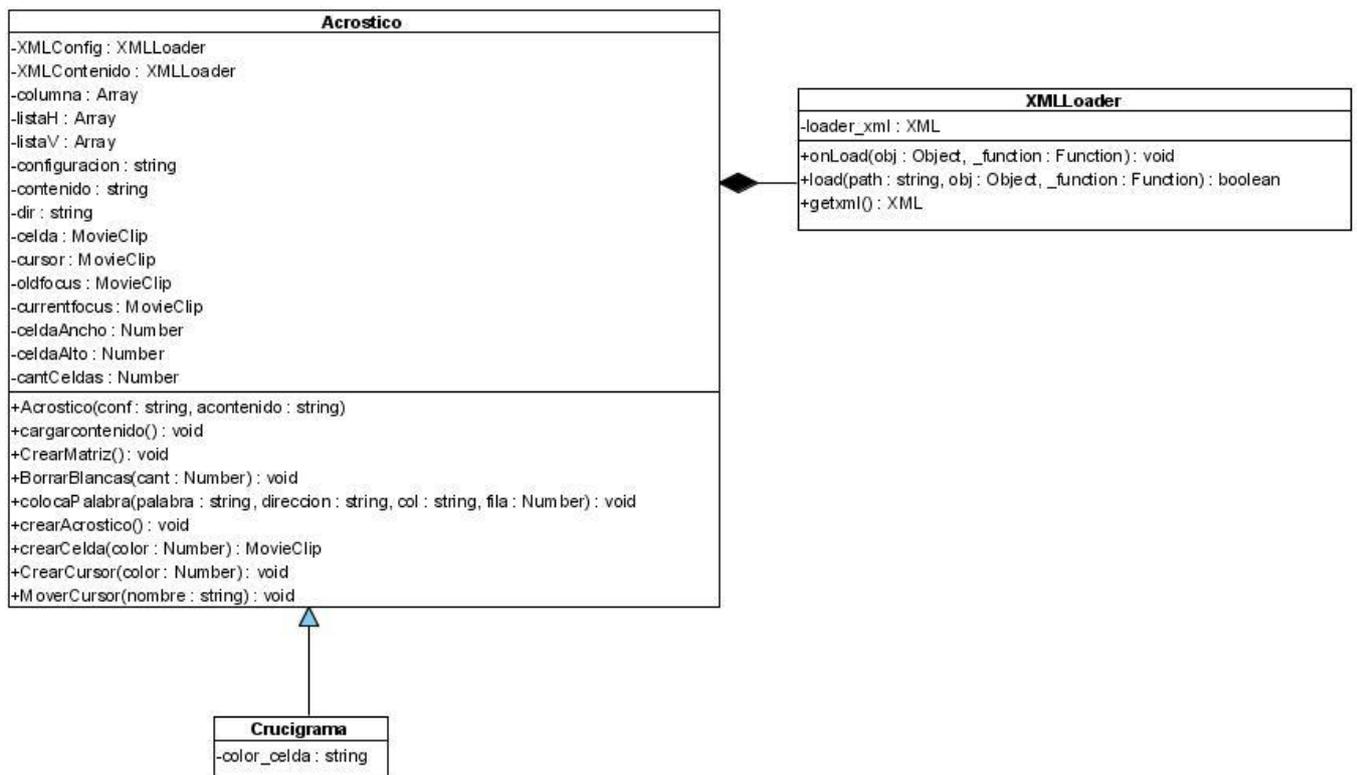


Figura 3.2: Diagrama de clases del Diseño “Acróstico y Crucigrama”

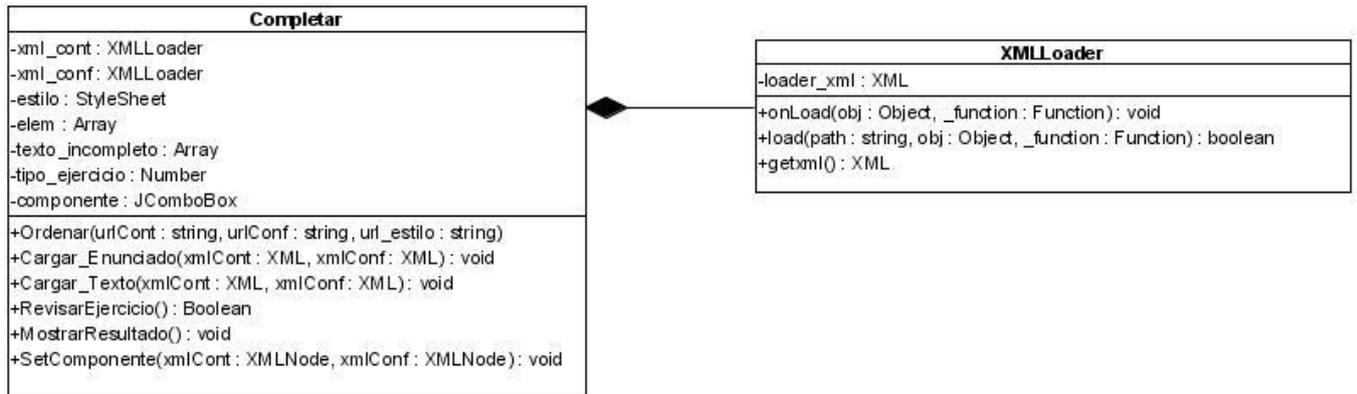


Figura 3.3: Diagrama de clases del Diseño “Completar”

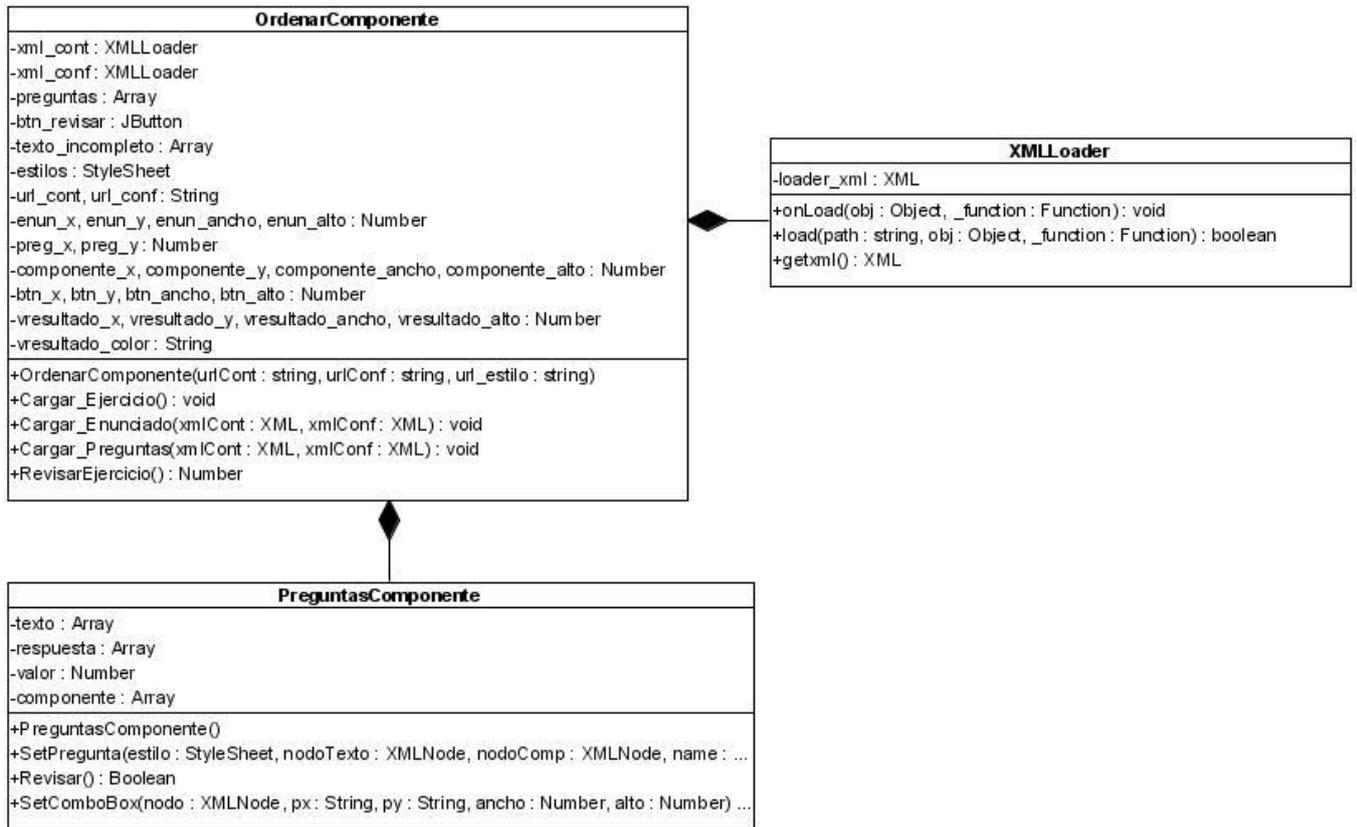


Figura 3.4: Diagrama de clases del Diseño “Ordenar con Componentes”

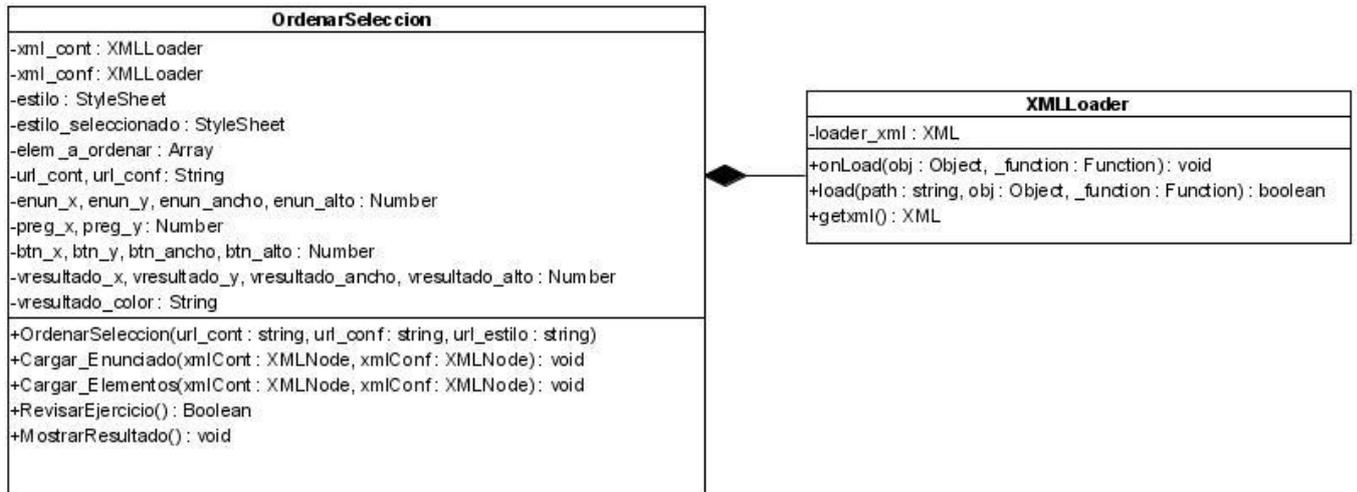


Figura 3.5: Diagrama de clases del Diseño “Ordenar con Selección”

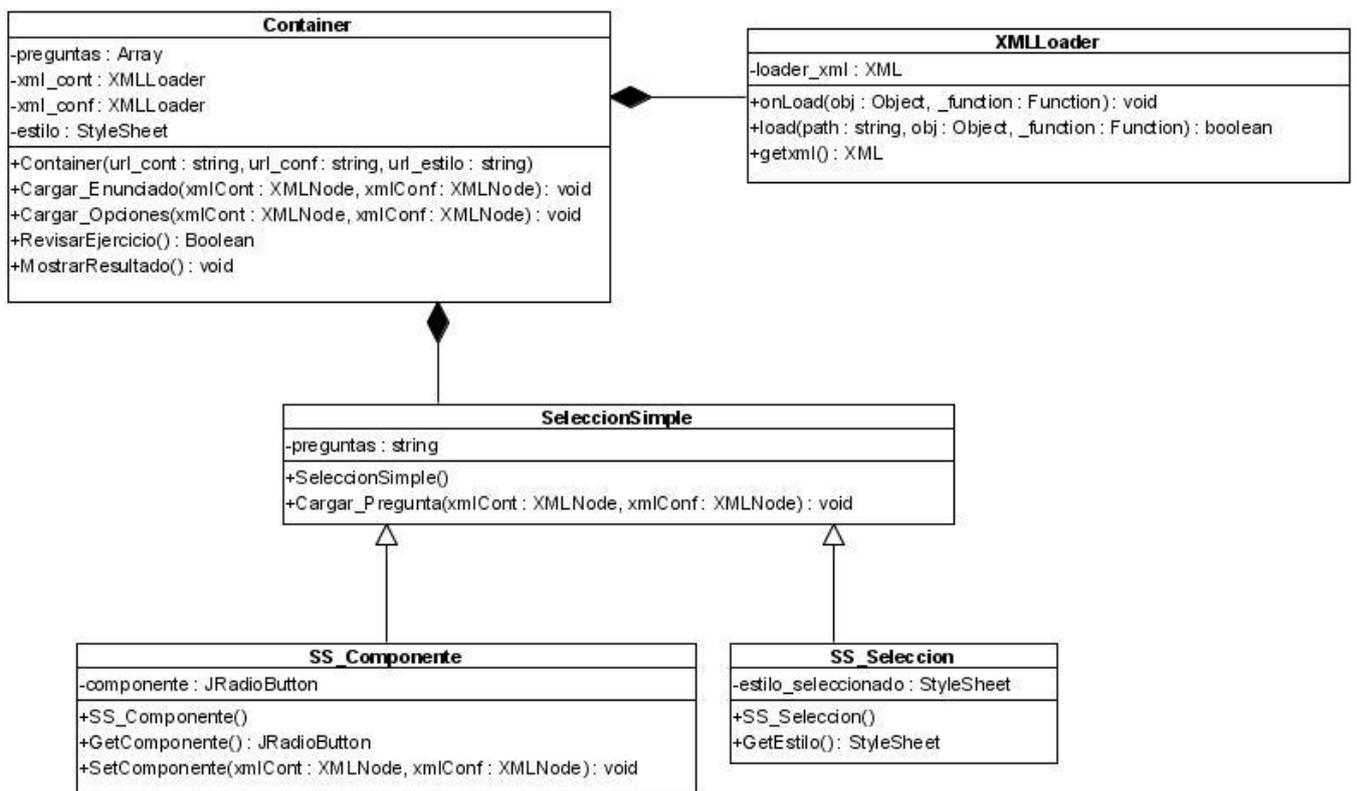


Figura 3.6: Diagrama de clases del Diseño “Selección Simple”

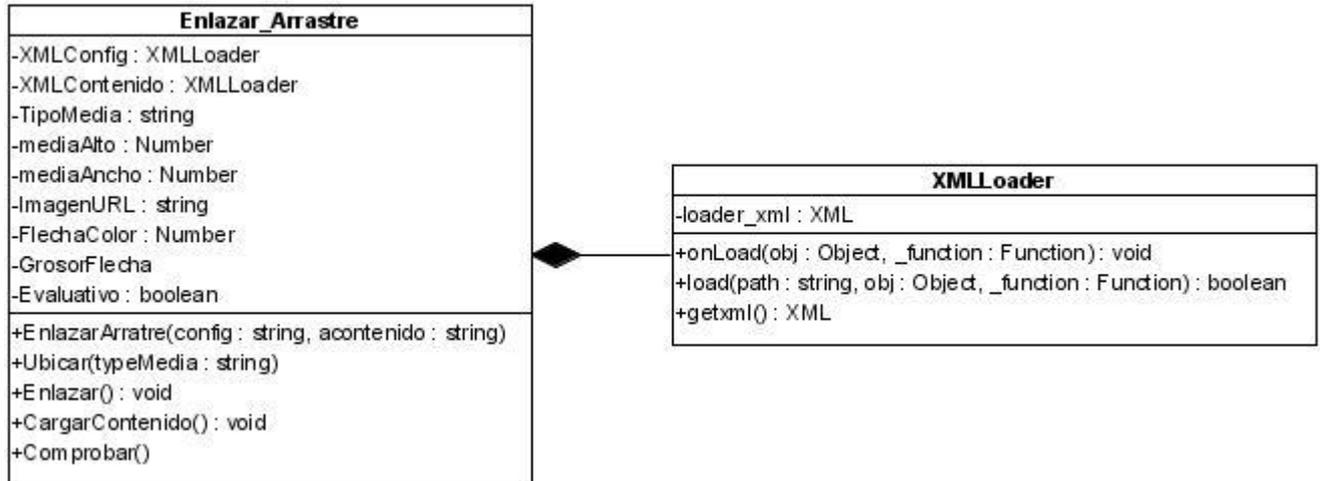


Figura 3.7: Diagrama de clases del Diseño “Enlazar con Arrastre”

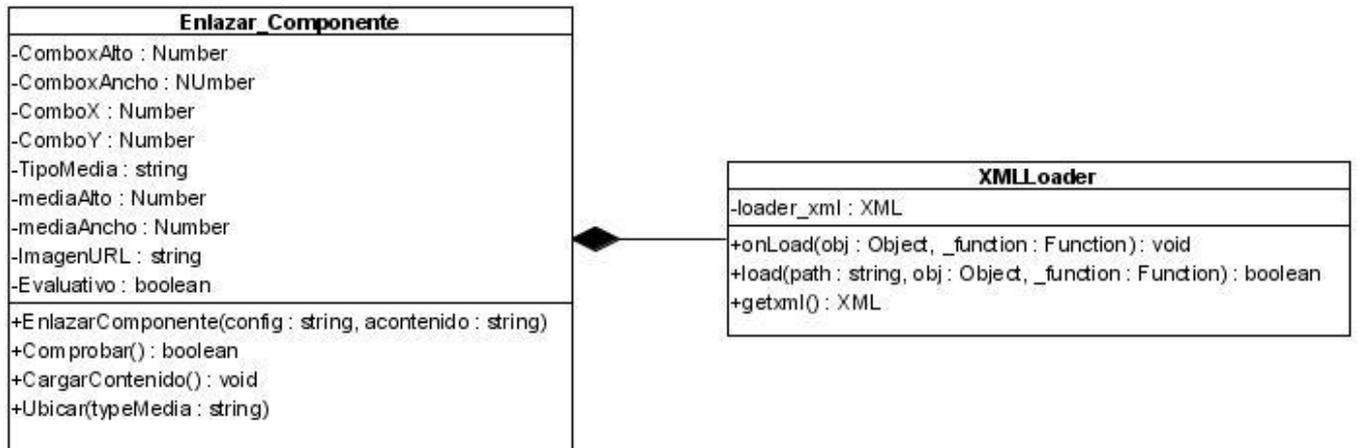


Figura 3.8: Diagrama de clases del Diseño “Enlazar con Componente”

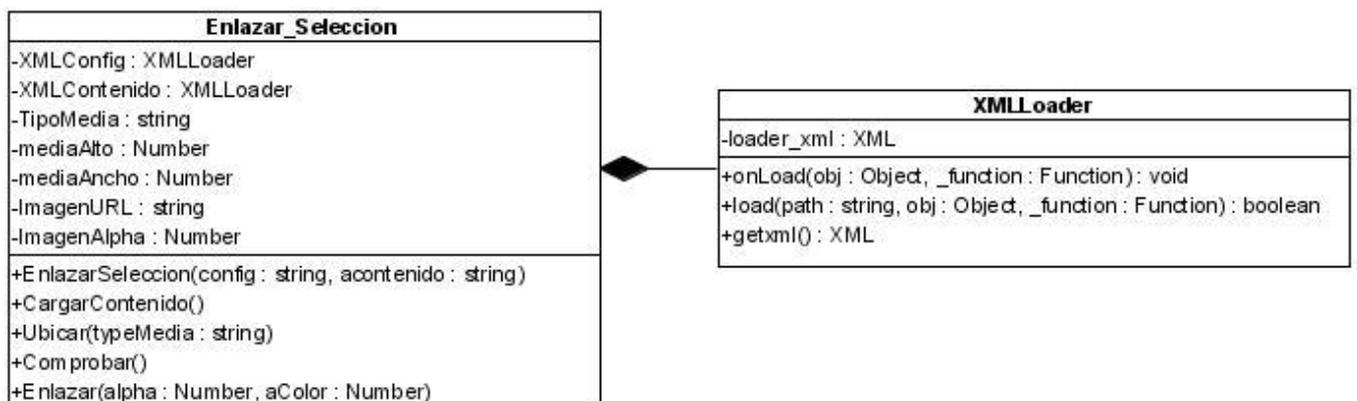


Figura 3.9: Diagrama de clases del Diseño “Enlazar con Selección”

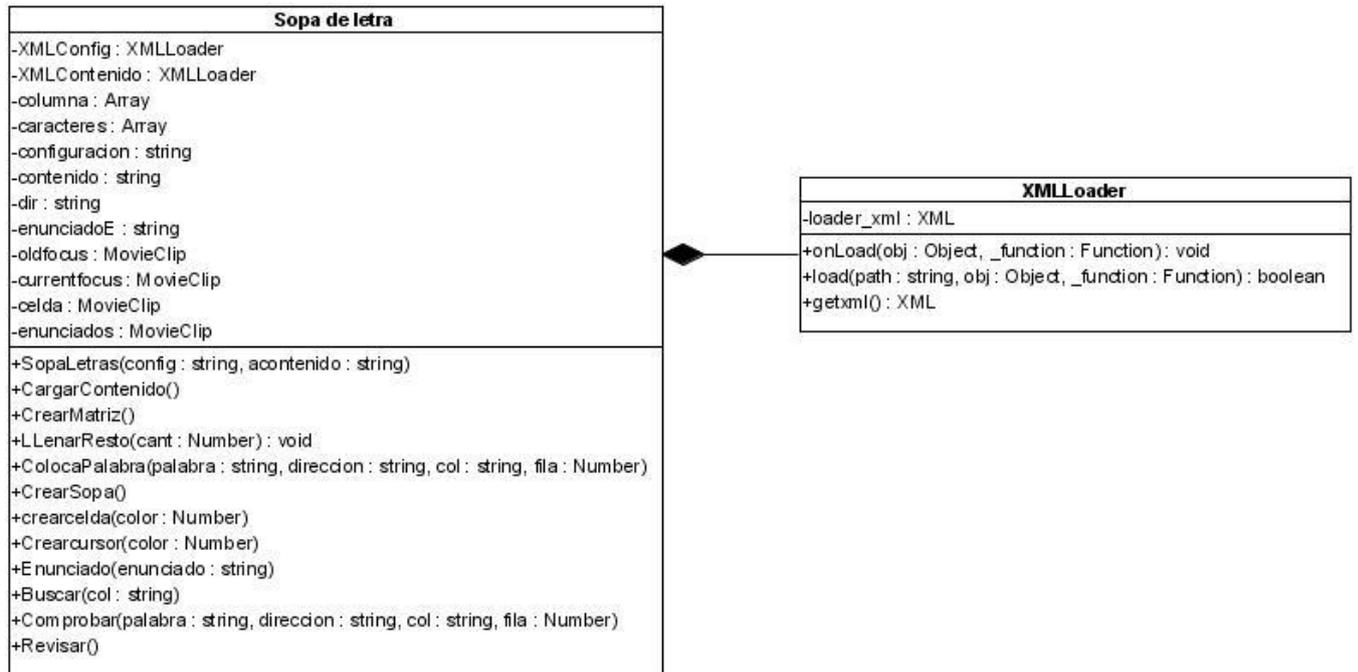


Figura 3.10: Diagrama de clases del Diseño “Sopa de Letras”

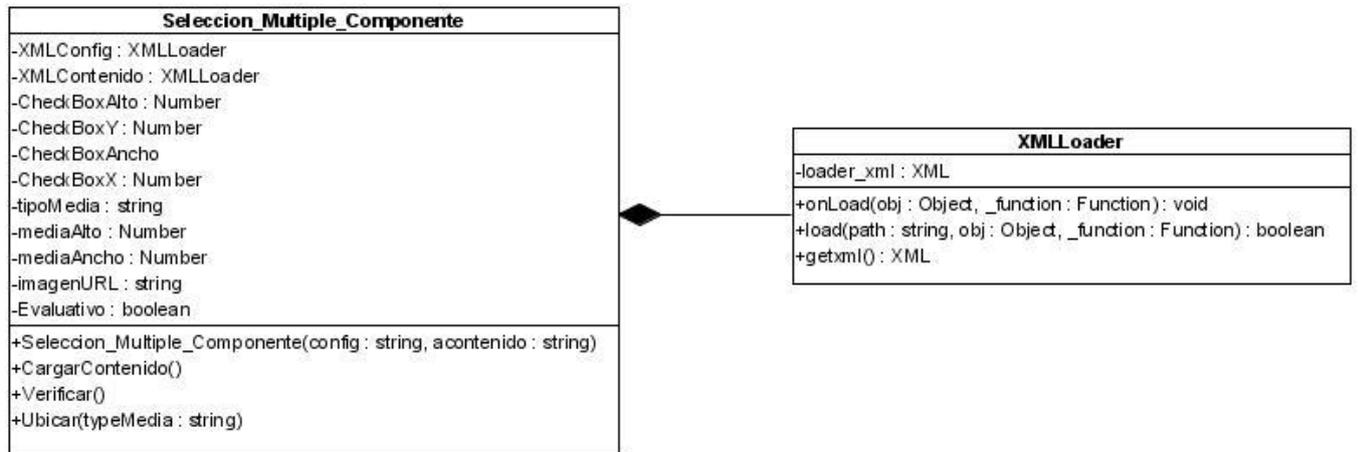


Figura 3.11: Diagrama de clases del Diseño “Selección Múltiple con Componente”

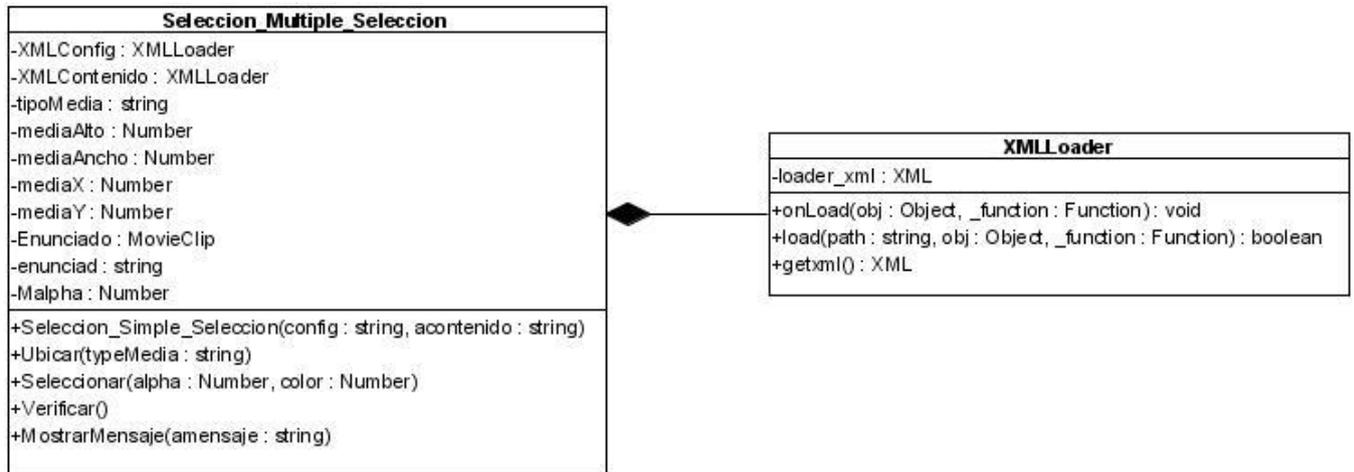


Figura 3.12: Diagrama de clases del Diseño “Selección Múltiple con Selección”

3.3 Arquitectura del Sistema

El conjunto de modelos que describen la Línea Base de la Arquitectura se denomina Descripción de la Arquitectura. El papel de la descripción de la arquitectura es guiar al equipo de desarrollo a través del ciclo de vida del sistema.

La información brindada en este documento sobre la arquitectura del sistema se concentra en el patrón utilizado para su desarrollo. Nuestro sistema se basa en una arquitectura de 3 capas organizadas jerárquicamente donde cada capa brinda servicios a la capa superior.

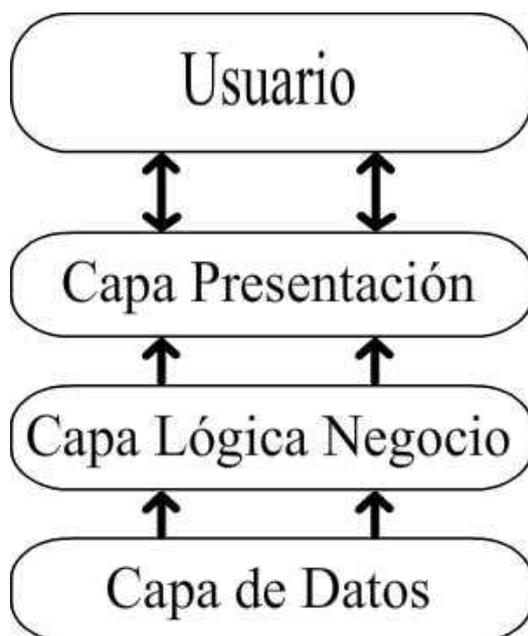


Figura 3.13: Arquitectura de 3 capas.

Ventajas:

- Facilita la descomposición del problema en varios niveles de abstracción.
- Soporta fácilmente la evolución del sistema; los cambios sólo afectan a las capas vecinas.
- Se pueden cambiar las implementaciones respetando las interfaces con las capas adyacentes.

Desventajas del uso de capas:

- No todos los sistemas pueden estructurarse en capas.
- A menudo es difícil encontrar la separación en capas adecuada.

La capa de presentación maneja la interacción entre el usuario y la aplicación es decir solo se encarga de la salida y la entrada de información al sistema.

La capa de lógica de negocio contiene la implementación de los casos de uso que dan respuesta a los requisitos funcionales.

La capa de acceso a datos contiene los medios de almacenamiento utilizados por la aplicación.

Descripción de la Arquitectura del Sistema:

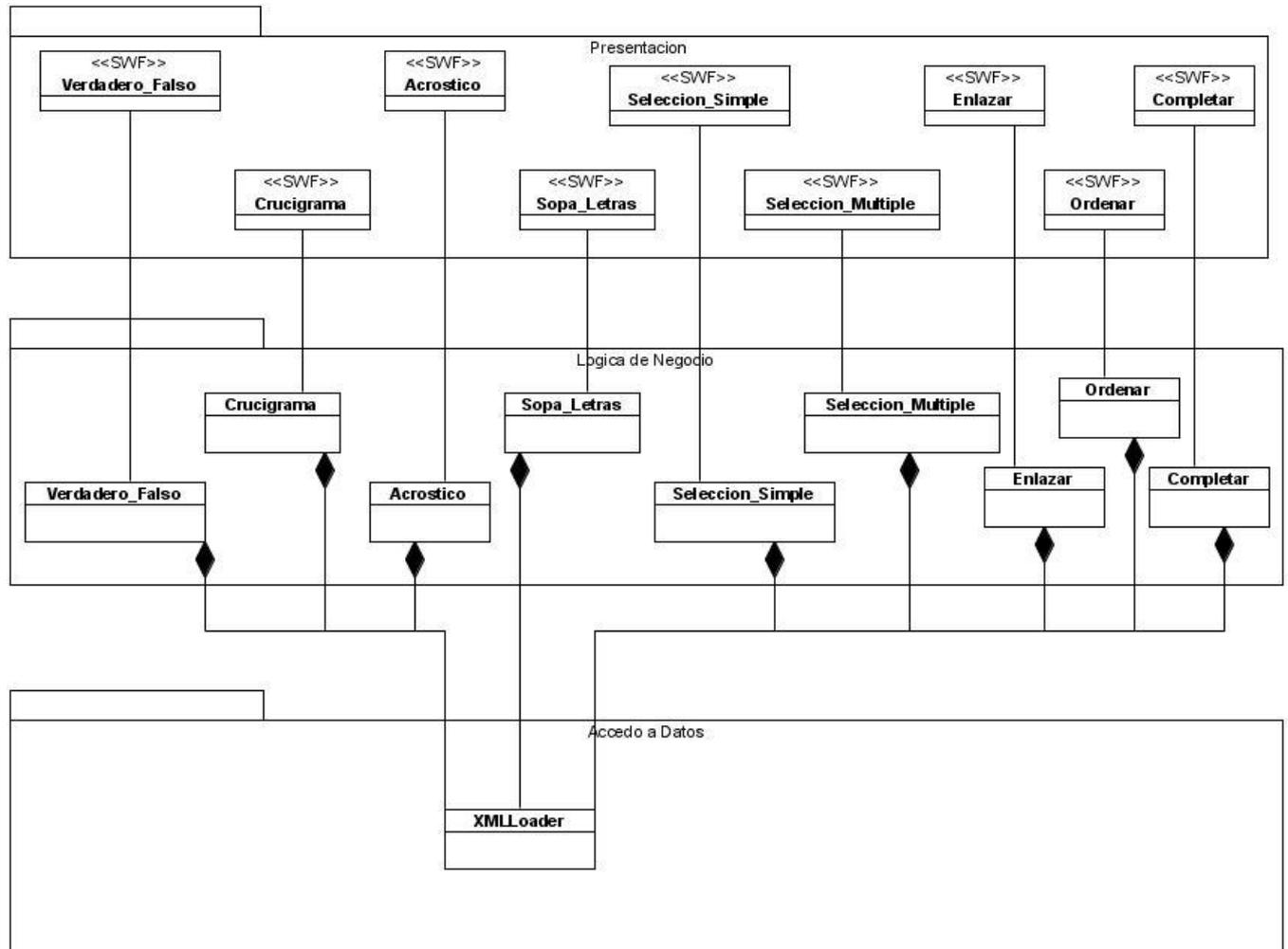


Figura 3.14: Arquitectura del Sistema

3.4 Patrones GRASP

Hemos hecho uso en el diseño del sistema de los patrones GRASP para la asignación de responsabilidades, ya que estos describen los principios fundamentales de diseño de objetos para dicha actividad. Además de que constituyen un apoyo para la enseñanza que ayuda a entender el diseño de objeto esencial y aplica el razonamiento para el diseño de una forma sistemática, racional y explicable.

Los patrones GRASP utilizados son:

Nombre del patrón	Problema	Solución
Expert - Experto	¿Cuál es un principio general para asignar responsabilidades a los objetos?	Asignar una responsabilidad al experto en información (la clase que tiene la información necesaria para la realización de la asignación).
Creator - Creador	¿Quién debería ser el responsable de la creación de una nueva instancia de alguna clase?	Asignar a la clase B la responsabilidad de crear una instancia de clase A si se cumple uno o más de los casos siguientes: <ol style="list-style-type: none"> 1. B agrega objetos de A 2. B contiene objetos de A 3. B registra instancias de objetos de A 4. B utiliza más estrechamente objetos de A. 5. B tiene datos de inicialización que se pasarán a un objeto de A cuando sea creado (por tanto, B es un Experto con respecto a la creación de A). 6. B es un creador de los objetos A.
Low Coupling - Bajo Acoplamiento	¿Cómo soportar bajas dependencias, bajo impacto del cambio e incremento de la reutilización?	El acoplamiento es una medida de la fuerza con que una clase está conectada a otras clases, una clase de bajo acoplamiento no depende de muchas otras.
High Cohesion - Alta cohesión	¿Cómo mantener la complejidad manejable?	La cohesión es una medida de cuan relacionadas y enfocadas están las responsabilidades de una clase. Una alta cohesión caracteriza a las clases con responsabilidades estrechamente relacionadas que no realicen un trabajo enorme. Las clases que no cumplen con esto presentan los

		<p>siguientes problemas:</p> <ol style="list-style-type: none"> 1. Son difíciles de comprender. 2. Son difíciles de reutilizar. 3. Son difíciles de conservar. 4. Son delicadas, las afecta constantemente los cambios.
Controller - Controlador	¿Quién debería ser el responsable de gestionar un evento de entrada al sistema?	<p>Asignar una responsabilidad de recibir o manejar un mensaje de evento del sistema a una clase que representa una de las opciones siguientes:</p> <ol style="list-style-type: none"> 1. Representa el sistema global, dispositivo o subsistema. 2. Representa un caso de uso en el que tiene lugar el evento del sistema a menudo denominado <nombre del caso de uso> Manejador, <nombre del caso de uso> coordinador, <nombre del caso de uso> Sesión. <ul style="list-style-type: none"> • Utilice la misma clase controlador para todos los eventos del sistema en el mismo escenario de caso de uso. • Informalmente, una sesión es una instancia de una conversación con un actor. Las sesiones pueden tener cualquier duración, pero se organizan a menudo en función de casos de uso.

Tabla 3.1 Patrones GRASP.

Para el acceso a los datos tanto de configuración como de contenido utilizamos una clase llamada XMLLoader la cual es la encargada de leer todo el XML y devolver un objeto con toda la información a procesar, de esta forma estamos utilizando el **Patrón de Diseño Experto**.

Uno de los tantos momentos en los cuales se pone de manifiesto el **Patrón de Diseño Creador** es cuando se establece una relación entre la clase Ejercicio y la clase Pregunta de la tipología de ejercicio Verdadero o Falso, en este caso la clase Ejercicio va a contener un conjunto de objetos de tipo Pregunta por lo tanto dicha clase es la que tiene la responsabilidad de crear objetos de tipo Pregunta.

Tanto en el flujo de trabajo de análisis y diseño como en implementación se hizo especial énfasis en que las clases a implementar cumplieran con los principios básicos de los patrones de diseño **Bajo Acoplamiento y Alta Cohesión**. En el primer caso apoyándonos en el patrón de arquitectura de 3 capas se logró que existieran entre las clases las menores dependencias posibles, permitiendo de esta manera que cualquier modificación en una de ellas no afectara el funcionamiento de las restantes. En el segundo caso se distribuyeron las funcionalidades quedando estrechamente relacionadas y sin ser de complejo entendimiento, permitiendo la reusabilidad y la tolerancia a posibles cambios.

En el desarrollo del sistema propuesto se cumplió con el patrón de diseño **Controlador** elaborando en cada caso de uso una clase controladora encargada de manejar toda la información y controlar los eventos que se generaron en la ejecución del mismo.

3.5 Modelo de Implementación.

El modelo de implementación se apoya en el diagrama de componentes para mostrar tanto los componentes software (código fuente, binario y ejecutable) como las relaciones lógicas entre ellos.

3.5.1 Diagrama de componentes de código fuente:

Lo que distingue a un diagrama de componentes de otros tipos de diagramas es su contenido. Normalmente contienen componentes, interfaces y relaciones entre ellos. Y como todos los diagramas, también puede contener paquetes utilizados para agrupar elementos del modelo. (Ana Fernandez Vilas 2001)

Estos diagramas representan las dependencias entre ficheros de código fuente, modelan las diferentes versiones de los ficheros de código fuente y guían las actividades de implementación.

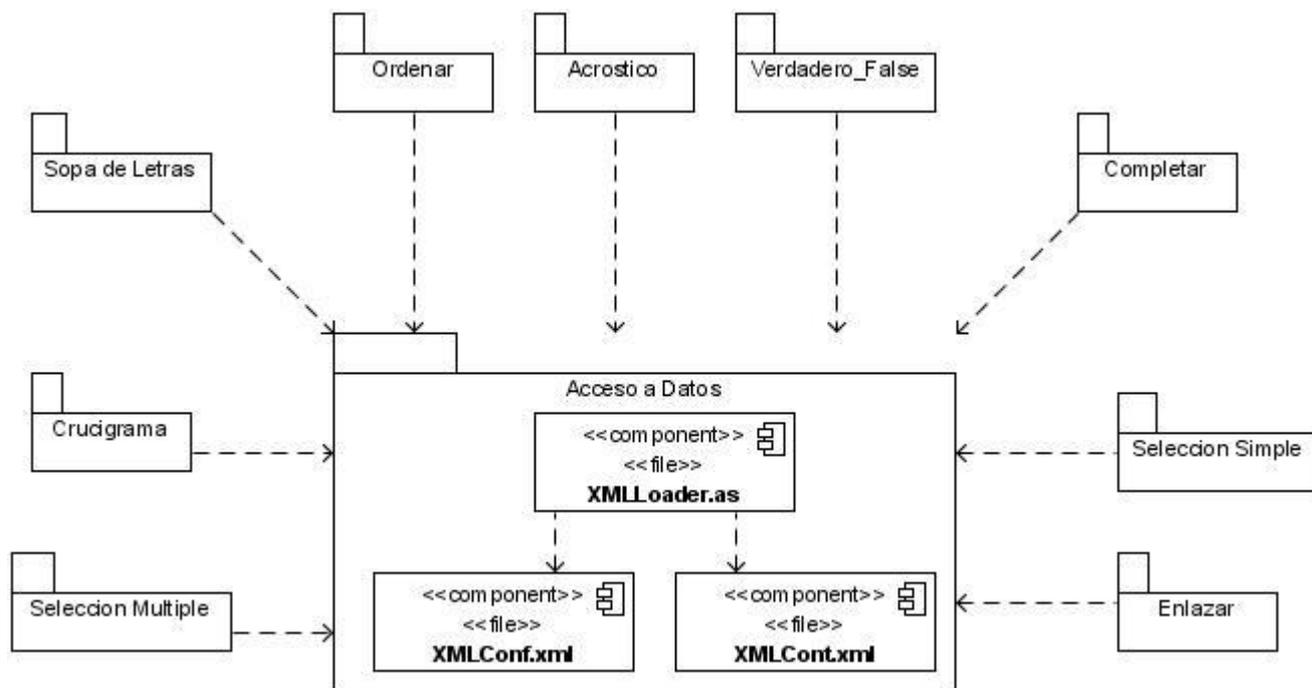
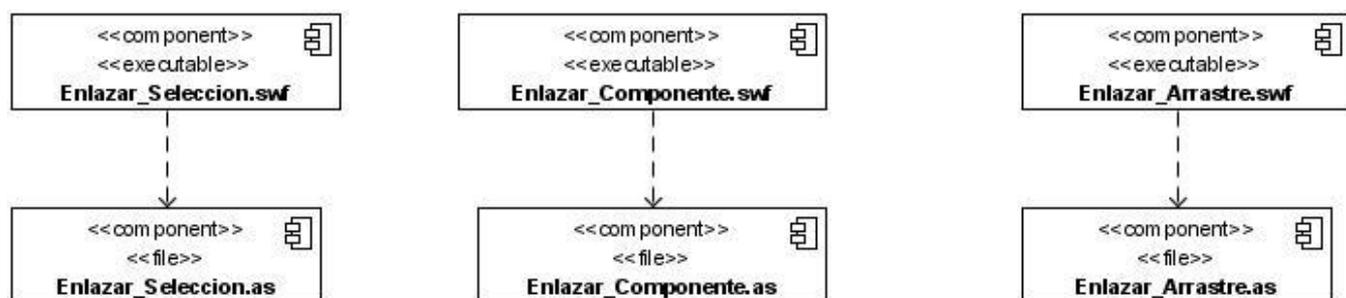
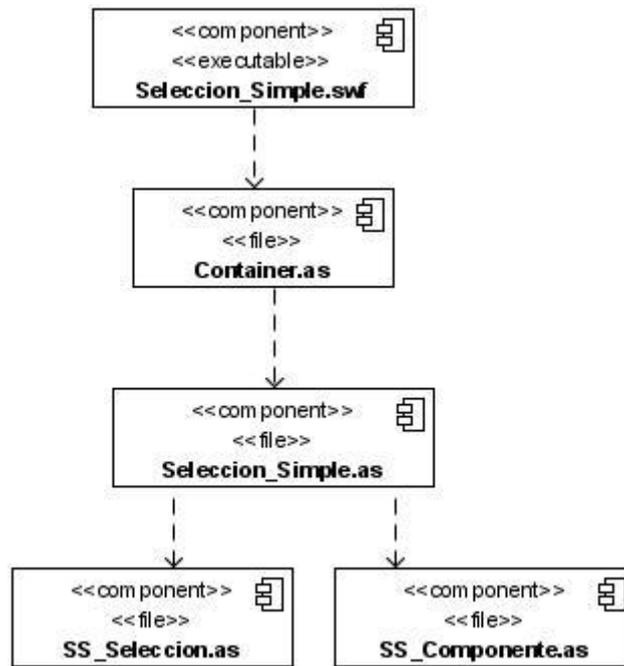


Diagrama de componentes agrupados por paquetes.

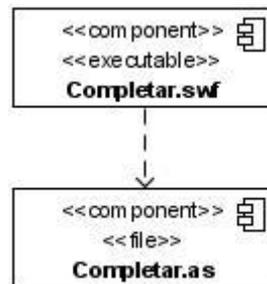
Enlazar:



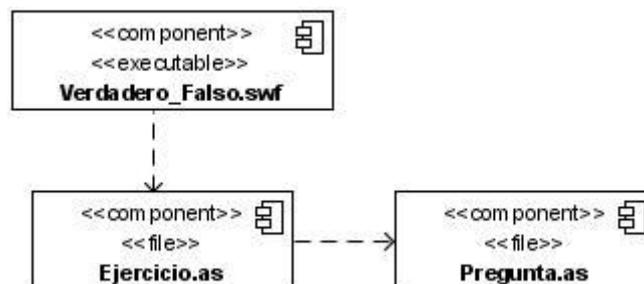
Selección Simple:



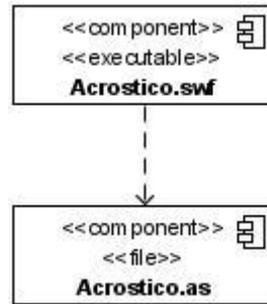
Completar:



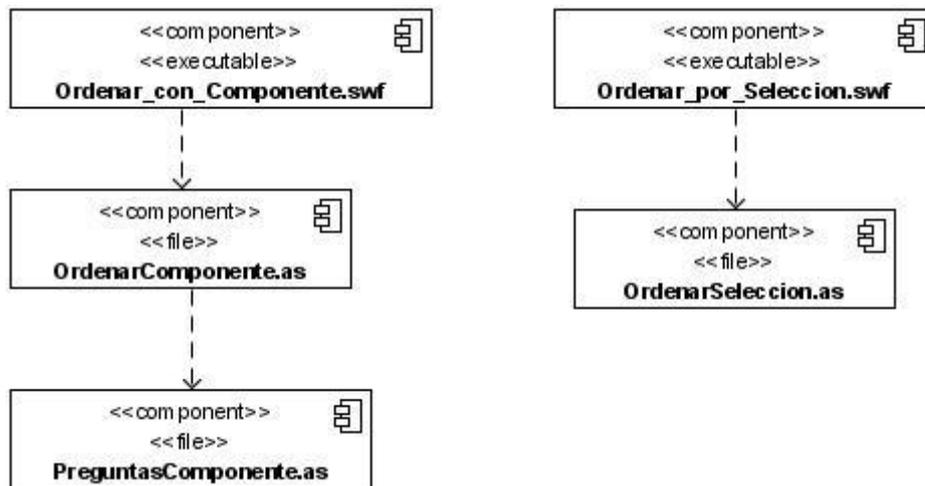
Verdadero Falso:



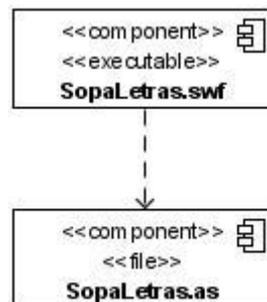
Acróstico:



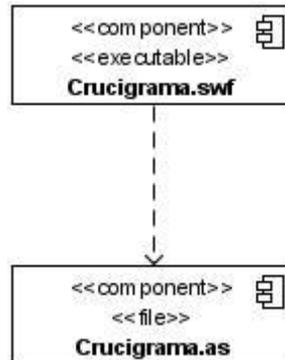
Ordenar:



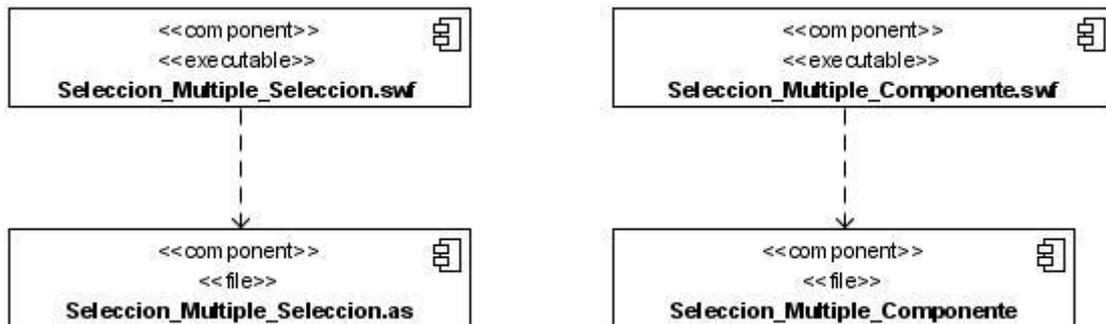
Sopa de Letras:



Crucigrama:



Selección Múltiple:



3.6 Conclusiones del Capitulo

En este capítulo se dio un acercamiento mucho más detallado de la solución expuesta en el capítulo anterior, a través del modelo de diseño y de implementación. Se describió la arquitectura basándonos en la arquitectura N capas y se expusieron los patrones de diseño, así como las clases y componentes utilizados en la construcción de la aplicación. Se analizaron los principios de diseño de la interfaz, los estándares de codificación a emplear durante la fase de construcción.

Capítulo 4

Factibilidad

4.1 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 método mediante el análisis de Puntos de Casos de Uso. Este método permite documentar los requerimientos de un sistema en términos de Actores y Casos de Uso.

4.2 Método de Estimación por Puntos de Casos de Uso

La estimación mediante el análisis de Puntos de Casos de Uso es un método propuesto originalmente por Gustav Karner de Objector y posteriormente refinado por muchos otros autores. Se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

4.2.1 Cálculo de Puntos de Casos de Uso sin ajustar.

$$UUCP = UAW + UUCW$$

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

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

Tipo de Actor	Descripción	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface)	1

Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica	3

Tabla 4.1 Calculo del Factor de Peso de los Actores sin ajustar

El usuario constituye un actor de tipo complejo, ya que es una persona que interactúa con el sistema mediante una interfaz gráfica, al cual se le asigna un peso 3.

$$UAW = 1 * 3 = 3$$

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

Tipo de Caso de Uso	Descripción	Factor de Peso
Simple	El Caso de Uso contiene de 1 a 3 transacciones	5
Medio	El Caso de Uso contiene de 4 a 7 transacciones	10
Complejo	El Caso de Uso contiene más de 8 transacciones	15

Tabla 4.2: Calculo del Factor de Peso de los Casos de Uso sin ajustar.

Casos de Uso	Tipo de Caso de Uso	Factor de Peso
Selección Múltiple con Componentes	Simple	5
Selección Múltiple con Selección	Simple	5
Selección Simple con Selección	Simple	5
Selección Simple con Componente	Simple	5
Verdadero o Falso	Simple	5
Ordenar_Palabras	Simple	5
Ordenar_Frases	Simple	5
Ordenar_Componente	Simple	5
Completar con Selección	Simple	5
Completar con Componente	Simple	5
Enlazar_Arrastre	Simple	5
Enlazar_Selección	Simple	5
Enlazar_Componente	Simple	5
Crucigrama	Simple	5
Sopa_Letras	Simple	5
Acróstico	Simple	5
Crear Ejercicio	Simple	5

Cargar Contenido XML	Simple	5
Configurar Ejercicio	Simple	5
Verificar Ejercicio	Medio	10
Total:		105

Tabla 4.3: Clasificación de los Casos de Uso.

UUCW= 105

UUCP = UAW + UUCW = 3 + 105 = 108

4.2.4 Cálculo de Puntos de Casos de Uso ajustados

UCP = UUCP x TCF x EF

UCP: Puntos de Casos de Uso ajustados

UUCP: Puntos de Casos de Uso sin ajustar

TCF: Factor de complejidad técnica

EF: Factor de ambiente

4.2.5 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	1	La velocidad es limitada por las entradas provistas por el usuario
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	1	Escasos requerimientos para la instalación
T7	Facilidad de uso	0.5	1	Normal

T8	Portabilidad	2	5	Se requiere que el sistema portable
T9	Facilidad de cambio	1	3	Se requiere un costo moderado de mantenimiento
T10	Concurrencia	1	0	No hay concurrencia
T11	Incluye objetivos específicos de seguridad	1	3	Seguridad normal
T12	Provee acceso directo a tercera partes	1	0	No tiene acceso a terceras partes
T13	Se requieren facilidades especiales de entrenamiento a usuarios	1	1	Sistema fácil de usar

Tabla 4.4: Factor de complejidad técnica (TCF)

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

$$TCF = 0.6 + 0.01 \times 26 = 0.6 + 0.26 = 0.86$$

4.2.6 Factor de ambiente (EF)

Factor	Descripción	Peso	Valor asignado	Comentario
E1	Familiaridad	1.5	4	El grupo esta bastante familiarizado con el modelo
E2	Experiencia con la aplicación	0.5	4	El grupo lleva tiempo trabajando con este tipo de aplicación
E3	Experiencia en orientación a objetos	1	4	El grupo programa orientado a objetos
E4	Capacidad del analista líder	0.5	4	Tiene experiencia en este tipo de trabajo
E5	Motivación	1	5	El grupo está motivado
E6	Estabilidad de los requerimientos	2	3	Se esperan algunos cambios
E7	Personal part-time	-1	0	Todo el grupo es full-time
E8	Dificultad del lenguaje de programación	-1	3	Se usará el lenguaje Action Script 2.0

Tabla 4.5: Factor de ambiente (EF)

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

$$EF = 1.4 - 0.03 \times 22 = 1.4 - 0.66 = 0.74$$

$$UCP = 108 \times 0.86 \times 0.74 = 68,7$$

4.2.7 Estimación del esfuerzo

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores E1 a E6.

Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores E7 y E8.

- Si el total es 2 o menos, se utiliza el **factor de conversión 20 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el **factor de conversión 28 horas-hombre/Punto de Casos de Uso**, es decir, un Punto de Caso de Uso toma 28 horas-hombre.
- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP \times CF$$

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

$$\text{Esfuerzo}_{\text{Implementación}} = 68,7 \times 20 = 1374 \text{ horas-hombre}$$

$$\text{Esfuerzo}_{\text{Total}} = 1374 \times 100 / 40 = 3435 \text{ horas-hombre}$$

$$\text{Tiempo de desarrollo} = 3435 \text{ horas} / (240 \text{ horas-mes}) = 14,3 \text{ meses}$$

$$\text{Tiempo de desarrollo por persona} = 14,3 \text{ meses} / 2 \text{ personas} = 7,1 \text{ meses-persona}$$

$$\text{Costo total proyecto} = 225 \text{ pesos} \times 14,3 \text{ meses} = 3217 \text{ pesos}$$

Actividad	Porcentaje	Horas- Hombre
Análisis	10 %	343 horas-hombre
Diseño	20 %	687 horas -hombre
Implementación	40 %	1374 horas -hombre

Prueba	15 %	525 horas -hombre
Sobrecarga	15 %	525 horas -hombre
Total	100 %	3435 horas-hombre

4.3 Beneficios tangibles

- Reusabilidad de código: Será posible reutilizar el código para futuros proyectos de desarrollo de software educativo en formato multimedia lo que agilizará y facilitará las actividades de implementación.
- Acceso al código fuente, lo que permite modificarlo y adaptarlo a las necesidades específicas de cualquier proyecto.
- Posibilidad de comercializarlo, al estar desarrollado con herramientas Open Source.

4.4 Beneficios intangibles

Aumento del conocimiento sobre el lenguaje de programación ActionScript.: Utilizando estas clases en los proyectos productivos se verá necesario conocer el funcionamiento y la estructura de ellas esto motivará a los programadores a investigar y aumentar sus conocimientos acerca de este lenguaje.

4.5 Conclusiones del Capítulo

La aplicación que se desarrolla forma parte de un proyecto de la dirección de software educativo de la Universidad de Ciencias Informáticas. El costo por desarrollar la aplicación es de **\$ 3217** pesos (moneda nacional), el cual es perfectamente recuperable con su futura comercialización.

Parámetros	Valores
Esfuerzo	3435 horas-hombre
Tiempo de desarrollo	7,1 meses
Cantidad de hombres	2 hombres
Salario medio	\$ 225
Costo	\$ 3217

Conclusiones Generales

Finalizado el desarrollo del presente trabajo se puede llegar a las siguientes conclusiones:

1. El estudio de las tendencias mundiales y herramientas existentes para el desarrollo de ejercicios con fines educativos, y en particular los enfocados al formato multimedia ha permitido determinar las funcionalidades del sistema utilizando tecnologías libres y mejores prácticas.
2. Se ha logrado obtener los requerimientos del sistema mediante el uso de las técnicas de obtención de información descritas.
3. La alta motivación del equipo de desarrollo y la experiencia en el lenguaje de programación permitieron un rápido aprendizaje y dominio de las nuevas herramientas de trabajo utilizadas para la implementación de los ejercicios, clases y componentes reutilizables.
4. En el presente trabajo no solo se propuso una solución a los problemas de migración al software libre de la Dirección de Producción #2 de la UCI sino que además dio origen a un producto para agilizar la producción de software educativos en formato multimedia. Culminado el desarrollo del sistema se dio cumplimiento, de forma satisfactoria, a los objetivos que se perseguían inicialmente:
5. Se logró un profundo estudio y documentación de las diferentes tipologías de ejercicios, los componentes y las clases que se pueden desarrollar en la implementación de software educativo multimedia.
6. Se logró el análisis y diseño para la implementación de componentes, clases y ejercicios de diferentes tipologías.
7. Se logró implementar componentes, clases y ejercicios reutilizables para la su utilización en el IDE a desarrollar.

En el estado actual del sistema solo se abarca en su primera versión de la implementación, una parte del espectro que se pretende lograr con el desarrollo de este conjunto de ejercicios y clases. Se concluye que el sistema brinda una solución factible a la problemática inicial y que su desarrollo y explotación significará una mejora considerable en las facilidades de producción nacional e internacional de software educativo en formato multimedia para la UCI.

Recomendaciones

A continuación se exponen un conjunto de recomendaciones y sugerencias que pueden ayudar a extender las funcionalidades actuales del producto:

- Elaborar una herramienta que permita generar los archivos de configuración y contenido en formato XML, con la estructura requerida para cada tipología de ejercicio.
- Utilizar el software en la producción de software educativo multimedia en la Universidad de Ciencias Informáticas.
- Mejorar aún más la integración los subsistemas Gestión Visual y Gestión de Código.
- Lograr una integración con los subsistemas Gestión Visual y Gestión de Código.
- Elaborar una guía para la utilización y elaboración de dichas clases.
- Implementar las restantes tipologías de ejercicios siguiendo los estándares de codificación establecidos y sobre todo las estructuras definidas para los ficheros XML.

Bibliografía:

Ana Fernández Vilas. Diagrama de Componentes. 2001. [Cited 10 June 2008]. Available from World Wide Web: <<http://www-gris.det.uvigo.es/~avilas/UML/node49.html>>.

Ana García-Valcárcel Muñoz-Repiso. INFORMÁTICA Y EDUCACIÓN. [Cited 10 June 2008]. Available from World Wide Web: <<http://web.usal.es/~anagv/arti5.htm>>.

Asociación javaHispano. AsWing: Swing sobre Flash - Asociación javaHispano. 2007. [cited 9 June 2008]. Available from World Wide Web: <<http://www.javahispano.org/contenidos.item.action?id=1935949118&menuId=NEWS>>.

Consejería de Educación y Ciencia. Portal de Educación de la Junta de Comunidades de Castilla - La Mancha (Contenido: Cuadernia). 2008. [cited 31 May 2008]. Available from World Wide Web: <<http://www.educa.jccm.es/educa-jccm/cm/temas/cuadernia>>.

David Grau Merconchini. Software libre II: Una estrategia decisiva de desarrollo - Cuba - Juventud Rebelde - Diario de la Juventud Cubana. 2008. [cited 10 June 2008]. Available from World Wide Web: <<http://www.juventudrebelde.cu/cuba/2008-02-14/software-libre-ii-una-estrategia-decisiva-de-desarrollo/>>.

Flash Develop Descargar | Download Flash Develop Gratis. 2007. [Cited 9 June 2008]. Available from World Wide Web: <<http://www.utilidades-utiles.com/descargar-flashdevelop.html>>.

Fran Macias. LIM. 2006. [Cited 31 May 2008]. Available from World Wide Web: <<http://www.educalim.com/edilim.htm>>.

JACOBSON, IVAR BOOCH, and GRADY RUMBAUGH. EL PROCESO UNIFICADO DE DESARROLLO DE SOFTWARE.

Juan David Soto. Mykme » Qué es SCORM? 2007. [cited 31 May 2008]. Available from World Wide Web: <<http://mykme.com/blog/2007/04/09/que-es-scorm/>>.

Junta de Extremadura. Atenex. 2007. [cited 31 May 2008]. Available from World Wide Web: <<http://atenex.educarex.es/lrASubSeccionFront.do?id=1>>.

Luis Ignacio Gómez. Exponen logros de la educación en Cuba. 2004. [cited 31 May 2008]. Available from World Wide Web: <<http://www.ain.cu/2004/noviembre/10ededucacion.htm>>.

María A. Mendoza Sánchez. Metodologías De Desarrollo De Software. 2004. [Cited 31 May 2008]. Available from World Wide Web:

<http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html>.

Miguel Rodríguez Cuervo. Computación Educativa. 2006. [Cited 31 May 2008]. Available from World Wide Web: <<http://www.rimed.cu/computacion/computacion.asp>>.

Rolando Alfredo Hernandez León, and Sayda Coello González. EL PARADIGMA CUANTITATIVO DE LA INVESTIGACIÓN CIENTÍFICA. 2002.

Sergio Martínez Dunstan. Impacto de la informática en la educación. Introducción. 2001. [Cited 31 May 2008]. Available from World Wide Web:

<<http://ciberhabitat.gob.mx/universidad/ui/eadei/eadei.htm>>.