



FACULTAD 8

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

*Colección Antonio Guiteras y Carlos Aponte:
Seis libros para la formación de conciencia política*

Autores: Mailyn Cabrera Torres

Hamlet Martínez Espino

Tutor: Ing. Guillermo Solenzal Fernández

Ciudad de la Habana, julio 2008



Declaración de Autoría

Declaramos ser los autores de la presente tesis y autorizamos a la Universidad de las Ciencias Informáticas a hacer uso de ella para lo que necesite, cediéndole de esta forma los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autores: Mailyn Cabrera Torres

Hamlet Martínez Espino

Tutor: Ing. Guillermo Solenzal Fernández



“...todos los días hay que luchar porque ese amor a la humanidad viviente se transforme en hechos concretos, en actos que sirvan de ejemplo, de movilización.”

Ernesto Guevara

De Hamlet:

A mi mamá, por darme el amor y el apoyo necesario en cada paso de mi vida, por estar siempre presente y por confiar tanto en mí.

A Migue, por estar siempre a mi lado apoyándome y dándome todo su amor de padre.

A mi abuelo Sergio que sin duda es la persona más importante de la familia y que lo quiero con toda mi vida.

A toda mi familia por su constante preocupación y por haber depositado tanta confianza en mí.

A Yunexis por todo su amor y cariño; y por su constante empuje para hacer de mí una persona mejor preparada profesionalmente.

A mis tres hermanos en la UCI (Yandry, Darién y Miguel A.) que siempre estuvimos juntos y tanto me ayudaron en los momentos en los que los necesité.

A Dayamis, Migue y Lidý por ser tan buenos amigos y por estar pendientes de mis resultados.

A todos mis compañeros de grupo con los cuales conviví 5 magníficos años.

Un agradecimiento especial a mi compañera de tesis; muchas gracias por toda tu comprensión y dedicación.

A mis compañeros del voly, en especial a Yordan (el father).

A mis compañeros del proyecto MENPET, en especial a mi tutor Guillermo y a Sergio por su constante preocupación porque las cosas salieran bien, por apoyarme y comprenderme cuando más lo necesité y por luchar incansablemente con este tema de tesis que surgió gracias a ellos.

De Maiflyn:

El mayor agradecimiento es para mi mamá, por demostrarme que sí podía cuando yo pensaba que no, por haberme dado tan buena educación, por confiar siempre en mí y alentarme a seguir...por ser la mejor madre del mundo.

A mi papi también le debo mucho, su recuerdo fue la mejor respuesta a mis preguntas e inseguridades.

Para mi hermano: gracias "chama", espero que sigas siendo mi fuente más actualizada de tecnología y series, gracias por convertirme en mi hermanito preferido.

A Guillermo, mi padrastro, por preocuparse tanto por mi estado de ánimo, por mi seguridad, por mi bienestar, en fin, por considerarme su hija.

Al resto de mi familia que también me ha apoyado mucho: mi tío Negro, Mariela, Carmen, Ale, Betty, Normita, mi abuela, todos, es una familia demasiado extensa y no podría mencionarlos uno a uno, pero de igual forma, muchas gracias.

A mi amigas de siempre: Lissy, La Chini, Dayi y Maydel, porque siempre han estado conmigo en mis buenos y malos momentos, y se han mantenido fieles a "nosotras".

A Deya y Aida, gracias por ser mis hermanitas en la UCI, por aprender a lidiar con mis incontables estados de ánimo y soportar mi mal genio, aunque bueno, por esa parte Deya me debe más....

A mi tutor: gracias por su ayuda sin límites, por su asesoramiento profesional y por su trato especial.

Para Gui: muchas gracias por tu cariño, por enseñarme cada día algo nuevo, por retarme a ser mejor, por apoyarme en mis decisiones y hacerme ver cuando estaba realmente equivocada, gracias por hacerme reír con tus ocurrencias, que son muchas, por colmarme de mimos y convertirme en tu princesa. Gracias por aparecer en mi vida.

A todos mis compañeros de grupo, de cada uno aprendí algo diferente y de cada uno me llevo un recuerdo único.

A Danyer un agradecimiento especial, no estuviera escribiendo estas líneas hoy, sino fuera por sus noches de dedicación para mi prueba de nivel, sin la mayor muestra de amistad que me proporcionó.

A mis compañeras de apartamento, por compartir años inolvidables, por hacer las mejores reuniones de debate sobre el prójimo y por formar parte de las historias para mis nietos.

A mis amigos de Venezuela que me ayudaron mucho a superarme profesionalmente y confiaron siempre en mí, que me brindaron su amistad de forma incondicional y me mostraron otras formas de ver la vida.

A Sergitín, mi cuñadito preferido, que bastante ha corrido por mi culpa y siempre está disponible para cuando lo necesito.

A todo el equipo del Proyecto MENPET, sobre todo a Nilber y a Pablo, que me han malcriado, y a los que les debo muchas meriendas.

A Keily...

A mi compañero de tesis, por su trabajo y su exquisitez con el documento.

A todos los que durante 5 años han intervenido en mi superación, a todos los amigos y compañeros con los que he compartido algún momento en esta universidad.

De Maifyn:

A mi papí, que sepa dónde quiera que esté, que todo lo he hecho por él, porque sé que estaría muy orgullosos de su “niña”... Y a mi mamá, que ha hecho muy buen trabajo con nosotros, que ha sido madre y padre a la vez, y que por encima de todo, y de todos, se merece mucho este título.

De Hamlet:

A mi mamá que es la razón principal de mi existencia, a quien le debo y dedico todo lo que he logrado en mi vida.

Resumen

El presente trabajo tiene como tema central el desarrollo de la aplicación Colección Antonio Guiteras y Carlos Aponte: seis libros para la formación de conciencia política, una aplicación que emplea la tecnología multimedia en su variante de libros electrónicos. Se realiza un estudio sobre las tendencias actuales que existen hoy para desarrollar aplicaciones de este tipo, así como el análisis y selección de las tecnologías y herramientas necesarias en este caso. Se exponen los resultados obtenidos de la investigación realizada sobre las metodologías de desarrollo de software y los lenguajes de modelado, y se justifica la selección de uno de ellos. Finalmente se realiza el análisis y el diseño del software, y a partir de los artefactos que se obtienen, la implementación del mismo.

Índice de contenidos

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 INTRODUCCIÓN	4
1.2 TECNOLOGÍA MULTIMEDIA.....	4
1.2.1 Libros Electrónicos	6
1.3 METODOLOGÍAS DE DESARROLLO DE SOFTWARE	7
1.3.1 MultiMet (Metodología de Diseño Nacional)	8
1.3.2 RUP (Proceso Unificado de Rational).....	9
1.3.3 XP (Programación Extrema).....	11
1.3.4 Metodología Seleccionada	13
1.4 LENGUAJES DE MODELADO DE SOFTWARE	14
1.4.1 Lenguaje Unificado de Modelado (UML).....	15
1.4.2 OMMMA – L: Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia	16
1.4.3 ApEM-L: Lenguaje para la Modelación de Aplicaciones Educativas	18
1.4.4 Selección del lenguaje de modelado	21
1.5 TECNOLOGÍAS ACTUALES PARA EL DESARROLLO	21
1.5.1 Macromedia Director	23
1.5.2 ToolBook.....	24
1.5.3 Macromedia Flash 8.0	25
1.5.4 Selección de la herramienta para el desarrollo de la aplicación.....	26
1.5.5 Lenguaje XML (Lenguaje de Marcas Extensible).....	27
1.5.6 Action Script 2.0	28
1.6 HERRAMIENTA DE MODELADO DE SOFTWARE.....	29
1.6.1 Rational Rose.....	29
1.7 CONCLUSIONES DEL CAPÍTULO	29
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA	30
2.1 INTRODUCCIÓN	30
2.2 SOLUCIÓN PROPUESTA.....	30

2.3 DESCRIPCIÓN DE UN MODELO DE DOMINIO	31
2.3.1 Identificación de los conceptos del dominio	32
2.4 REQUERIMIENTOS DEL SISTEMA	33
2.4.1 Requerimientos funcionales	34
2.4.2 Requerimientos no funcionales	35
2.5 VISTA DE GESTIÓN DEL MODELO	36
2.6 DIAGRAMAS DE ESTRUCTURA DE NAVEGACIÓN	37
2.7 JUSTIFICACIÓN DE LOS ACTORES DEL SISTEMA	41
2.8 DESCRIPCIÓN TEXTUAL DE LAS VISTAS DE PRESENTACIÓN	41
2.9 CONCLUSIONES DEL CAPÍTULO	63
CAPÍTULO 3: CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA	64
3.1 INTRODUCCIÓN	64
3.2 DIAGRAMAS DE ESTRUCTURA DE PRESENTACIÓN (DEP)	64
3.3 MODELO DE DISEÑO	68
3.3.1 Diagramas de clases	69
3.3.2 Diagramas de Interacción	73
3.4 MODELO DE IMPLEMENTACIÓN	73
3.4.1 Diagrama de Componentes	73
3.4.2 Diagrama de Despliegue	77
3.5 DESCRIPCIÓN DE ARCHIVOS XML	77
3.5.1 Archivo XML Libros	78
3.5.2 Archivo XML Glosario	78
3.6 CONCLUSIONES DEL CAPÍTULO	79
CAPÍTULO 4: ESTUDIO DE FACTIBILIDAD	80
4.1 INTRODUCCIÓN	80
4.2 ANÁLISIS DE PUNTOS DE FUNCIÓN Y COCOMO II	81
4.3 ESTIMACIÓN DEL ESFUERZO, DE LA CANTIDAD DE HOMBRES, DEL TIEMPO DE DESARROLLO Y DEL COSTO	84
4.4 BENEFICIOS TANGIBLES	88
4.5 BENEFICIOS INTANGIBLES	89
4.6 ANÁLISIS DE COSTOS Y BENEFICIOS	89
4.7 CONCLUSIONES DEL CAPÍTULO	90

CONCLUSIONES	91
RECOMENDACIONES.....	92
REFERENCIAS BIBLIOGRÁFICAS.....	93
BIBLIOGRAFÍA.....	96
ANEXOS	98
GLOSARIO DE TÉRMINOS	105

Índice de Figuras

ILUSTRACIÓN 1: RUP EN DOS DIMENSIONES	10
ILUSTRACIÓN 2: FASES DE XP	13
ILUSTRACIÓN 3: ESQUEMA DE LAS FORMAS DE DESARROLLO DE UNA MULTIMEDIA.....	22
ILUSTRACIÓN 4: PANTALLA PRINCIPAL DE LA APLICACIÓN.....	31
ILUSTRACIÓN 5: MODELO DE DOMINIO	33
ILUSTRACIÓN 6: VISTA DE GESTIÓN DEL MODELO	37
ILUSTRACIÓN 7: DEN SUBSISTEMA APLICACIÓN	38
ILUSTRACIÓN 8: DEN SUBSISTEMA OPCIONES PRINCIPALES	39
ILUSTRACIÓN 9: DEN SUBSISTEMA CONTENIDO	40
ILUSTRACIÓN 10: DEN SUBSISTEMA COMÚN.....	40
ILUSTRACIÓN 11: DEP VISTA PRESENTACIÓN.....	65
ILUSTRACIÓN 12: DEP VISTA INICIO.....	66
ILUSTRACIÓN 13: DEP VISTA PORTADA	67
ILUSTRACIÓN 14: DEP VISTA INFORMACIÓN	68
ILUSTRACIÓN 15: DIAGRAMA DE CLASES DE LA VISTA PRESENTACIÓN	69
ILUSTRACIÓN 16: DIAGRAMA DE CLASES DE LA VISTA INICIO.....	70
ILUSTRACIÓN 17: DIAGRAMA DE CLASES DE LA VISTA PORTADA	71
ILUSTRACIÓN 18: DIAGRAMA DE CLASES DE LA VISTA SALIR.....	72
ILUSTRACIÓN 19: DIAGRAMA GENERAL DE COMPONENTES.....	74
ILUSTRACIÓN 20: DIAGRAMA DE COMPONENTES DEL PAQUETE LIBROS.....	75
ILUSTRACIÓN 21: DIAGRAMA DE COMPONENTES DEL PAQUETE CLASES.....	76
ILUSTRACIÓN 22: DIAGRAMA DE COMPONENTES DEL PAQUETE COMÚN	76
ILUSTRACIÓN 23: DIAGRAMA DE DESPLIEGUE	77

Índice de Tablas

TABLA 1: DESCRIPCIÓN TEXTUAL DE LA VISTA PRESENTACIÓN	41
TABLA 2: DESCRIPCIÓN TEXTUAL DE LA VISTA INICIO.....	43
TABLA 3: DESCRIPCIÓN TEXTUAL DE LA VISTA PORTADA	45
TABLA 4: DESCRIPCIÓN TEXTUAL DE LA VISTA TÓPICOS	48
TABLA 5: DESCRIPCIÓN TEXTUAL DE LA VISTA INFORMACIÓN.....	50
TABLA 6: DESCRIPCIÓN TEXTUAL DE LA VISTA PALABRAS_CALIENTES	53
TABLA 7: DESCRIPCIÓN TEXTUAL DE LA VISTA RESULTADOS_BÚSQUEDA	55
TABLA 8: DESCRIPCIÓN TEXTUAL DE LA VISTA COMÚN	57
TABLA 9: DESCRIPCIÓN TEXTUAL DE LA VISTA OPCIONES_PRINCIPALES	59
TABLA 10: DESCRIPCIÓN TEXTUAL DE LA VISTA SALIR.....	61
TABLA 11: SALIDAS EXTERNAS	82
TABLA 12: CONSULTAS EXTERNAS	82
TABLA 13: FICHEROS LÓGICOS INTERNOS	82
TABLA 14: PUNTOS DE FUNCIÓN.....	83
TABLA 15: VARIABLES ESCALARES.....	85
TABLA 16: MULTIPLICADORES DE ESFUERZO	86

INTRODUCCIÓN

El 6 de diciembre de 1998 se hizo evidente un amplio sentimiento popular a favor de cambios sustantivos en la conducción de los asuntos del Estado Venezolano con la elección de Hugo Rafael Chávez Frías como Presidente de la República, quien con el respaldo del pueblo, llamó a la unidad y al diálogo constructivo para hacer una patria en la cual todos tengan cabida y participación, resguardados en el funcionamiento eficiente de las instituciones y el respeto a las leyes.

A partir de ese momento y durante los 8 años posteriores desde el comienzo del proceso revolucionario en Venezuela, muchos han sido los cambios que se han producido y que continúan efectuándose de manera continua en la actualidad como resultado de este proceso.

Uno de los cambios por los que se aboga hoy en Venezuela, es por la masificación de la cultura, por lograr la formación de un pueblo culto y socialmente preparado. Por otra parte, se necesitan líderes con una conciencia política bien formada, con ideas firmes y que defiendan los ideales por los que se lucha cada día.

En Venezuela y en otros países de Latinoamérica, se le denomina *parroquia* a la unidad político-territorial en la que se dividen algunos municipios, y que cuenta con un Jefe Civil y una Junta Parroquial. En la actualidad, apenas existe una formación política en los dirigentes de dichas parroquias y es de vital importancia formar en ellos un hombre nuevo y honesto, que contribuya al éxito de la Revolución que se quiere lograr.

El Ministerio del Poder Popular para la Energía y Petróleo de Venezuela (MENPET) cuenta con una Oficina de Comunicación y Relaciones Institucionales, que tiene como misión, entre otras, de difundir a diferentes sectores de la población, información sobre la conservación del medio ambiente, el uso de energías alternativas, la revolución energética, la formación política y otros temas, que por su importancia cognitiva y social, es de interés para la organización, que el pueblo conozca y alcance una cultura vasta sobre las ideas que se les quiere transmitir.

Teniendo en cuenta que los materiales interactivos para tratar los temas relacionados con la preparación política en el país son escasos, y aprovechando el marco de convenios Cuba-Venezuela, el MENPET solicitó a la Universidad de las Ciencias Informáticas (UCI) la creación de un software, que proporcione información política e ideológica en el ámbito nacional e internacional, y contenidos que tributen a la formación de valores morales y sociales. Dada la gran importancia que tiene este tema para el desarrollo político del país, el producto estará dirigido principalmente a los dirigentes de

parroquias y líderes políticos, que en su mayoría poseen un nivel de escolarización mínimo de 6to grado, y que pueden estar insertados en muchos casos a las misiones de superación o continuidad de estudio en diferentes estados o zonas del país, facilitándoles adquirir la conciencia política que necesitan para dirigir al pueblo en el proceso revolucionario, ya que a pesar de los esfuerzos realizados por el gobierno, todavía la escasa preparación política de sus líderes y dirigentes de parroquias sigue siendo uno de los mayores problemas a resolver.

Resumiendo entonces la situación existente alrededor de este tema, queda definido como problema científico: la escasa existencia de materiales interactivos que proporcionen información política para contribuir a la formación de los dirigentes de parroquias venezolanos.

A partir del problema planteado se decide estudiar el proceso de desarrollo de software con tecnología multimedia, y como campo de acción específicamente el proceso de análisis, diseño, e implementación de software con tecnología multimedia.

Con el propósito de encontrar solución al problema se define como objetivo general: desarrollar un software con tecnología multimedia que contribuya a la formación política de los dirigentes de parroquias venezolanos.

Para dar cumplimiento a este objetivo se han propuesto las siguientes tareas de investigación:

- ✓ Realizar un estudio sobre las tendencias actuales para el desarrollo de aplicaciones con tecnología multimedia.
- ✓ Realizar un análisis y selección de las tecnologías y herramientas a emplear para el desarrollo de este tipo de aplicaciones.
- ✓ Investigar sobre las metodologías existentes para el desarrollo de aplicaciones con tecnología multimedia, así como los lenguajes necesarios para su modelación.
- ✓ Realizar el análisis y diseño de la aplicación.
- ✓ Implementar el producto utilizando la herramienta y el lenguaje de programación seleccionado.

Del desarrollo de la colección de libros electrónicos con tecnología multimedia, y por ende del presente trabajo de diploma, se espera obtener como primer resultado, un producto de calidad que permita a los dirigentes de parroquias venezolanos enriquecer su cultura y formación política y convertirse en hombres nuevos capaces de entender el momento histórico que hoy se vive en Venezuela y guiar al pueblo por los caminos de la revolución, que brinde la posibilidad de obtener (a partir de la impresión) los contenidos mostrados en la aplicación, o al menos parte de ellos, que posibilite la búsqueda exacta

de determinados contenidos, que facilite la lectura y que se amenice con el apoyo de música referente al tema. También resultará un avance en el proceso de formar política y moralmente a los dirigentes de este país y de fomentar la cultura política en la población. Por otra parte se espera documentar el producto de forma tal que tanto los desarrolladores como los clientes y usuarios finales entiendan sin dificultad sus características y funcionamiento.

Para abordar todo el contenido asociado al presente trabajo, se decidió estructurar el mismo en 4 capítulos:

Capítulo 1: Fundamentación teórica. Resume el estudio realizado sobre software con tecnología multimedia, los conceptos principales de esta tecnología, sus aplicaciones y las variantes en las que se presenta. Además se aborda el tema de las metodologías de desarrollo de software y de los lenguajes de modelado y se fundamenta su selección para el proceso de desarrollo actual. Las tendencias actuales de las herramientas de desarrollo, es otro aspecto tratado en este capítulo, así como los lenguajes de programación y las herramientas CASE para el modelado del software.

Capítulo 2: Descripción de la solución propuesta. En este capítulo se realiza el modelamiento del negocio estudiado, a través de un modelo de dominio. Se describe la solución propuesta para el problema planteado en la investigación, se describen los requisitos definidos por el cliente para la elaboración del producto y se definen las vistas de presentación del software y sus descripciones.

Capítulo 3: Construcción de la solución propuesta. Capítulo que resume los flujos de trabajo de Análisis y Diseño e Implementación, mostrando los diagramas de presentación, de clases, de componentes y de despliegue, que constituyen los artefactos más significativos de estos flujos.

Capítulo 4: Estudio de factibilidad. Este capítulo incluye el estudio de factibilidad del producto a desarrollar basado en el método de estimación COCOMO II.

Capítulo 1

Fundamentación teórica

1.1 Introducción

La tecnología Multimedia, con sus reconocidas aplicaciones e influencias en lo que puede considerarse un mundo actual casi automatizado, y entiéndase por automatizar la acción de controlar o guiar determinados procesos de la vida humana a través de computadoras, ha tomado auge en los últimos años en el mundo de la producción de software, debido principalmente a las facilidades que brinda para captar la atención del público y transmitirle información.

El objetivo de este primer capítulo es fundamentar teóricamente la selección de este tipo de tecnología como solución al problema científico planteado, por lo que se introducirán temas relacionados con el ámbito de multimedia, enfatizando en sus aplicaciones y conceptos generales asociados, y se analizarán las diferentes metodologías, lenguajes y herramientas de desarrollo de software.

1.2 Tecnología Multimedia

La tecnología Multimedia se inicia en 1984, año en que Apple Computer lanzó la Macintosh, la primera computadora con amplias capacidades de reproducción de sonidos equivalentes a los de un buen radio AM. Esta característica, unida a que su sistema operativo y programas se desarrollaron, en la forma que ahora se conocen como ambiente Windows, propicios para el diseño gráfico y la edición, hicieron de la Macintosh la primera posibilidad de lo que se conoce como Multimedia.

Según la Vigésima Segunda Edición del Diccionario de la Real Academia de la Lengua Española, el término multimedia se define como:

1. adj. Que utiliza conjunta y simultáneamente diversos medios, como imágenes, sonidos y texto, en la transmisión de una información.

Esta definición está bastante cercana a lo que se conoce hoy en el mundo como tecnología multimedia, sólo necesita ser adaptada al entorno Informático. Para ello, es necesario agregar que todos los medios que se integran (imágenes, textos, videos, sonidos) lo hacen en un único programa

informático (software) que es controlado por el usuario a través de una computadora, dando la posibilidad de que exista o no interactividad entre ellos.

Son muchas las aplicaciones que ha encontrado esta nueva tecnología en diversos campos en la sociedad. Comenzó por aplicaciones en la diversión y el entretenimiento a través de los juegos de video. De allí se pasó a las aplicaciones en la información y la educación, para pasar al campo de la capacitación y la instrucción, a la publicidad y marketing hasta llegar a las presentaciones de negocios, a la oferta de servicios y productos y a la administración. Inicialmente, lo que se aprovecha de este recurso es su enorme capacidad de ofrecer información atractiva, y su característica de interactividad, que permite navegar por el programa y buscar la información sin tener que recorrerlo todo, logrando con ello que la tecnología se aplique en los nuevos medios de dos modos diferentes y se use de tres formas alternativas. En los nuevos medios se aplica como medio de aprendizaje (por interacción, al ritmo personal, simulando situaciones reales y con juegos que agilizan habilidades) y como medio informativo (conectado a bibliotecas electrónicas). Los usos de esta tecnología están enmarcados en: los medios de orientación (presentaciones multimedia de índices de orientación en bancos y museos y por módulos o kioscos de información), los medios didácticos (capacitación: interactividad y simulaciones) y los libros electrónicos (mediante el CD-ROM se puede tener acceso a libros y bibliotecas). De todas las ventajas que desde su surgimiento ha tenido la multimedia como tecnología, su gran aporte al desarrollo de la educación, incluso a la educación a distancia, ha sido el de mayor influencia en la sociedad.

Aunque la definición de multimedia es sencilla, trabajar en ella es complicado. No sólo se debe comprender cómo hacer que cada elemento se levante, sino también se necesita saber cómo utilizar las herramientas computacionales y las tecnologías de multimedia para que trabajen en conjunto.

Una de las variantes en las que se encuentran hoy aplicaciones de este tipo, es como libros electrónicos, los cuales permiten la comunicación de ideas y conocimientos entre autor y lector, tal vez de la misma forma en que lo haría tradicionalmente el libro en papel, pero añadiéndole otras capacidades y características propias de la tecnología multimedia. Precisamente esta variante de aplicaciones será la adoptada para dar solución al problema planteado en la investigación.

1.2.1 Libros Electrónicos

El libro electrónico surge por la búsqueda de un soporte que se aproximara más a la capacidad de la memoria humana. Desde su creación ha provocado una serie de transformaciones y ha obligado a la industria editorial a adentrarse en una nueva forma de producción de libros. [1] Con la creación de nuevos modelos en la escritura, la pantalla se convierte en el nuevo espacio de la misma, y por consiguiente, en un nuevo espacio para la lectura.

La escritura en este nuevo soporte se libera de la linealidad tradicionalmente obligada entre las hojas de los libros. Con el libro electrónico existe la posibilidad de que al llegar a una palabra, con sólo tocarla, la lectura tome otro rumbo, que aparezca otro texto distinto del que llegaría si se continuara con una lectura tradicional. El texto adquiere una cualidad hipertextual, imposible de lograr en un texto impreso, y convirtiendo al lector en un navegante de la información.

Son muchas las definiciones que puede tener un libro electrónico. Para algunos autores, los libros electrónicos son conjuntos de piezas de información de distintos tipos y de naturaleza heterogénea y multimedia que constituyen una unidad lógica desde el punto de vista de la comunicación, y están organizados según una estructura susceptible de consulta no-secuencial, por navegación y otros medios. En otras palabras, un libro electrónico es un hipertexto con imágenes, gráficos y sonidos, en el que se tiene la posibilidad de navegar a través de él en forma interactiva; es un programa de hipermedias, que organiza la información basándose en la metáfora de un libro. [1]

Sin la limitación que impone la impresión y la encuadernación, los libros electrónicos ofrecen enlaces de hipertexto, ejecutan búsquedas de palabras claves, y amplían la noción del conocimiento y el aprendizaje de muchas otras maneras. Estos libros aumentan enormemente las posibilidades del texto, al integrar en el mismo, prestaciones multimedia: vídeo, fotografía, animación y gráficos. Las herramientas de las bases de datos hacen posible los enlaces de referencias cruzadas a gran escala, las búsquedas en función de múltiples criterios y una información a la medida de las necesidades del usuario.

En la actualidad se está viviendo una alfabetización de carácter tecnológico, en la que el hombre debe aprender a leer y a escribir de manera no lineal. De esta forma el libro electrónico podrá empezar a ser visto por más personas como un objeto interesante y con muchas ventajas. [2]

Como todo software de calidad, este tipo de aplicaciones con tecnología multimedia también deben realizarse con la puesta en práctica de alguna metodología que guíe su desarrollo, o sea, alguna

metodología de desarrollo de software, pues es esta la única forma de obtener un producto que satisfaga completamente las necesidades del cliente y con la máxima calidad requerida.

1.3 Metodologías de desarrollo de software

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

“Las empresas que desarrollan software no pueden ignorar que su negocio es un negocio de software, y que el modelo que cada una adopte para las actividades de desarrollo y mantenimiento tiene implicaciones relevantes en la eficiencia general del negocio. El problema que pueden encontrar quienes deciden implantar métodos más eficientes, es caer en la desorientación ante el abanico de modelos de calidad, de procesos y de técnicas de trabajo desplegado en la última década; o abrazar al primero que se presenta en la puerta de la organización como “solución” de eficiencia y calidad”. [3]

“Las organizaciones que desarrollan o mantienen software pueden optar por trabajar a la buena de Dios, o por seguir una metodología. Hacerlo a la buena de Dios no es tan raro. Es la primera forma que se empleó para desarrollar programas: Aquí tenemos unos ordenadores, aquí unos señores a los que les encanta leer los manuales de programación, y se trata sencillamente de conseguir que estas máquinas terminen imprimiendo facturas (o haciendo lo que sea)”. [3]

Luego de haber decidido que sí es necesario utilizar alguna de las metodologías existentes para realizar un software, aparece una incógnita mayor a la de si decidirse o no por utilizarla, la incógnita de: cuál de todas es la mejor, cuál se corresponde más con el tipo de producto que se quiere obtener, cuál está más acorde con las características del equipo de trabajo, del entorno, del tiempo máximo para la entrega, en fin, deben considerarse aspectos muy importantes antes de adoptar una determinada metodología de desarrollo.

Hoy en día existen muchas tendencias de metodologías que brindan diferentes marcos que los desarrolladores pueden emplear a la hora de realizar su trabajo, algunas de ellas son:

- ✓ MultiMet (Metodología de diseño nacional)
- ✓ RUP (Proceso Unificado del Rational)
- ✓ XP (Programación Extrema)

“Para desarrollar un proyecto de software es necesario establecer un enfoque disciplinado y sistemático. Las metodologías de desarrollo influyen directamente en el proceso de construcción y se elaboran a partir del marco definido por uno o más ciclos de vida”. [4]

1.3.1 MultiMet (Metodología de Diseño Nacional)

Metodología de diseño nacional que describe etapas generales de la organización de un proyecto informático de multimedia. Tiene como objetivo que cada especialista integrante del equipo de desarrollo conozca la aplicación de forma integral y pueda dirigir su trabajo hacia un fin común. Se inicia con un estudio preliminar donde deben quedar definidos algunos elementos básicos relacionados con las necesidades de los usuarios y los objetivos, la tecnología necesaria, el personal de desarrollo, un estudio del mercado potencial y la estrategia de comercialización. En adición se confecciona un plan que incluye todas las etapas del desarrollo con fecha de inicio, de terminación y responsables. Se hace un estudio de factibilidad económica y técnica centrado en la relación costos - beneficios, el impacto del producto final, costo de los elementos que hacen falta para el desarrollo, crecimiento potencial en el mercado y recursos disponibles. Luego de este estudio se determina si es factible o no desarrollar el producto y continuar con el resto de las etapas. La siguiente es la etapa de definición de contenidos, donde se definen los objetivos desde el punto de vista de la aplicación propiamente dicho, teniendo en cuenta si es educativa, demostrativa o informativa, con la identificación del usuario final del sistema, basado en que los criterios de diseño están en función de su satisfacción. No modela la arquitectura del producto, sino la idea de su funcionamiento. [5]

En el paso de implementación, considera preparada toda la información a incluir y el funcionamiento integral del sistema desde el punto de vista de las acciones del usuario, selecciona entonces la herramienta de autor a utilizar y comienza el montaje del software.

Por último en la fase de prueba garantiza la revisión por dos puntos de vista: solidez de la información y el funcionamiento adecuado. Elabora un plan de pruebas propia, espera la revisión del usuario y se centra en los aspectos de distribución del producto.

Analizando algunos aspectos de esta metodología se señala la descripción lineal del proceso y cómo la selección de la herramienta a desarrollar se realiza en una fase cercana a la implementación y después de la elaboración. Es en la etapa de inicio donde se debe decidir con qué herramienta trabajar, para poder orientar la estructura y descripción del contenido hacia las potencialidades de la misma, y no esperar a la etapa de producción para indagar cuál se ajusta mejor al contenido. Describe

la navegación del producto a través de un diagrama de flujo y la utilización de medias y sus tipos en tablas. [6]

1.3.2 RUP (Proceso Unificado de Rational)

Como todas las metodologías, RUP es también un proceso que define Quién (Trabajadores) debe hacer Qué (Artefactos), Cuándo (Flujo de trabajos) y Cómo (Actividades) debe hacerlo. Tiene como características fundamentales:

- ✓ **Dirigido por casos de uso:** Los casos de uso reflejan lo que los usuarios futuros necesitan y desean, lo cual se capta cuando se modela el negocio y se representa a través de los requerimientos. A partir de aquí los casos de uso guían el proceso de desarrollo ya que los modelos que se obtienen, como resultado de los diferentes flujos de trabajo, representan la realización de los casos de uso (cómo se llevan a cabo). Los casos de uso se especifican, se diseñan y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba.
- ✓ **Centrado en la arquitectura:** La arquitectura muestra la visión común del sistema completo en la que el equipo de proyecto y los usuarios deben estar de acuerdo, por lo que describe los elementos del modelo que son más importantes para su construcción, los cimientos del sistema que son necesarios como base para comprenderlo, desarrollarlo y producirlo económicamente. Surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se refleja en los casos de uso.
- ✓ **Iterativo e Incremental:** RUP propone que cada fase se desarrolle en iteraciones. Una iteración involucra actividades de todos los flujos de trabajo, aunque desarrolla fundamentalmente algunos más que otros. Las iteraciones hacen referencia a pasos en los flujos de trabajo, y los incrementos, al crecimiento del producto. [7]

Estos conceptos - los de desarrollo dirigido por los casos de uso, centrado en la arquitectura, iterativo e incremental - son de igual importancia. La arquitectura proporciona la estructura sobre la cual guiar las iteraciones, mientras que los casos de uso definen los objetivos y dirigen el trabajo de cada iteración. [7]

El ciclo de desarrollo en la vida de un producto de software se divide en 4 fases: inicio, elaboración, construcción y transición, donde RUP propone que cada una de ellas se desarrolle en iteraciones y se basa en 2 disciplinas: Disciplina de Desarrollo, en la que se encuentran los flujos de trabajos

Capítulo 1: Fundamentación teórica

Modelamiento del Negocio, Requerimientos, Análisis y Diseño, Implementación y Pruebas; y Configuración y administración del cambio, Administración del proyecto y Ambiente, dentro de la Disciplina de Soporte. La fase de Inicio tiene como objetivo determinar la visión del proyecto, por lo que los Flujos de trabajo de mayor importancia en ella son Modelamiento del Negocio y Requerimientos. Durante la fase de elaboración se debe trabajar en la búsqueda de una arquitectura sólida para el proyecto, siendo el Flujo de Análisis y Diseño la prioridad en este caso. La fase de Construcción termina una vez que se haya obtenido la capacidad operacional del producto y para la fase de Transición se traza como objetivo obtener el release del mismo, o sea, el producto final, listo para ser entregado al cliente.

Cada una de estas etapas es desarrollada mediante el ciclo de iteraciones, que 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.

En la figura 1 se representa la arquitectura global de RUP, que tiene dos dimensiones: el eje horizontal representa el tiempo y muestra los aspectos del ciclo vital del proceso a medida que se desarrolla y el eje vertical representa las disciplinas que agrupan de forma lógica las actividades por naturaleza. [8]

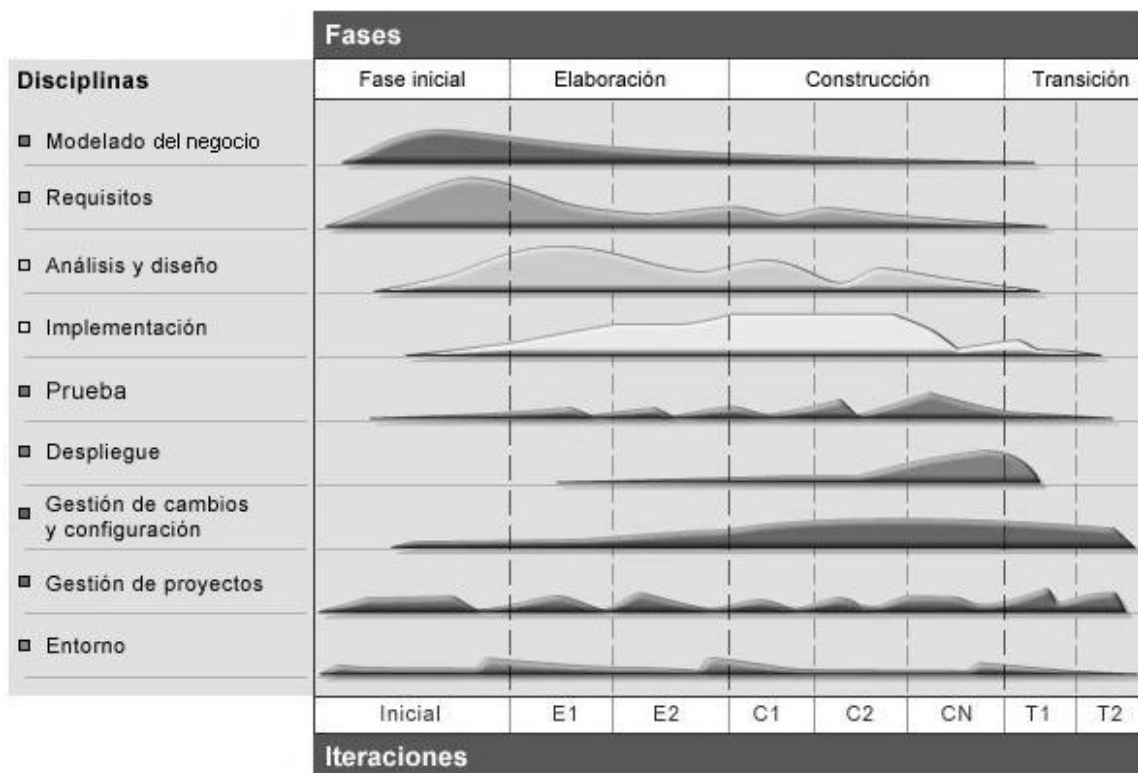


Ilustración 1: RUP en dos dimensiones

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

Es una metodología con un enfoque basado en modelos, utilizando un lenguaje bien definido para tal fin, el UML. Está preparado para desarrollar grandes y complejos proyectos, unifica los mejores elementos de metodologías anteriores y es orientado a objetos.

1.3.3 XP (Programación Extrema)

“Todo en el software cambia. Los requisitos cambian. El diseño cambia. El negocio cambia. La tecnología cambia. El equipo cambia. Los miembros del equipo cambian. El problema no es el cambio en sí mismo, puesto que sabemos que el cambio va a suceder; el problema es la incapacidad de adaptarnos a dicho cambio cuando éste tiene lugar.” [9]

XP (eXtreme Programming) nace como nueva metodología de desarrollo de software, aproximadamente en el año 1996, y desde entonces ha causado un gran revuelo entre el colectivo de programadores del mundo. Kent Beck, su autor, ha conseguido con sus teorías el respaldo de gran parte de la industria del software y el rechazo, por supuesto, de otra parte que no comparte la filosofía de XP.

La Programación Extrema es una metodología ágil o ligera de desarrollo de software que se basa en la simplicidad, la comunicación y la realimentación o reutilización del código desarrollado. Surgió como respuesta y posible solución a los problemas derivados del cambio en los requerimientos y para aumentar la productividad en proyectos de corto plazo, corto equipo y cuyo plazo de entrega era ayer.

Los objetivos de XP son muy simples, en primer lugar tiene la satisfacción del cliente: esta metodología trata de dar al cliente el software que él necesita y cuando lo necesita, respondiendo muy rápido a sus necesidades, incluso cuando los cambios aparezcan al final del ciclo de la programación. Es por esta razón que el cliente forma parte del equipo de desarrollo del software, ocupando un rol importante dentro de este. El segundo objetivo es potenciar al máximo el trabajo en grupo: tanto los jefes de proyecto, los clientes y desarrolladores, son parte del equipo y están involucrados en el desarrollo del software. [10]

Esta metodología cuenta con 4 fases de desarrollo, Planificación, Diseño, Desarrollo y Prueba. En la fase de Planificación se realiza lo que se conoce tradicionalmente como levantamiento de requisitos, en este caso a través de historias de usuario, que son tarjetas escritas por los propios clientes, tal y

como ellos ven las necesidades del sistema (sin describir la interfaz de usuario), y que cumplen el mismo propósito que los casos de uso que se definen en otras metodologías como RUP. Las iteraciones en XP suelen ser muy cortas y promueve que los programadores busquen soluciones y experimenten con ellas, programar sin miedo a descomponer el sistema permite un trabajo mucho más ágil y creativo. Por supuesto, esto no sería posible sin las buenas prácticas de control de versiones que para la XP son indispensables. En la segunda fase de desarrollo se trabaja para obtener un diseño lo más simple posible, con la menor cantidad de clases y métodos, tratando siempre de que el resultado obtenido funcione con todas las pruebas, no tenga lógica duplicada y manifieste cada intención importante para los programadores. Uno de los aspectos más significativo que tiene en cuenta XP para el Desarrollo es la refactorización (se basa en la reutilización de código, para lo cual se crean patrones o modelos estándares, siendo más flexible al cambio) y la programación en parejas, donde dos personas programan juntos la misma funcionalidad, siendo uno el que escribe el código y el otro quien lo rectifica y aporta nuevas ideas, y estas parejas cambian constantemente.

La ejecución automatizada de pruebas es un elemento clave de XP. Existen tanto pruebas internas (o test de unidad), para garantizar que el mismo es correcto, como pruebas de aceptación, para garantizar que el código hace lo que debe hacer. El cliente es el responsable de definir las pruebas de aceptación, ya que es la persona mejor cualificada para decidir cuál es la funcionalidad más valiosa. El hecho de que las pruebas sean automatizadas es el único modo de garantizar que todo funciona: desde el punto de vista de XP, si no hay pruebas, las cosas sólo funcionan en apariencia. El objetivo de las pruebas no es corregir errores, sino prevenirlos. Por ejemplo, las pruebas siempre se escriben antes que el código a probar, no después: esto aporta un gran valor adicional, pues fuerza a los desarrolladores a pensar cómo se va a usar el código que escriben, poniéndolos en la posición de consumidores del software. Elaborar las pruebas exige pensar por adelantado cuáles son los problemas más graves que se pueden presentar, y cuáles son los puntos dudosos. Esto evita muchos problemas y dudas que pudieran aparecer durante el desarrollo.

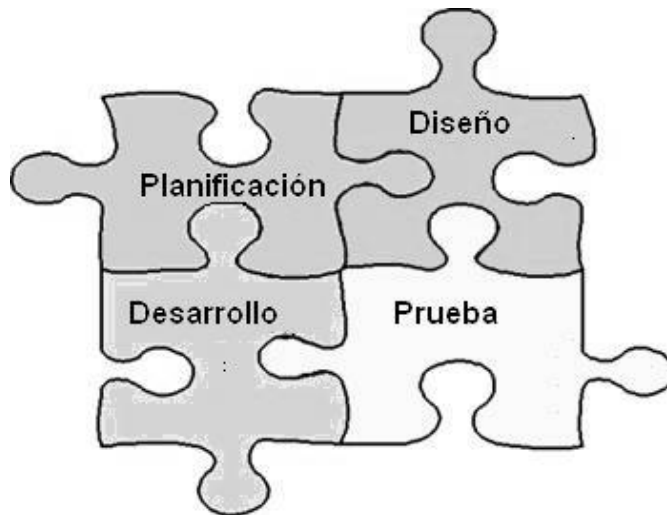


Ilustración 2: Fases de XP

La metodología XP, como todas las metodologías ágiles existentes hoy, basa sus principios en el Manifiesto Ágil, por lo que cumple que:

- ✓ Los individuos e interacciones son más importantes que los procesos y herramientas.
- ✓ Software que funcione es más importante que documentación exhaustiva.
- ✓ La colaboración con el cliente es más importante que la negociación de contratos.
- ✓ La respuesta ante el cambio es más importante que el seguimiento de un plan. [11]

1.3.4 Metodología Seleccionada

Una vez concluido el estudio realizado sobre las posibles metodologías de desarrollo de software que podrían haber guiado el proceso actual, RUP resultó ser la metodología más indicada en este caso, y es que, además de utilizar los Casos de Uso para guiar el proceso, y definir ciclos de trabajo, presta especial atención al establecimiento temprano de una buena arquitectura que no se vea fuertemente impactada ante cambios posteriores durante la construcción y el mantenimiento. Por otra parte su enfoque basado en modelos permite un mejor entendimiento entre clientes y desarrolladores y con ello una mejor calidad del producto final. Otra de las características que hacen de RUP una metodología muy exitosa y que influyó en la selección, es sin duda, su enfoque iterativo: la metodología parte del supuesto de que se trabajará en iteraciones cortas en tiempo y con metas muy claras. Cada iteración tiene entregables claros y en la medida de lo posible, el sistema debe ser funcional desde las primeras iteraciones de desarrollo. Si una funcionalidad no puede terminarse en una iteración, se terminará la

iteración sin ella, es más importante terminar la iteración con un sistema funcional y sin errores o detalles incompletos que un sistema que no puede ser evaluado.

La metodología XP, a pesar de las múltiples ventajas que ofrece como metodología ágil de desarrollo de software no se ajusta al proceso actual, en primer lugar porque una de las características de esta metodología es que el cliente forme parte del equipo de desarrollo, requisito que no se cumple debido a la lejanía del cliente; por otra parte, los requerimientos del software son totalmente estables, por lo que no haría falta XP, que trabaja sobre proyectos en constante cambio, y como último inconveniente está la poca experiencia del equipo de desarrollo, sobre todo de los programadores, situación que afectaría el concepto ágil de esta metodología.

Por su parte la metodología MultiMet no se centra en la especificación de la estructura al nivel de programación, llevando a un nivel elemental el análisis y diseño, con lo que fuerza al programador a confeccionar módulos que pueden resultar complejos. Carece de herramientas de sostén para la descripción del proceso de implementación y define la herramienta de desarrollo prácticamente en la etapa de implementación, decisión que podría implicar la reestructuración del proyecto, pues podría suceder que después de varios meses de trabajo no apareciera una herramienta que se ajustara al diseño propuesto o resultara más factible hacerlo con una que emplea una filosofía diferente a la planificada; implicando esto un pérdida de tiempo irreparable.

1.4 Lenguajes de modelado de software

Según el análisis realizado sobre la metodología RUP, metodología que por sus características fue seleccionada para conducir el desarrollo del software, se requiere de un lenguaje para realizar el modelado del sistema a implementar.

Antes de adentrarse en el análisis de los lenguajes de modelado, es necesario conocer que un modelo es una abstracción del sistema que permite especificarlo a un determinado nivel. Por otra parte modelado no es más que la acción de modelar, o sea, de ajustarse a un modelo.

En Ingeniería de Software (ISW) desarrollar modelos, implica representar los sistemas de software en cada una de sus partes, desde diferentes vistas: lógicas, estructurales, de comportamiento, de arquitectura, de ensamblaje; para de esta manera permitir el consenso en el lenguaje, el entendimiento del problema y las soluciones y la toma de decisiones de los equipos de desarrollo y los clientes.

La modelación de los sistemas permite propósitos como:

1. Para capturar y enumerar exhaustivamente los requisitos y el dominio de conocimiento, de forma que todos los implicados puedan entenderlos y estar de acuerdo con ellos.
2. Para pensar en el diseño de un sistema.
3. Para capturar decisiones del diseño en una forma mutable a partir de los requisitos.
4. Para generar productos aprovechables para el trabajo.
5. Para organizar, encontrar, filtrar, recuperar, examinar y corregir la información en grandes sistemas.
6. Para explorar económicamente grandes soluciones.
7. Para domesticar los sistemas complejos. [12]

1.4.1 Lenguaje Unificado de Modelado (UML)

UML es un lenguaje para especificar, construir, visualizar y documentar los artefactos de un sistema de software orientado a objetos (OO). Fue desarrollado por Ivar Jacobson, Grady Booch y James Rumbaugh, en la Corporación "Rational Software", en el año 1996 en su primera versión, 1.0, mejorado en la actualidad a una versión superior 2.0.

UML no es un método de desarrollo, no explica cómo pasar del análisis al diseño y de este al código. Es independiente del ciclo de desarrollo que se vaya a seguir, puede encajar en un tradicional ciclo en cascada, o en un evolutivo ciclo en espiral o incluso en los métodos ágiles de desarrollo.

UML aporta las siguientes ventajas:

- ✓ Mayor rigor en la especificación.
- ✓ Permite realizar una verificación y validación del modelo realizado.
- ✓ Se pueden automatizar determinados procesos y permite generar código a partir de los modelos y a la inversa (a partir del código fuente generar los modelos). Esto permite que el modelo y el código estén actualizados, con lo que siempre se puede mantener la visión en el diseño, de más alto nivel, de la estructura de un proyecto.

El Lenguaje Unificado de Modelado consta de 5 vistas: Vista de Casos de Uso, Vista de Implementación, Vista de Despliegue, Vista de diseño, y Vista de procesos; para la modelación de los productos, a través de un conjunto de diagramas distribuidos por cada una de estas vistas.

Es un lenguaje que surge como resultado de la unión de: Unified Method (Grady Booch, 1994), OMT (Grady Booch y James Rumbaugh, 1995) así como el Método OOSE (Ivar Jacobson, 1996). Posee varias herramientas CASE de modelado como son Rational Rose Enterprise Edition y Visual Paradigm.

Hoy en día los lenguajes de modelación de software están normalmente basados en el paradigma Orientado a Objetos. Este paradigma brinda un concepto uniforme para el desarrollo de software y numerosas ventajas como la especificación integrada de la estructura y sus comportamientos en la integración, a través de todas las fases de desarrollo. Varios lenguajes de modelación orientada a objetos han surgido señalando a UML como el último y el más aceptado por la comunidad desarrolladora de sistemas informáticos de todo tipo. Para aplicaciones con tecnología multimedia este lenguaje tiene la dificultad de no poder soportar todos sus aspectos de una forma adecuada e intuitiva, especialmente, las características del lenguaje para modelar los aspectos de la interfaz de usuario, no se aplican explícitamente en los entornos de aplicaciones con tecnología multimedia.

1.4.2OMMMA – L: Lenguaje para la modelación Orientada a Objetos de Aplicaciones Multimedia

OMMMA – L fue desarrollado en la Universidad de Paderborn, Alemania, en el año 2001, como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y el MVC (Modelo Vista Controlador) para la interfaz de usuario, reparando la dificultad que presenta UML para representar este tipo de aplicaciones.

El paradigma MVC es un modelo de arquitectura conocido en el desarrollo de aplicaciones orientadas a objetos que distinguen un componente modelo sosteniendo la funcionalidad del núcleo y los datos, un componente vista para mostrar la información al usuario y un componente controlador para manipular los eventos de interacción. Un mecanismo de propagación de cambios asegura la consistencia entre el modelo y la interfaz visual. Extendiendo este paradigma para multimedia a las peculiaridades de comportamiento estático y dinámico identificadas anteriormente, se obtiene MVC_{MM}, sobre el que se basa las especificaciones de OMMMA – L. [13]

OMMMA – L consta de cuatro vistas fundamentales en su modelación: vista lógica, vista de presentación espacial, vista de comportamiento temporal predefinido y vista de control interactivo.

- ✓ **Vista Lógica:** Modelada a través del diagrama de clases OMMMA-L, extendido del Diagrama de clases UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias. Divide en dos áreas dicho diagrama, una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.
- ✓ **Vista de Presentación espacial:** modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagramas tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización e interacción.
- ✓ **Vista de Comportamiento Temporal predefinido:** Modelada por el diagrama de secuencia de OMMMA-L, extendido a partir del diagrama de secuencias de UML. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos: marcas de inicio y fin de ejecución, marcas de activación y desactivación de demoras en objetos de tipo media, activación compuesta de objetos media para la agrupación de objetos concurrentemente activo.
- ✓ **Vista de Control Interactivo:** Modelada a través del diagrama de estado, extendido a partir del diagrama de estado de UML, sintácticamente igual a este, pero con la diferencia semántica de que en el orden de unir los controles interactivos y predefinidos no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombre de diagrama de secuencia en vez de diagramas de estado empotrados, lo que quiere decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia. [13]

Se puede decir entonces que OMMMA-L no es un lenguaje nuevo, sino una extensión del UML por lo que no es necesario aprenderlo, sino interpretar las características extendidas, centrados a la lógica de funcionamiento de una multimedia, que es por lo general, sencilla. Es robusto y altamente descriptivo, el proceso se refleja en todas sus etapas y hereda de RUP el ciclo de vida basado en iteraciones y el flujo de trabajo iterativo e incremental, centrado en casos de uso y en la arquitectura. [13]

1.4.3 ApEM-L: Lenguaje para la Modelación de Aplicaciones Educativas

“Un número de enfoques han sido propuestos para la modelación de aplicaciones multimedia. Predominantemente, estos se enfocan en la modelación de las relaciones temporales y la sincronización de las presentaciones multimedia. Algunos modelos más elaborados incluyen la interactividad. Otros se concentran en la estructura lógica y los conceptos de navegación para hipermedia.” [13]

ApEM – L se presenta como una extensión de UML, tomando como bases teóricas principales OMMMA – L (2001) y OCL – 2.0 (2003), lo que produce las siguientes ventajas:

- ✓ Puede utilizar para su representación todas las herramientas CASE que existen actualmente para la modelación de UML.
- ✓ Es un lenguaje que utiliza el estándar internacional OCL, para la modelación de la programación Orientada a Objetos.
- ✓ No modifica la semántica del lenguaje base UML, sino que trabaja en estereotipos restrictivos, por lo que a su vez produce modificaciones descriptivas y decorativas en la representación de los componentes del lenguaje base. [14]

ApEM-L surgió con el objetivo, en primer lugar, de desarrollar una extensión del lenguaje de modelado UML, tomándolo como base e incorporando a este, a través de sus mecanismos de extensión, los elementos fundamentales del proceso productivo UCI. De esta forma se produjo un lenguaje de propósito particular para la modelación de aplicaciones educativas. Por otra parte, este lenguaje no pretende convertirse en un método de desarrollo de aplicaciones con tecnología multimedia, sino en el área de la representación y la documentación de este tipo de aplicaciones. No circunscribirse a un proceso de desarrollo en específico, sino expresarlo de manera tal que pueda ser utilizado con cualquiera de los existentes, aunque sugiere la utilización de procesos de desarrollo iterativos, incrementales y basados en prototipos, que permitan la modelación de sistemas orientados a objetos, es otro de los objetivos de ApEM-L.

Como parte de los cambios propuestos por este lenguaje de modelado para el software educativo en la UCI y en Cuba, aparece una variante de solución al patrón MVC_{MM} para las aplicaciones educativas cubanas, Modelo – Vista – Controlador – Entidad (MVC-E), donde se dividen las responsabilidades de la clase modelo, concerniente al procesamiento y almacenamiento de la información persistente de las aplicaciones, en una nueva clase denominada *Modelo Entidad*, con dos tipos fundamentales, la clase

Modelo-Entidad-Media y *Modelo-Entidad-Persistente*. La primera de estas con la responsabilidad de agrupar las clases que identifican las medias y su árbol de jerarquía en la aplicación y la segunda tiene como responsabilidad la gestión de la información persistente, que antes sobrecargaba a la clase *Modelo* del patrón MVC original. Esta variación a su vez sustenta las características actuales de los sistemas multimedia como son: comunicación con bases de datos, archivos XML o sistemas externos. [14]

Los conceptos y modelos de ApEM – L pueden agruparse en:

- ✓ **Estructura lógica:** está compuesta por la vista estática y la vista de arquitectura. Cualquiera de los modelos presentados por ApEM – L define los conceptos claves de la aplicación que modela, las propiedades internas de estos y sus relaciones. Estos conceptos son modelados como clases, describiendo cada una un conjunto de objetos que almacenan información y se comunican para implementar su comportamiento.
- ✓ **Comportamiento dinámico:** el comportamiento de la aplicación está descrito por la vista de comportamiento, la cual está compuesta por los diagramas: de actividad, de secuencia, de colaboración y de estados, donde solo ha sido modificado el segundo de los listados anteriormente; adicionando una variable de tiempo donde quiera que sea necesario su especificación para un mejor entendimiento.
- ✓ **Gestión del modelo:** esta área es la que ha sufrido grandes cambios tanto en su carácter semántico como sintáctico, con la incorporación de estereotipos restrictivos en todos los diagramas a partir de nuevos conceptos incorporados a los diagramas de clases originales o básicas de UML. Se crean dos nuevos diagramas: el de estructura de la presentación y el de estructura de la navegación. [14]

ApEM – L se ha dividido en varias *vistas*: *Vista Estática*, *Vista de Arquitectura*, *Vista de Comportamiento* y *Vista de Presentación*, modelando cada una de estas construcciones que representan un aspecto del sistema. La división ha sido sobre la base de las áreas conceptuales ya presentadas: *estructura lógica*, *comportamiento dinámico* y *gestión del modelo*.

- ✓ **Vista Estática:** está compuesta por el *Diagrama de clases*, que se divide en dos grandes zonas, la de la izquierda dedicada al árbol jerárquico de las *clases modelo entidad medias* que representan los recursos mediáticos de la aplicación y en la zona de la derecha del diagrama las clases que controlan la lógica del negocio de la aplicación propiamente dicha. La zona de la derecha vuelve a subdividirse en cuatro zonas. La primera dedicada a las clases *vista*, la

contigua a esta y en el extremo superior derecho dedicada a las clases *controladoras*, inmediatamente debajo de esta sección, la destinada a las clases *modelo*, quedando una banda inferior derecha dedicada en su extremo derecho a las clases *modelo entidad persistentes* para el tratamiento de la información persistente de la aplicación; y en el extremo izquierdo las clases correspondientes al Lenguaje de Alto Nivel (HLL) con el que se programe (Lingo, ActionScript, C#, C++, Object Pascal, PHP, etc.).

- ✓ **Vista de Arquitectura:** Compuesta por el *Diagrama de componentes* y el *Diagrama de despliegue*. El último de los mencionados no sufre cambios en ApEM - L, no así el de componentes donde se incorporan restricciones en los tipos de componentes. Al seguir la arquitectura propuesta por el patrón MVC-E, normalmente los componentes podrán ser organizados por paquetes, que identificarían unidades físicas de encapsulamiento del código. Solo sería necesario luego un diagrama de componentes a un nivel de abstracción superior que represente como se comunican los distintos tipos de paquetes componentes del sistema.
- ✓ **Vista de Comportamiento:** Compuesta por cuatro diagramas: *de actividades*, *de estado*, *de secuencia* y *de colaboración*; siendo estos dos últimos generalizados en diagramas de interacción, pues como plantea la semántica de UML, representan la manera en la que los objetos de la aplicación intercambian mensajes para darle cumplimiento a sus responsabilidades. En ApEM – L solo ha sido modificado el diagrama de interacción de secuencia, con un estereotipo descriptivo y por ende decorativo para denotar el tiempo como variable de sumo interés en aplicaciones de este tipo.
- ✓ **Vista de Presentación:** Ha sido incorporada completamente a UML como parte de la extensión que realiza ApEM - L a este lenguaje base; con el fin de permitir utilizar la semántica original de dicho lenguaje en la construcción de estructuras lógicas de presentación y navegación, incorporando un conjunto de estereotipos restrictivos y descriptivos para una mejor modelación, construyendo el *diagrama de estructura de navegación* y el *diagrama de estructura de presentación*. La inserción de esta vista resuelve el principal problema existente en la modelación de aplicaciones educativas: los elementos de presentación y navegación, los cuales son de sumo interés para los desarrolladores de este tipo de aplicaciones. [14]

1.4.4 Selección del lenguaje de modelado

Luego del análisis realizado sobre los 3 lenguajes de modelado expuestos con anterioridad, se decidió que ApEM-L sería el lenguaje que ayudará a modelar la solución al problema planteado. Las razones que llevaron a esta decisión fueron, en primer lugar, que es un lenguaje extensivo de UML, por lo que utiliza sus fundamentos y su estilo de trabajo, tiene como bases teóricas OMMMA-L (2001) y OCL 2.0 (2003), lo que le permite modelar aplicaciones con tecnología multimedia que utilicen el paradigma Orientado a Objetos. Por otra parte, este lenguaje incluye nuevos elementos de modelado como son las Vistas de Presentación, artefactos que permiten representar componentes de una interfaz de comunicación con el usuario (pantalla), y que ayudan tanto a los desarrolladores como a todo el equipo de trabajo a entender y familiarizarse más con las aplicación, con su entorno y funcionalidades, dejando a un lado los casos de uso, que para este tipo de aplicaciones, donde lo más importante es el estímulo visual, no son del todo prácticos, por lo que también incluye un nuevo formato para la descripción textual, en este caso de sus Vistas de Presentación, que incorpora las descripciones de las medias y su localización, asemejándose un poco a lo que se conoce como el guión de contenidos para aplicaciones con esta tecnología. [15] Define nuevas clases y redefine el patrón MVC_{MM} , para representar exclusivamente y con características altamente descriptivas, los elementos de una multimedia, y lograr, como objetivo principal, la calidad que requiere un producto de software. Incorpora los diagramas de Estructura de Navegación y de Estructura de Presentación, con los que resuelve el principal problema existente en la modelación de aplicaciones de este tipo: los elementos de presentación y navegación, que son de gran importancia para el desarrollo de estas aplicaciones. Otra de las ventajas que presenta ApEM-L es que para su representación puede utilizar todas las herramientas CASE existentes en la actualidad para la modelación de UML, no siendo así para OMMMA-L, ya que los diagramas de Presentación que define este lenguaje no pueden ser modelados con las herramientas CASE existentes hoy. La selección de este lenguaje fue motivada además por la capacidad que tiene ApEM-L de abordar de forma más detallada y esclarecedora los conceptos relacionados con software con tecnología multimedia, de proporcionar una mejor visión de sus componentes a los desarrolladores, al equipo de trabajo en general y al cliente.

1.5 Tecnologías actuales para el desarrollo

Los sistemas multimedia han impulsado el desarrollo de las tecnologías de la información y de la comunicación. Gracias a las tecnologías de la información, la multimedia ha hecho posible superar la

idea de la información contenida en un texto donde se explora cada vez más en el campo de la comunicación audiovisual, de la transmisión de sensaciones y de innumerables novedades.

Las características específicas de la multimedia están determinadas por las confluencias de sustancias expresivas, pero también de soportes o “medios” que implican una determinada mediación técnica en el resultado final y, fundamentalmente, una determinada organización de la información no lineal, flexible y configurable (en diversos grados) por los usuarios del programa.

Herramientas de autor

Generalmente se plantean dos alternativas para desarrollar una aplicación con tecnología multimedia: una vinculada con el empleo de lenguajes de programación de propósito general (Visual Basic, Delphi, C++), y otra a través del empleo de sistemas de autor (AuthorWare, ToolBook, Director, Flash) [16], tal como se muestra en la ilustración 3.



Ilustración 3: Esquema de las formas de desarrollo de una multimedia

Las herramientas de autor (también denominados entornos de autor o lenguajes visuales) son aplicaciones informáticas que permiten elaborar sistemas multimedia. Ofrecen un entorno de trabajo que permite una programación basada en íconos, objetos y menús de opciones.

Es usual encontrar en los sistemas de autor estructuras especializadas en la solución de tareas clásicas de la multimedia como son:

- ✓ Carga y visualización de imágenes.
- ✓ Uso de efectos de transición en la navegación o presentación de la información.
- ✓ Ejecución y sincronización de archivos de sonido, video y animaciones.

- ✓ Definición de objetos visuales con funcionalidades específicas como son controles, objetos gráficos, contenedores y visualizadores de diferentes tipos de medias, ventanas y cuadros de diálogo.
- ✓ Recepción de entradas del usuario y asignación de acciones en calidad de respuesta por parte del sistema.

Estas tareas en lenguajes de propósito general requerirían de un significativo número de instrucciones, mientras que a través de los sistemas de autor generalmente se asocian a una operación o instrucción simple. [16]

Otra de las características de estos modernos lenguajes es su tendencia a acercarse al lenguaje natural desde el punto de vista sintáctico. Aparecen en muchos de ellos fenómenos no habituales en los lenguajes de programación de propósito general como son fenómenos de "sinonimia", el uso de artículos, preposiciones y pronombres, lo cual evidentemente eleva su asequibilidad. [16]

1.5.1 Macromedia Director

El nombre del programa está acorde con la interfaz del mismo. Se trata de crear una película (movie). Para ello, existen ventanas como el Reparto de "actores" (Cast), otra para el Montaje (Score), otra para los Guiones (Scripts) y otra para ver los resultados (Stage). Es decir, el usuario es como el director de la película, que controla todos sus aspectos.

Permite la combinación de texto, gráficos, sonido, animación y vídeo en un documento que se reproduce en el ordenador y que es presentado con múltiples detalles. La filosofía seguida por este programa es la de una línea de tiempo en la que irán sucediendo diferentes acontecimientos según se vayan necesitando. Este proceso no tiene por qué ser necesariamente lineal ni continuo, sino que permite detenerse en un punto del tiempo y saltar de un punto a otro en esa línea temporal. Director tiene un lenguaje propio de programación "Lingo", con el que se consigue comportamientos muy sofisticados, por defecto hay comportamientos y rutinas que ya vienen preparados. Este lenguaje agiliza los tiempos de desarrollo y ayuda a integrar a sus producciones una interactividad única y de alto nivel. [17]

Permite el uso de los llamados xtras, que no son más que pequeños programas desarrollados en otros lenguajes (C++) por otros usuarios, que le proporcionan a la aplicación infinidad de utilidades. También se pueden realizar otras funciones menos típicas del software de desarrollo multimedia, como el manejo de bases de datos, colas, listas y trabajo con variables. Crea una amplia gama de aplicaciones

3D sumamente interactivas, incluyendo juegos, comercialización electrónica, aprendizaje electrónica y demostraciones de productos.

Una desventaja que presenta esta herramienta, a pesar de las múltiples facilidades y opciones que brinda para el desarrollo de aplicaciones con tecnología multimedia, es que no es multiplataforma.

1.5.2 ToolBook

ToolBook brinda un ambiente de programación orientada a objeto para de esta manera construir proyectos, libros, con el fin de presentar gráficamente información como textos, animaciones, sonidos, dibujos e imágenes digitalizadas a color. Además ofrece interfaces gráfica Windows. [18]

Tiene dos niveles de trabajo:

Instructor: Herramienta de autor orientada a personal especializado en Informática y con un amplio dominio de técnicas de programación.

Assistant: Sistema de autor orientado a personal docente no especializado en Informática y sin dominio de técnicas de programación.

En la actualidad ToolBook Instructor es una herramienta de autor concebida para el desarrollo de aplicaciones educativas distribuibles en disquetes, CD-ROM, redes locales y globales. Además de una amplia gama de software multimedia limitado prácticamente solo por la creatividad del autor, es posible crear cursos dinámicos e interactivos dotados de los más modernos recursos multimedia como sonido, animaciones, video y gráficos.

La existencia de un poderoso entorno de programación basado en el lenguaje orientado a objetos y dirigido por eventos denominado OpenScript, dota al sistema de la flexibilidad necesaria para desarrollar la más amplia gama de aplicaciones bajo el sistema operativo Windows. (Windows 9x, Windows ME, Windows NT, Windows 2000). [19]

Una interesante característica de este asistente es que permite ser configurado a gusto propio, de forma que la siguiente vez que se tenga que crear un nuevo proyecto, el asistente trabajará acorde a lo que se haya especificado.

1.5.3 Macromedia Flash 8.0

Macromedia Flash, es una herramienta de autor, que utiliza gráficos vectoriales e imágenes, sonido, lenguaje orientado a objetos: ActionScript, flujo de vídeo y audio bidireccional. Permite a los diseñadores y desarrolladores integrar las diferentes medias en experiencias dinámicas que le permiten al cliente adentrarse en su vivencia y que producen resultados superiores para marketing y presentaciones interactivas, aprendizaje electrónico e interfaces de usuarios de aplicaciones.

Los archivos de Flash aparecen muy a menudo como animaciones en páginas Web y sitios Web multimedia. Son también ampliamente utilizados en anuncios de la Web. Flash es independiente del navegador y el plug-in es universal, por lo que las animaciones diseñadas con este programa se verán casi idénticamente en cualquier plataforma y navegador.

La única desventaja que tienen las películas Flash, es que para poder visualizarlas, es necesario tener instalado el plug-in, aunque por el impacto que ha tenido esta tecnología, a partir de la versión 4.0 de los navegadores, el plug-in ya se incluye dentro de la instalación. [20]

Macromedia ha ampliado Flash más allá de las animaciones simples, convirtiéndola en una herramienta de desarrollo completa, para crear principalmente elementos multimedia e interactivos para Internet.

Características principales:

- ✓ Interfaz gráfica amigable, potente y sencilla de usar.
- ✓ Soporta vídeo con nuevas funcionalidades.
- ✓ Carga dinámica de imágenes, video y sonido.
- ✓ Previsualización de animaciones.
- ✓ Ayuda tanto para la programación como para el diseño de animaciones.
- ✓ Incluye componentes ya creados que te pueden ayudar a la hora de hacer tus animaciones.
- ✓ Puede interactuar con una base de datos.
- ✓ Librería de símbolos.
- ✓ Soporte de audio MP3.
- ✓ Interacción con otros lenguajes como XML.

1.5.4 Selección de la herramienta para el desarrollo de la aplicación

La herramienta de autor seleccionada para el desarrollo de la aplicación es Flash 8.0. Dicha selección está justificada por el hecho de que soporta flujo progresivo por defecto (los fotogramas de la animación se cargan individualmente y pueden ser mostrados antes de que se cargue el archivo entero), emplea gráficos vectoriales, por lo que los archivos tienen un menor tamaño y consumen menos ancho de banda al ser transmitidos que las imágenes en mapa de bits, su reproductor está considerado como uno de los más rápidos en iniciarse, soporta características avanzadas para la carga de datos a través de XML, sonido mp3, imágenes JPEG y otras películas de Flash, es multiplataforma y cuenta con una interfaz sencilla y fácil de usar. Permite detectar errores ortográficos en los textos que se incluyan en las animaciones, incluye dicha revisión ortográfica en varios idiomas y posibilita además definir un diccionario propio para realizar esta función. Otra de las novedades que incluye Flash 8 es la incorporación de nuevos componentes para el desarrollo de las aplicaciones, además permite insertar videos transparentes y aplicarle los efectos gráficos, posibilitando que se modifiquen los canales alpha de los mismos. El aumento de hasta 10 veces más en la velocidad de las películas también es una ventaja que proporciona Flash 8.

Otras de las características que presenta esta nueva versión de Flash y que influyeron en su selección como herramienta de desarrollo son:

- ✓ Diseños más atractivos: Permite el uso de efectos visuales que facilita la creación de animaciones, presentaciones y formularios más atractivos y profesionales, así mismo, pone a disposición mecanismos para hacer el trabajo más cómodo y rápido, tales como la existencia de filtros y modos de mezcla añadidos en esta versión.
- ✓ Optimización de fuentes: Incorpora también opciones de legibilidad para fuentes pequeñas, haciendo la lectura de los textos más agradables y de alta legibilidad. Además de poder modificar la optimización, Flash permite también la selección de configuraciones preestablecidas para textos dinámicos y estáticos.
- ✓ Mayor potencia de animación: Permite un mayor control de las interpolaciones habilitando un modo de edición desde el que se podrá modificar independientemente la velocidad en la que se apliquen los diferentes cambios de rotación, forma, color, movimiento, etc., de las interpolaciones.

- ✓ Mayor potencia gráfica: Evita la repetición innecesaria de la representación de objetos vectoriales simplemente señalando un objeto como mapa de bits. Aunque el objeto se convierta al formato de mapa de bits, los datos vectoriales se mantienen tal cual, con el fin de que, en todo momento, el objeto pueda convertirse de nuevo al formato vectorial.
- ✓ Mejoras en la importación de video: Para facilitar el resultado con formatos de video, incluye un códec independiente de calidad superior capaz de competir con los mejores códec de videos actuales con un tamaño de archivo mucho más pequeño. Además de una gran posibilidad de revestimientos para los controles de éste en la película. [21]

Lenguajes de Programación

1.5.5 Lenguaje XML (Lenguaje de Marcas Extensible)

XML es la sigla del inglés eXtensible Markup Language (lenguaje de marcado ampliable o extensible) desarrollado por el Consorcio de la Web (W3C), organización que establece los estándares de la Web. Es el resultado de la evolución de los lenguajes SGML (Standard Generalized Markup Language) y HTML (Hypertext Markup Language). Es un lenguaje de marcado que fue diseñado para representar y describir datos. Su principal característica es que, a diferencia de HTML, no tiene marcas o etiquetas predefinidas sino que debe crearlas uno mismo.

Es una tecnología sencilla que tiene a su alrededor otras que la complementan y la hacen mucho más grande y con posibilidades mayores. Tiene un papel muy importante en la actualidad ya que permite la compatibilidad entre sistemas para compartir la información de una manera segura, fiable y fácil. [22]

Algunas de las ventajas que ofrece esta nueva tecnología son:

- ✓ Una vez diseñado un lenguaje y puesto en producción, es posible extenderlo con la adición de nuevas etiquetas de manera de que los antiguos consumidores de la vieja versión todavía puedan entender el nuevo formato, esto es posible gracias a su capacidad de extensión.
- ✓ El analizador es un componente estándar, no es necesario crear un analizador específico para cada lenguaje. Esto posibilita el empleo de uno de los tantos disponibles.
- ✓ Si un tercero decide usar un documento creado en XML, es sencillo entender su estructura y procesarlo. Mejora la compatibilidad entre aplicaciones.
- ✓ Permite la portabilidad de los datos, por lo que puede considerarse como una base de datos.

- ✓ No requiere licencias, es independiente de la plataforma y tiene un amplio soporte.

Los documentos XML pueden ser leídos por la herramienta de autor Flash, siendo ésta una manera de añadir contenidos a una película flash una vez terminada, exportada y publicada. La gran ventaja es que es muy fácil modificar el XML, añadir o quitar información, debido fundamentalmente a que es elaborado en ficheros de texto simple. Flash puede comunicarse con el exterior mediante el protocolo HTTP, puede recibir y enviar datos XML y puede incluso hacer una conexión a un puerto determinado utilizando sockets XML. Una de las principales razones por las que se recomienda la integración de XML con Action Script (lenguaje de programación de Flash) es para evitar la recurrente edición del archivo fuente (.fla) cada vez que se necesite introducir un cambio en el contenido.

Lectores dinámicos de noticias, sistemas de gestión de weblogs y foros son algunas de las aplicaciones donde el uso del lenguaje XML se hace prácticamente imprescindible, no obstante su campo de acción no se limita únicamente a las aplicaciones de carácter dinámico.

En la actualidad, XML se usa habitualmente para transferir datos entre diferentes aplicaciones de bases de datos. La mayoría de los Sistemas de Administración de Bases de Datos permiten exportar datos como archivos XML.

XML se ha convertido hoy en el lenguaje de moda para representar e intercambiar datos, entre aplicaciones, entre aplicaciones y personas, y entre empresas.

1.5.6 Action Script 2.0

ActionScript es un lenguaje de programación orientado a objetos (POO) y un lenguaje de scripts integrado en Flash, lanzado con la versión 4 de esta herramienta, y desde entonces hasta ahora, ha ido ampliándose poco a poco, hasta llegar a niveles de dinamismo y versatilidad muy altos. Posibilita el desarrollo de complejas aplicaciones multimedia y sitios web dinámicos bajo esta tecnología. Su campo de aplicación crece cada día, abarcando aplicaciones tan diversas como el desarrollo de juegos, simulaciones, presentaciones interactivas y animación dinámica con sorprendentes efectos visuales. [23]

Como todo lenguaje de script no requiere la creación de un programa completo para que la aplicación alcance los objetivos.

La segunda versión de este lenguaje (ActionScript 2.0) realmente es un increíble avance, implementa una estructura más fiel al modelo de POO, incorporando un nuevo set de elementos y características

especiales que permiten al programador definir las clases y operar en base a ellas de una manera más práctica y adecuada. Permite examinar o modificar las propiedades de los elementos de una película, tales como: cambiar el color y la localización de un objeto, reducir el volumen de un sonido, campos de texto que permiten a los usuarios entrar datos a la película como en un formulario y modificar las propiedades repetidamente produciendo comportamientos únicos como son los movimientos basados en la física y la detección de colisiones.

1.6 Herramienta de modelado de software

1.6.1 Rational Rose

Rational Rose es la herramienta CASE que comercializan los desarrolladores de UML (Booch, Rumbaugh y Jacobson) y que soporta de forma completa la especificación del UML, ventaja que permite a los arquitectos de software y desarrolladores visualizar el sistema completo utilizando un lenguaje común. Rose es una herramienta que ayuda a la comunicación entre los miembros del equipo, a monitorear el tiempo de desarrollo y a entender el entorno de los sistemas. Es una de las más poderosas herramientas de modelado visual para el análisis y diseño de sistemas basados en objetos. Se utiliza para modelar un sistema antes de proceder a construirlo. Cubre todo el ciclo de vida de un proyecto: concepción y formalización del modelo, construcción de los componentes, transición a los usuarios y certificación de las distintas fases. Permite mantener la consistencia de los modelos del sistema software, realiza chequeo de la sintaxis UML y genera documentos automáticamente.

Otra ventaja de Rose es que los diseñadores pueden modelar sus componentes e interfaces en forma individual y luego unirlos con la función de integración de modelos con otros componentes del proyecto. [24]

1.7 Conclusiones del capítulo

Se expusieron los elementos teóricos que fundamentarán el desarrollo del proceso de software, quedando definida RUP como metodología de desarrollo. Se utilizará ApEM-L como extensión de UML para la integración y modelación de especificaciones de sistemas con tecnología multimedia basados en el paradigma orientado a objetos. Además se seleccionó la herramienta Rational Rose para el modelado del software y como herramienta de autor para la creación de la aplicación, Macromedia Flash 8.0, con lenguaje de programación Action Script 2.0 y XML para el tratamiento e intercambio de datos.

Capítulo 2

Descripción de la solución propuesta

2.1 Introducción

El presente capítulo está dedicado a describir la solución que se propone para resolver el problema planteado en la investigación, solución que tiene como guía de desarrollo al Proceso Unificado.

La identificación de los conceptos asociados al dominio y de los requisitos funcionales y no funcionales del sistema, son algunas de las temáticas que estarán presentes en el capítulo. Por otra parte, se pretende modelar el dominio a partir de los conceptos identificados, representar la Vista de Gestión del Modelo, realizar los diagramas de estructura de navegación, describir las vistas de presentación identificadas en el producto, y, de forma general, describir el sistema propuesto.

2.2 Solución propuesta

La solución que se propone para resolver el problema planteado de la escasa formación política en los dirigentes de parroquias venezolanos, es la realización de una aplicación con tecnología multimedia en una de sus modalidades: libro electrónico, que llevará por título “Colección Antonio Guiteras y Carlos Aponte: seis libros para la formación de conciencia política”. Será una colección formada exactamente por 6 libros electrónicos interactivos: El Vuelo de la Mariposa Amarilla, La Gaviota Humanocrática, Rumbo al Socialismo, Un Grano de Maíz I, Un Grano de Maíz II y Un Grano de Maíz III, todos con el objetivo de difundir información política entre estos dirigentes, informar sobre la situación política nacional e internacional, inculcar la conciencia del deber social, la formación de valores ya perdidos por la sociedad, y formar líderes con una conciencia política acorde al proceso revolucionario que hoy se vive.

El producto está dirigido precisamente a los dirigentes de las parroquias venezolanas y a líderes políticos, personas que poseen un nivel de escolarización mínimo de 6to grado, que pueden estar insertados en muchos casos a las misiones de superación o continuidad de estudio y que pertenecen a diferentes estados o zonas del país. Permitirá acceder al contenido de cada libro a través de un menú

Capítulo 2: Descripción de la solución propuesta

de tópicos, así como realizar búsquedas de palabras o frases dentro del libro, e imprimir el texto correspondiente a una página, a un tópico o a un libro en general. Brindará la opción de escuchar un tema musical de fondo una vez iniciada la presentación, dando la posibilidad de desactivar y activar este sonido cada vez que se desee.

La figura precedente muestra el diseño de la portada del producto que se implementará como solución propuesta.



Ilustración 4: Pantalla principal de la aplicación

2.3 Descripción de un modelo de dominio

Un sistema, por pequeño que sea, generalmente es complicado. Por eso se necesita dividirlo en piezas si se pretende comprenderlo y gestionar su complejidad. Esas piezas se pueden representar a través de modelos que permitan abstraer sus características esenciales. Dependiendo de la situación que se necesite modelar, existen varias alternativas para llevar a cabo este proceso, RUP, como metodología de desarrollo propone la realización de un modelo de negocio para el caso en el que los

procesos dentro del entorno estén claramente definidos, y la realización de un modelo de dominio para los escenarios en los que no puedan identificarse tales procesos del negocio.

Analizando que para la modelación del sistema con tecnología multimedia propuesto no se lograron definir procesos específicos en el entorno, solamente identificar conceptos y objetos relacionados con el mismo, se propone la realización de un modelo de dominio.

El Modelo de Dominio (o Modelo Conceptual) es una representación visual de los conceptos u objetos del mundo real, significativos para un problema o área de interés. Captura los tipos más importantes de objetos en el contexto del sistema. Los objetos del dominio representan las cosas que existen o los eventos que suceden en el entorno en el que trabaja el sistema. Representa conceptos del mundo real, no de los componentes de software. [7]

Su representación en UML es a través de un Diagrama de Clases en el que se muestran los conceptos u objetos del dominio del problema (clases conceptuales), las asociaciones entre ellas y los atributos que poseen. En el Modelo de Dominio no se muestra comportamiento, por lo que las clases conceptuales pueden tener atributos, pero no métodos.

2.3.1 Identificación de los conceptos del dominio

Dirigente de Parroquia Venezolano: Categoría con la que se reconoce a los líderes de las parroquias (unidad político-territorial en la que se dividen algunos municipios) en Venezuela. Son las personas más beneficiadas con la colección de libros, por lo que constituirán los usuarios principales del sistema a desarrollar, los que accederán al mismo e iniciarán de alguna forma la interacción.

Libro: Conjunto de tópicos o epígrafes relacionados con un tema en general y que aparecen como un todo, unificados bajo un mismo título y autor.

Tópico: Epígrafe o tema específico dentro de un libro.

Contenido: Texto que contiene la multimedia, información específica relacionada con un tema.

Colección de libros: Conjunto de libros relacionados con un tema específico, en este caso, el tema general de los libros que conforman la colección, es el acontecer político en Venezuela y en el mundo.

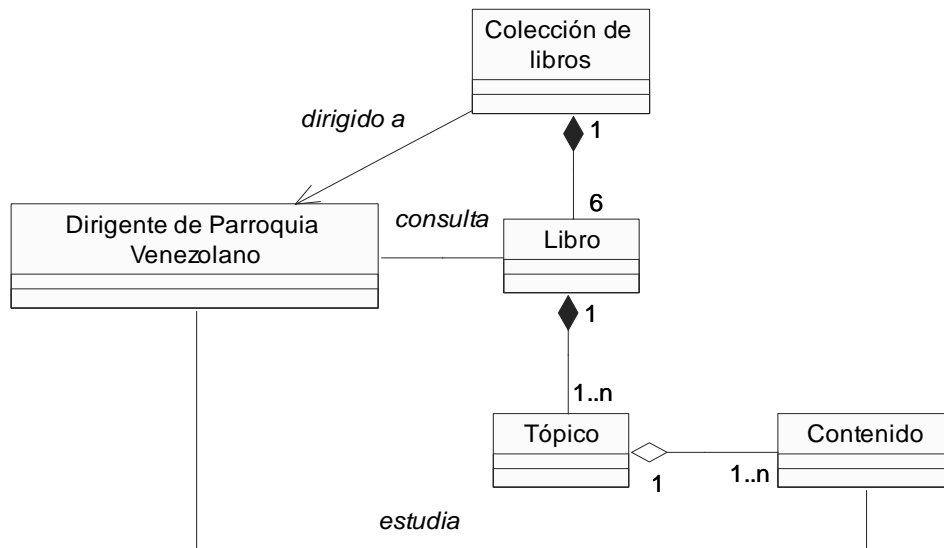


Ilustración 5: Modelo de dominio

2.4 Requerimientos del sistema

“La captura de requisitos es el proceso de averiguar, normalmente en circunstancias difíciles, lo que se debe construir”. [7]

Son muchas las causas que pueden llevar al fracaso en el desarrollo de un software, pero sin dudas, una de las que más afectaciones provoca, y que con más frecuencia se encuentra hoy en la inmensa industria del software, es la errónea identificación de los requerimientos, es decir, las incongruencias entre lo que el usuario quería, lo que realmente necesitaba, lo que interpretaba cada miembro del equipo de desarrollo y lo que realmente se obtiene.

Antes de comenzar a diseñar e implementar un software, es necesario saber qué se quiere que haga dicho software, dejar bien claro lo que los interesados (clientes) desean de éste. Para obtener exitosamente esta información y desarrollar con calidad el producto, ha surgido una disciplina encargada de definir lo que se desea producir, la Ingeniería de Requerimientos de Software. Esta nueva disciplina establece los servicios que un sistema debe suministrar así como las restricciones bajo las cuales debe operar. Lograr un entendimiento entre los clientes que solicitan el software y el equipo de desarrollo del mismo, es la clave que se tiene hoy para alcanzar el éxito en la producción de software, he aquí la importancia que se le debe dar a la Ingeniería de Requerimientos de Software, es decir, a la identificación de los requerimientos, como parte del proceso de desarrollo de software.

Es necesario aclarar que un requerimiento responde a una condición o capacidad que tiene que ser alcanzada o poseída por un sistema o componente de un sistema para satisfacer un contrato. También se pueden encontrar otras definiciones para este término, tal es el caso de la definición dada por la IEEE Standard Glossary of Software Engineering Terminology, que define como requerimiento a una condición o capacidad que necesita un usuario para resolver un problema o lograr un objetivo.

Los requerimientos pueden clasificarse en dos grandes grupos: funcionales y no funcionales.

Los **requerimientos funcionales** definen las condiciones que el sistema debe cumplir, o sea, las funciones que debe realizar para proporcionar un determinado resultado al usuario.

Los **requerimientos no funcionales** responden a las cualidades o propiedades que el sistema debe tener, es decir, son las características que permiten obtener un producto atractivo, usable, rápido o confiable. Pueden marcar la diferencia entre un producto bien aceptado y uno con poca aceptación.

2.4.1 Requerimientos funcionales

R1: Mostrar pantalla principal del producto.

R2: Permitir la navegación por cada uno de los 6 libros que componen el producto.

R3: Mostrar el contenido de cada uno de los 6 libros que componen el producto.

R3.1: Mostrar en la pantalla de presentación o inicio de cada libro: título, autor y temas o tópicos del libro.

R3.2: Mostrar el texto del contenido del tema o tópico seleccionado por el usuario.

R4: Mostrar palabras claves o calientes.

R4.1: Mostrar a través de ventanas emergentes, información adicional relacionada con la palabra caliente seleccionada por el usuario.

R5: Permitir la navegabilidad entre las pantallas internas de cada tema o tópico a través de botones característicos de “atrás” y “adelante”.

R6: Permitir realizar búsquedas en el libro seleccionado por el usuario.

R6.1: Realizar búsqueda por determinados criterios: palabra, frase.

R6.2: Mostrar en una ventana emergente los resultados de la búsqueda.

R6.3: Permitir acceder directamente al texto o pantalla que contiene la información deseada a través de la selección por el usuario.

R7: Permitir imprimir el texto.

R7.1: Imprimir página.

R7.2: Imprimir un tema o tópico de un libro determinado.

R7.3: Imprimir un libro determinado.

R8: Permitir control de la música.

R8.1: Permitir activar y desactivar la música.

R9: Permitir la salida de la aplicación desde cualquier pantalla en que se encuentre el usuario cuando esta sea solicitada.

R9.1: Mostrar una ventana de confirmación de salida.

R10: Mostrar créditos una vez que el usuario confirme que desea salir.

2.4.2 Requerimientos no funcionales

1. Requerimientos de apariencia o interfaz externa

- ✓ El sistema tendrá una interfaz sencilla, intuitiva y amigable para sus usuarios.
- ✓ El sistema proporcionará claridad y correcta organización de la información, permitiendo la interpretación correcta e inequívoca de esta, facilitando el uso de la aplicación.
- ✓ El sistema implementará la ejecución de acciones de una manera rápida, minimizando los pasos a ejecutar en cada proceso.
- ✓ El diseño de la interfaz gráfica deberá garantizar la distinción visual entre los elementos del sistema.

2. Usabilidad

- ✓ Los usuarios que utilizarán el sistema deberán tener conocimiento básico del manejo de la computadora, así como del trabajo con sistemas operativos visuales.
- ✓ La aplicación deberá poseer una interfaz y navegación asequibles y funcionales tanto para usuarios expertos como para los que no tienen conocimientos profundos de informática.

3. Requerimientos de Portabilidad

- ✓ El producto podrá ser usado bajo el sistema operativo Windows.

4. Requerimientos de Software

- ✓ Computadora personal con sistema operativo Microsoft Windows 98 o superior.
- ✓ Macromedia Flash Player 8.0.

5. Requerimientos de Hardware

- ✓ Procesador Pentium II o superior.
- ✓ 64 MB de RAM (Mínimo).
- ✓ Kit de Multimedia y Mouse.
- ✓ Lector de CD-ROM.

6. Restricciones en el diseño y la implementación

- ✓ La herramienta de desarrollo de la aplicación será Macromedia Flash 8.0.
- ✓ El lenguaje de programación será Action Script 2.0.

7. Interfaz de hardware

- ✓ El sistema interactuará con una impresora para imprimir los diferentes contenidos como respuesta a las funcionalidades del sistema. Estas deberán tener una asignación de papel y tonel que les permita ejecutar todas las peticiones que la aplicación les haga.

2.5 Vista de Gestión del Modelo

Dentro de los muchos aportes que ha brindado ApEM-L al Lenguaje Unificado de Modelado (UML), se encuentra la definición de un nuevo modelo, la Vista de Gestión del Modelo (VGM), que representa fundamentalmente, la división de la aplicación por subsistemas y la relación entre ellos. Para el caso específico de la aplicación que se desarrolla como parte de este trabajo, se han identificado 4 subsistemas, definidos por las características y funcionalidades de las vistas que lo conforman.

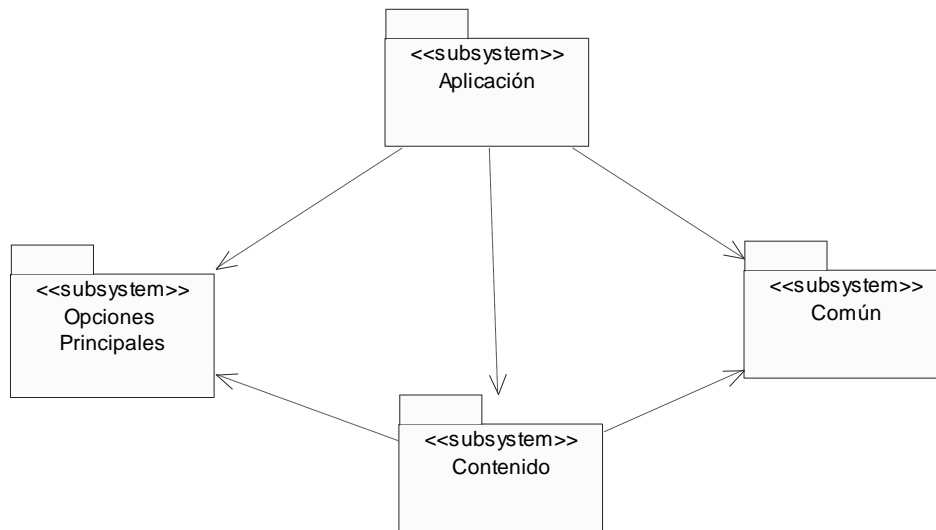


Ilustración 6: Vista de Gestión del Modelo

2.6 Diagramas de Estructura de Navegación

“Los diagramas clásicos provistos por UML, como el diagrama de clases o el de estado no son suficientes para modelar aspectos de los sistemas multimedia, por ejemplo, para modelar el espacio de navegación y para representar este modelo gráficamente.” [25]

ApEM-L define un nuevo diagrama para resolver este problema, el Diagrama de Estructura de Navegación (DEN), el cual identifica nuevas clases que permiten representar con más detalles la navegación en el producto. Las clases son: *clase menú*, *clase índice*, *clase consulta* y *clase botón*, *clases modelo-entidad-media texto* y *clase modelo-entidad-media imagen*.

Capítulo 2: Descripción de la solución propuesta

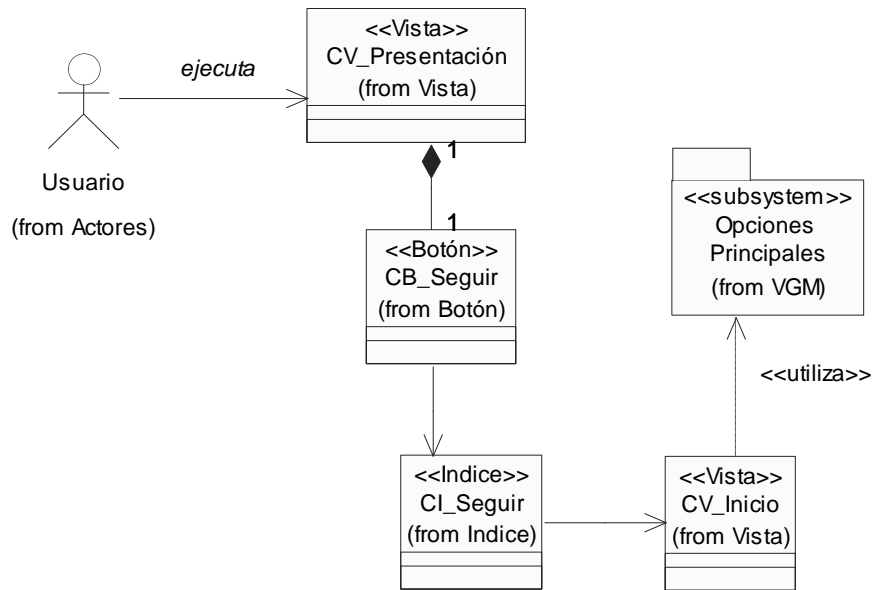


Ilustración 7: DEN Subsistema Aplicación

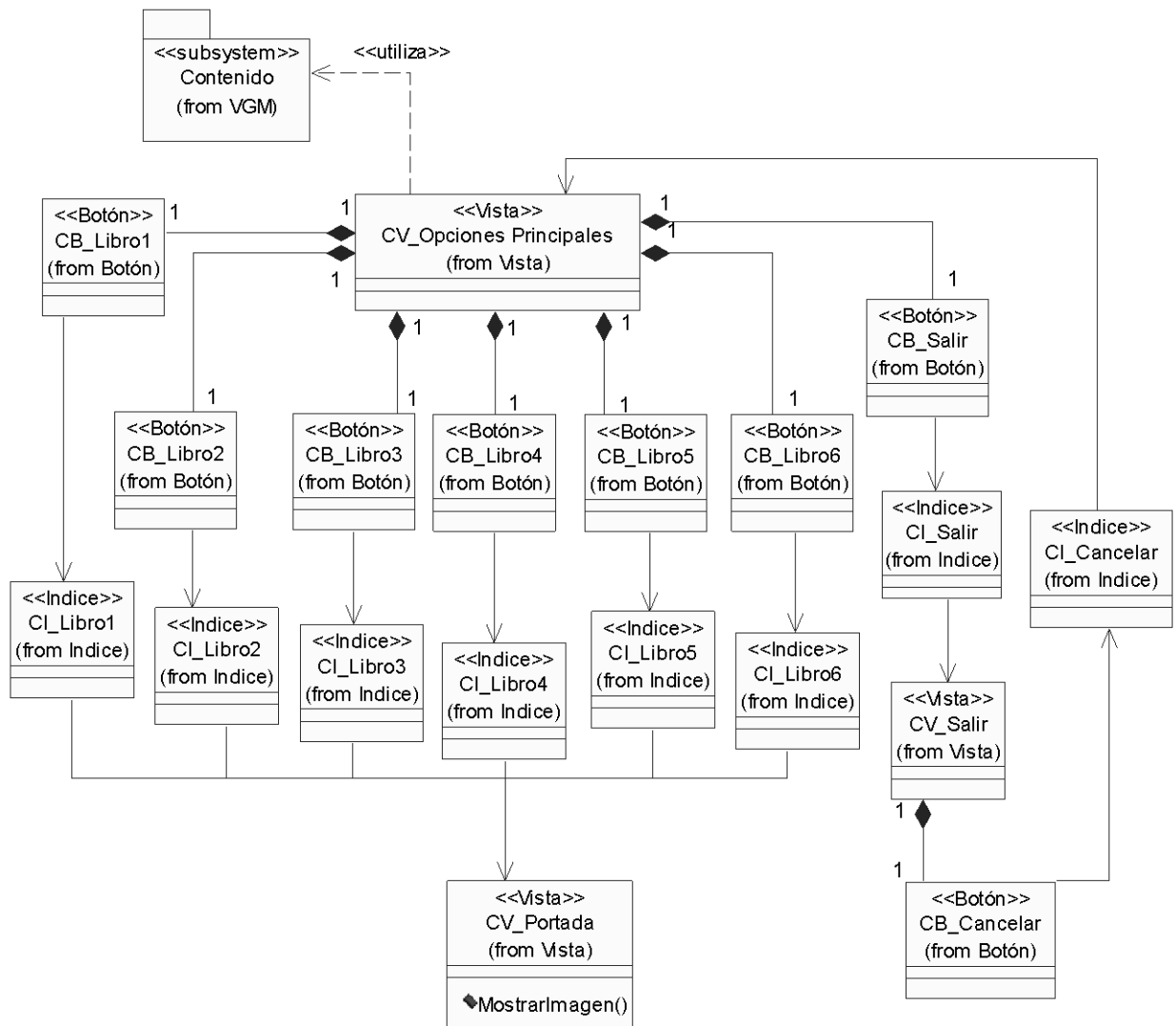


Ilustración 8: DEN Subistema Opciones Principales

Capítulo 2: Descripción de la solución propuesta

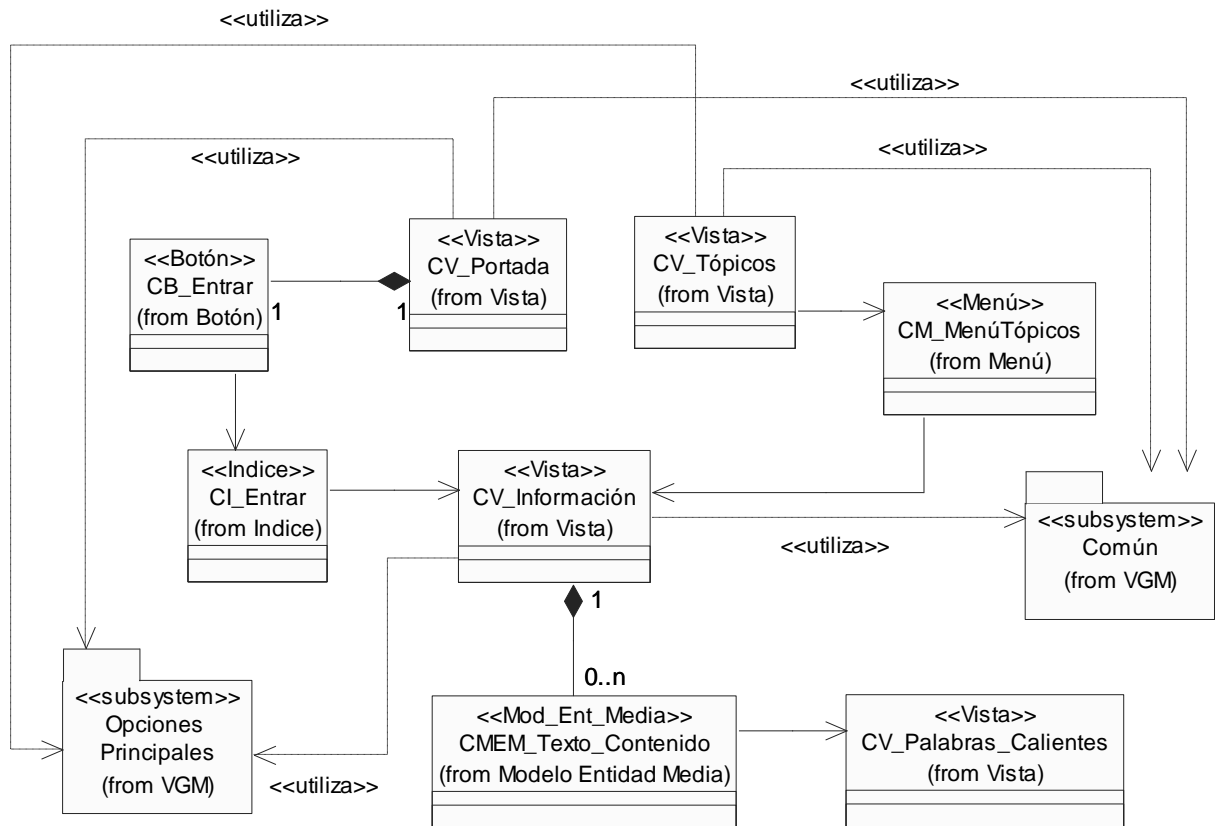


Ilustración 9: DEN Subsistema Contenido

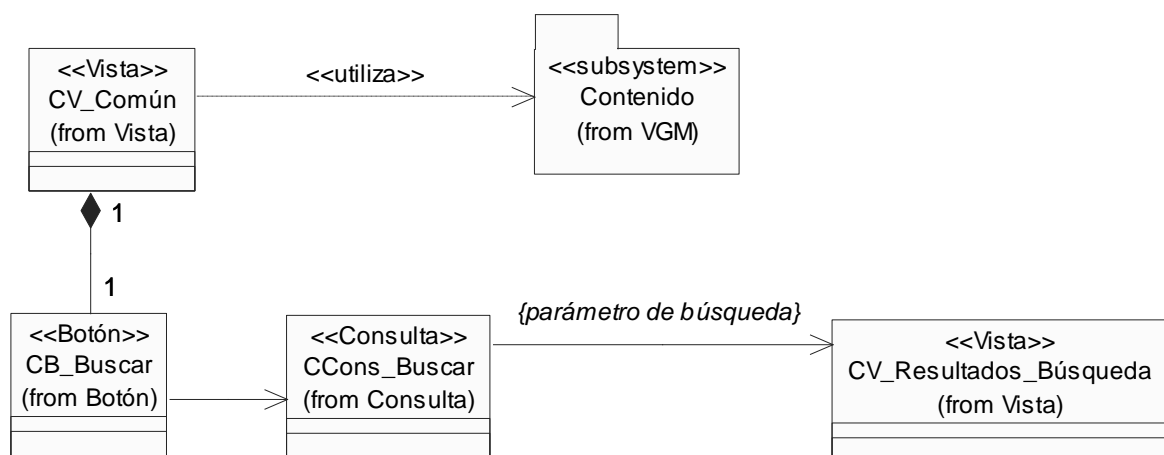


Ilustración 10: DEN Subsistema Común

2.7 Justificación de los actores del sistema

Actor: Usuario

Descripción: Representa a todas las personas que interactúan con la aplicación, personas que serán fundamentalmente dirigentes de parroquias venezolanos. Es el que obtendrá los beneficios de la interacción, pues se nutrirá de los contenidos presentes en los libros.

2.8 Descripción textual de las Vistas de Presentación

Tabla 1: Descripción textual de la Vista Presentación

Descripción textual de la Vista de Presentación <i>“Presentación”</i>		
Actores de la Vista de Presentación	Usuario	
Propósito	Permitir al usuario visualizar la presentación de la aplicación y acceder a la pantalla principal de la misma.	
Objetivos Pedagógicos	-	
Resumen	La vista se inicia cuando el usuario levanta la aplicación. El sistema muestra la animación que da comienzo al producto y brinda la opción de acceder a la pantalla principal del mismo, siendo la selección de esta opción la culminación de la vista.	
Vistas asociadas	Vista <i>“Inicio”</i> (<i>Subsistema Aplicación</i>)	
Referencias	R1	
Precondiciones	-	
Poscondiciones	Se mostró la presentación de la aplicación.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario ejecuta la aplicación.		
	2. Se reproduce el tema musical de fondo en la aplicación.	
	3. Se muestra la animación de la presentación.	

Capítulo 2: Descripción de la solución propuesta

	4. Muestra un texto informativo sobre la aplicación, su objetivo y la misión que tiene, y brinda la opción de acceder a la pantalla principal de la misma.	
		4.1 La opción de acceder a la pantalla principal aparecerá en el borde inferior del texto, centrada y representada por una flecha que indica seguir.
5. Selecciona la opción de ir a la pantalla de inicio de la aplicación (seguir).		
	6. Termina la Vista actual.	

Curso alternativo de los eventos

Acción	Curso alternativo
Prioridad	Crítica
Mejoras	-

Medias a utilizar

Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	A_Presentación	Comienza con la aparición en el centro de la pantalla del logotipo representativo del Ministerio del Poder Popular para la Energía y Petróleo de Venezuela, el cual se desvanece y da paso a la aparición en el mismo lugar del logotipo de la Misión Energética en Venezuela, que también se desvanece.	Existente
Sonido	Sonido_01	Tema Musical "La Internacional"	Existente
Texto	T_Inf	Texto que describe el objetivo de la aplicación, su misión y la importancia que tiene.	Existente

Capítulo 2: Descripción de la solución propuesta

Reglas Pedagógicas

-

Tabla 2: Descripción textual de la Vista Inicio

Descripción textual de la Vista de Presentación “Inicio”		
Actores de la Vista de Presentación	Usuario	
Propósito	Permitir al usuario visualizar la pantalla principal de la aplicación.	
Objetivos Pedagógicos	-	
Resumen	La vista se inicia cuando el usuario selecciona la opción de seguir de la vista de Presentación, se muestran las imágenes de fondo y culmina cuando se muestra la Vista Opciones Principales.	
Vistas asociadas	Vista “ <i>Opciones Principales</i> ” (<i>Subsistema Opciones Principales</i>)	
Referencias	R1	
Precondiciones	Debe haberse ejecutado la Vista Presentación.	
Poscondiciones	Vista Inicio visualizada.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. Selecciona la opción de seguir en la Vista Presentación.		
	2. Se presentan las imágenes de fondo de la pantalla.	
		2.1 La imagen 01 aparece desplazándose desde abajo hacia arriba por el borde izquierdo de la pantalla. 2.2 La imagen 02 se desplaza desde el extremo superior hasta el inferior y se posiciona a la derecha de la imagen 01. 2.3 La imagen 03 se desplaza igual que la

Capítulo 2: Descripción de la solución propuesta

		<p>imagen 01 y se coloca a la derecha de la imagen 02.</p> <p>2.4 La imagen 04 se desplaza igual que la imagen 02 y se coloca a la derecha de la imagen 03.</p> <p>2.5 La imagen 05 se desplaza desde el extremo inferior hacia el extremo superior de la pantalla y se coloca a la derecha de la imagen 04.</p> <p>2.6 La imagen 06 se desplaza igual que la imagen 05 y se coloca a su derecha.</p> <p>2.7 La imagen 07 se desplaza desde el borde superior hacia el borde inferior de la pantalla y se coloca a la derecha de la imagen 06.</p> <p>2.8 La imagen 08 se desplaza desde el borde derecho de la pantalla hacia el borde izquierdo, por el extremo superior y se coloca encima del resto de las imágenes.</p> <p>2.9 La imagen 09 viene desde el borde inferior de la pantalla y se posiciona de forma vertical a la derecha de la imagen 07.</p>
	<p>7. Se muestra la Vista “Opciones Principales” y culmina la vista actual.</p>	
<p>Curso alternativo de los eventos</p>		
<p>Acción</p>	<p>Curso alternativo</p>	
<p>Prioridad</p>	<p>Crítica</p>	
<p>Mejoras</p>	<p>-</p>	
<p>Medias a utilizar</p>		

Capítulo 2: Descripción de la solución propuesta

Tipo de media	Nombre	Descripción	Estado
Imagen	I_01	Imagen que representa el libro El Vuelo de la Mariposa Amarilla.	Existente
Imagen	I_02	Imagen que representa el libro La Gaviota Humanocrática.	Existente
Imagen	I_03	Imagen que representa el libro Rumbo al Socialismo.	Existente
Imagen	I_04	Imagen que representa el libro Un Grano de Maíz I.	Existente
Imagen	I_05	Imagen que representa el libro Un Grano de Maíz II.	Existente
Imagen	I_06	Imagen que representa el libro Un Grano de Maíz III.	Existente
Imagen	I_07	Barra de color negro	Existente
Imagen	I_08	Imagen que muestra el título de la colección de libros electrónicos, y de la aplicación.	Existente
Imagen	I_09	Barra roja con trazos	Existente
Video o Animación	-		
Sonido	Sonido_01	Tema Musical “La Internacional”	Existente
Texto	-		
Reglas Pedagógicas			
-			

Tabla 3: Descripción textual de la Vista Portada

Descripción textual de la Vista de Presentación <i>Portada</i>	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario conocer los datos principales de un libro, tales como: título, autor y tópicos que lo componen, y acceder a su contenido.
Objetivos Pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona uno de los libros incluidos como temas en la aplicación. Culmina cuando selecciona la opción de ver contenido.

Capítulo 2: Descripción de la solución propuesta

Vistas asociadas	Vista "Contenido" (Subsistema Contenido), Vista "Tópicos" (Subsistema Común), Vista "Opciones Principales" (Subsistema Opciones Principales), Vista "Común" (Subsistema Común)	
Referencias	R3	
Precondiciones		
Poscondiciones	Se muestran los datos relevantes (título, autor y una imagen de identificación) del libro seleccionado.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario selecciona un libro.		
	2. Muestra la imagen de fondo correspondiente al libro seleccionado.	
		2.1 La imagen se mostrará en la pantalla, cubriendo desde el borde derecho de la misma, hasta donde se encuentran ubicados los elementos de la Vista Menú, y acotada superior e inferiormente por las Vistas Opciones Principales y Común.
	3. Se muestra una opción gráfica interactiva para permitir al usuario la visualización del contenido del libro.	
		3.1 Aparecerá encima de la imagen de identificación del libro, debajo del título y en forma de flecha.
4. Selecciona la opción de ver contenido.		
	5. Se ejecuta la Vista Contenido y culmina la vista.	
Curso alternativo de los eventos		
Acción	Curso alternativo	

Capítulo 2: Descripción de la solución propuesta

1. a El usuario selecciona el libro 1, El Vuelo de la Mariposa Amarilla.			
	1. a.1 Muestra la imagen I_01.		
1. b El usuario selecciona el libro 2, La Gaviota Humanocrática.			
	1. b.1 Muestra la imagen I_02.		
1. c El usuario selecciona el libro 3, Rumbo al Socialismo.			
	1. c.1 Muestra la imagen I_03.		
1.d El usuario selecciona el libro 4, Un Grano de Maíz I.			
	1. d.1 Muestra la imagen I_04.		
1. e El usuario selecciona el libro 5, Un Grano de Maíz II.			
	1. e.1 Muestra la imagen I_05.		
1. f El usuario selecciona el libro 6, Un Grano de Maíz III.			
	1. f.1 Muestra la imagen I_06.		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	I_01	Imagen correspondiente al libro El Vuelo de la Mariposa Amarilla, conformada por el título del libro, el autor del mismo y una pequeña imagen de un niño con una mariposa en la esquina inferior derecha.	Existente
Imagen	I_02	Imagen correspondiente al libro La Gaviota Humanocrática, conformada por el título del libro, el autor del mismo y una pequeña imagen de una gaviota, el presidente Chávez y una estrella en la esquina inferior derecha.	Existente
Imagen	I_03	Imagen correspondiente al libro Rumbo al Socialismo, conformada por el título del libro, el autor del mismo y una	Existente

Capítulo 2: Descripción de la solución propuesta

		pequeña imagen del presidente Chávez en la esquina inferior derecha.	
Imagen	I_04	Imagen correspondiente al libro Un Grano de Maíz 1, conformada por el título del libro, el autor del mismo y una pequeña imagen del Quijote en la esquina inferior derecha.	Existente
Imagen	I_05	Imagen correspondiente al libro Un Grano de Maíz 2, conformada por el título del libro, el autor del mismo y una pequeña imagen del Quijote en la esquina inferior derecha.	Existente
Imagen	I_06	Imagen correspondiente al libro Un Grano de Maíz 3, conformada por el título del libro, el autor del mismo y una pequeña imagen del Quijote en la esquina inferior derecha.	Existente
Video o Animación	-		
Sonido	-		
Texto	-		
Reglas Pedagógicas			
-			

Tabla 4: Descripción textual de la Vista Tópicos

Descripción textual de la Vista de Presentación Tópicos	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario conocer los tópicos (capítulos o artículos) de un libro determinado.
Objetivos Pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona uno de los libros incluidos como temas en la aplicación. Culmina cuando selecciona la opción de ver el contenido de uno de los tópicos.
Vistas asociadas	Vista “ <i>Información</i> ” (Subsistema Contenido), Vista “ <i>Opciones_ Principales</i> ” (Subsistema Opciones Principales), Vista “ <i>Común</i> ” (Subsistema Común)

Capítulo 2: Descripción de la solución propuesta

Referencias	R3	
Precondiciones	Debe haberse seleccionado alguno de los libros.	
Poscondiciones	Se muestran los tópicos del libro seleccionado.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario selecciona un libro.		
	2. Muestra en forma de menú, los capítulos o artículos que conforman el libro seleccionado, que constituyen una de las formas de acceder al contenido del libro.	
		2.1 El texto 05 aparecerá en el borde izquierdo de la pantalla, en posición vertical, formando un menú interactivo. Tendrá un color de fondo y de fuente diferente para cada libro.
	3. Muestra una opción gráfica para desplazarse verticalmente en el menú de tópicos.	
4. Selecciona un tópico.		
	5. Se ejecuta la Vista "Información"	
	6. Culmina la Vista actual.	
Curso alternativo de los eventos		
Acción	Curso alternativo	
Prioridad	Crítica	
Mejoras	-	
Medias a utilizar		

Capítulo 2: Descripción de la solución propuesta

Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	-		
Sonido	-		
Texto	T_01	Texto que representa los tópicos de un libro, varía para cada libro. Hace función de menú, ya que constituye un vínculo al contenido. El color de la fuente y del fondo del mismo está en correspondencia con el color designado para identificar al libro, por lo que es único para cada uno.	Existente
Reglas Pedagógicas			
-			

Tabla 5: Descripción textual de la Vista Información

Descripción textual de la Vista de Presentación <i>Información</i>	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario conocer la información asociada al contenido de cada libro.
Objetivos Pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona la opción de ver el contenido.
Vistas asociadas	Vista " <i>Portada</i> " (<i>Subsistema Contenido</i>), Vista " <i>Tópicos</i> " (<i>Subsistema Contenido</i>), Vista " <i>Palabras_Calientes</i> " (<i>Subsistema Contenido</i>), Vista " <i>Opciones_Principales</i> " (<i>Subsistema Opciones Principales</i>), Vista " <i>Común</i> " (<i>Subsistema Común</i>)
Referencias	R3, R5
Precondiciones	Debe seleccionarse un tópico para ver su contenido.
Poscondiciones	Se muestra el contenido referente al tópico seleccionado en el libro.
Curso normal de eventos	

Capítulo 2: Descripción de la solución propuesta

Acciones del actor	Respuesta del sistema	Elementos de la vista
1. El usuario selecciona la opción de ver contenidos.		
	2. Muestra el texto correspondiente al contenido del libro seleccionado.	
		2.1 El Texto 02 aparecerá en el área derecha de la pantalla. Como encabezado tendrá el título del tópico al que corresponde, que estará separado del resto del texto por una línea del color identificador del libro. Contiene palabras y frases significativas para el contenido del libro (palabras calientes), que aparecerán en color diferente y permitirán ser seleccionadas para conocer información adicional sobre ellas.
	3. Muestra el texto correspondiente al número de la página donde se encuentra ubicado el usuario, y el total de páginas del libro.	
		3.1 El texto 03 se mostrará en la parte inferior del área de información, centrado y entre las opciones de navegación interna del libro, o sea, en el centro entre las opciones de ir hacia la página siguiente y anterior.
	4. Muestra las opciones gráficas de ir hacia la siguiente página de contenidos, a la anterior, al siguiente tópico y al anterior.	

Capítulo 2: Descripción de la solución propuesta

		4.1 Las opciones aparecerán centradas en la parte inferior del área de texto. Las opciones de moverse por las páginas de un tópico (siguiente y anterior) se muestran en forma de flecha; y las opciones de moverse por lo tópicos del libro aparecerán representadas por dos flechas continuas y a los extremos de las opciones de navegación interna entre páginas.
5. Selecciona la opción de ir hacia la siguiente página de contenido.		
	6. Muestra el contenido de la siguiente página del tópico actual.	
	7. Termina la Vista.	

Curso alternativo de los eventos

Acción	Curso alternativo
2. a Selecciona la opción de ver una palabra caliente.	
	2. a.1 Se ejecuta la Vista Palabras Calientes y culmina la vista.
4. a El usuario selecciona opción de ir hacia la página anterior del tópico.	
	4. a.1 Muestra el contenido de la página anterior del tópico actual.
4. b Selecciona la opción de ir hacia el siguiente tópico del libro.	
	4. b.1 Muestra el contenido de la primera página del tópico siguiente del libro.
4. c Selecciona la opción de ir hacia el tópico anterior del libro.	
	4. c.1 Muestra el contenido de la primera página del tópico anterior del libro.
Prioridad	Crítica

Capítulo 2: Descripción de la solución propuesta

Mejoras		-	
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	-		
Sonido	-		
Texto	T_02	Texto correspondiente al contenido presente en cada tópico del libro. Si el libro está formado por artículos, como el caso de Un Grano de Maíz I, II y III, entonces, el texto además incluirá, la fecha del artículo, que se ubicará entre la línea divisoria y el contenido del artículo.	Existente
	T_03	Texto que representa el número de página actual y el total que tiene el libro. El formato definido es: número de la página de total de página.	Existente
Reglas Pedagógicas			
-			

Tabla 6: Descripción textual de la Vista Palabras_Calientes

Descripción textual de la Vista de Presentación <i>Palabras_Calientes</i>	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario conocer información adicional sobre palabras y frases específicas, y de importancia relevante dentro del libro.
Objetivos Pedagógicos	-
Resumen	La vista se inicia cuando el usuario selecciona una palabra caliente. Se muestra la información correspondiente y culmina cuando el usuario selecciona la opción de salir.
Vistas asociadas	Vista “ <i>Información</i> ” (<i>Subsistema Contenido</i>)

Capítulo 2: Descripción de la solución propuesta

Referencias	R4		
Precondiciones	Debe haberse ejecutado la Vista de Información.		
Poscondiciones	Se muestra información relacionada con la palabra o frase seleccionada.		
Curso normal de eventos			
Acciones del actor	Respuesta del sistema	Elementos de la vista	
1. Selecciona una palabra caliente.			
	2. Muestra información relacionada con la palabra seleccionada.		
		2.1 El texto 04 aparecerá cubriendo toda la pantalla.	
	3. Muestra la opción de salir de la vista.		
		3.1 La opción salir aparecerá como un botón con una X, que indica cerrar, y estará ubicado en el borde superior derecho de la pantalla.	
Curso alternativo de los eventos			
Acción	Curso alternativo		
2. a Selecciona la opción de salir de la vista.			
	2. a.1 Culmina la vista.		
	2. a.2 Regresa a la vista anterior.		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	-		

Capítulo 2: Descripción de la solución propuesta

Sonido	-		
Texto	T_04	Texto que no aparece en el contexto del libro y que brinda información adicional sobre una palabra o frase determinada y de importancia relevante. Este texto está formado por la palabra o frase y por la información asociada a ella.	Existente
Reglas Pedagógicas			
-			

Tabla 7: Descripción textual de la Vista Resultados_Búsqueda

Descripción textual de la Vista de Presentación Resultados_ Búsqueda		
Actores de la Vista de Presentación	Usuario	
Propósito	Permitir al usuario conocer las coincidencias encontradas en el libro en el que se encuentra, con el parámetro de búsqueda especificado y acceder a la página de contenido donde se ubica la palabra o frase buscada.	
Objetivos Pedagógicos	-	
Resumen	La vista se inicia cuando el usuario presiona la opción de buscar, el sistema muestra las coincidencias halladas con el parámetro especificado y termina cuando el usuario selecciona la opción de salir.	
Vistas asociadas	Vista "Información" (<i>Subsistema Contenido</i>), Vista "Común" (<i>Subsistema Común</i>)	
Referencias	R6	
Precondiciones	Debe haber introducido algún parámetro de búsqueda.	
Poscondiciones	Se muestran los resultados encontrados a través de la búsqueda.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. Selecciona la opción de buscar.		

Capítulo 2: Descripción de la solución propuesta

	<p>2. Busca los elementos, entiéndase por elementos a las palabras o frases, que coincidan con el parámetro de búsqueda especificado por el usuario y los muestra en una lista.</p> <p>Permite seleccionar entre los resultados encontrados para ver el contenido asociado a este.</p> <p>Permite seleccionar la opción de salir de la vista.</p>	
		<p>2.1 El texto 05, correspondiente a las coincidencias halladas, se mostrará en toda la pantalla, subrayando la frase o palabra introducida como parámetro para la búsqueda, la cual hará función de vínculo hacia la página donde se encuentra el contenido relacionado con ella.</p> <p>2.2 La opción salir aparecerá como un botón con una X, que indica cerrar, y estará ubicado en el borde derecho superior de la pantalla.</p>
3. Selecciona un resultado de la lista.		
	4. Muestra el contenido relacionado con el resultado seleccionado. (Ver <i>Vista Información</i>).	
Curso alternativo de los eventos		
Acción	Curso alternativo	
2. a Selecciona la opción de salir de la vista.		
	2. a.1 Culmina la vista.	
2. b No existe ningún elemento que coincida con el parámetro de búsqueda		

Capítulo 2: Descripción de la solución propuesta

especificado por el usuario.			
		2. b.1 Muestra un mensaje indicando que la búsqueda no obtuvo ningún resultado favorable.	
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	-		
Sonido	-		
Texto	T_05	Texto correspondiente a los resultados de la búsqueda. Tiene el siguiente formato: primero aparece el tópico donde se encontró, luego el número de la página y por último una pequeña frase donde se encuentra insertada la palabra o frase buscada.	Existente
Reglas Pedagógicas			
-			

Tabla 8: Descripción textual de la Vista Común

Descripción textual de la Vista de Presentación <i>Común</i>	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario acceder a las opciones comunes de la aplicación (imprimir, buscar y música)
Objetivos Pedagógicos	-
Resumen	Muestra las opciones comunes y culmina cuando se selecciona una de ellas.
Vistas asociadas	Vista " <i>Resultados_Búsqueda</i> " (<i>Subsistema Común</i>)
Referencias	R6, R7, R8
Precondiciones	

Capítulo 2: Descripción de la solución propuesta

Poscondiciones		
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
	1. Muestra la opción de Imprimir y permite al usuario que la seleccione.	
	2. Muestra la opción de Música, permitiendo que ésta sea activada o desactivada.	
	3. Muestra el texto “Buscar”, brinda la opción al usuario de introducir el parámetro de búsqueda y permite iniciar la búsqueda.	
4. Selecciona la opción <i>Imprimir</i> .		
	5. Se muestran las opciones de impresión que brinda la aplicación: Imprimir la página actual Imprimir el tópico actual Imprimir el libro Permitiendo que se seleccione una de ellas.	
		5.1 Las opciones de impresión se muestran en forma de menú al colocarse encima del botón y cuando se quita el mouse sobre él, desaparece.
6. Selecciona la opción de imprimir la página actual.		
	7. Imprime el texto contenido en la página actual visualizada por el usuario.	
	8. Culmina la vista.	
Curso alterno de los eventos		
Acción	Curso alterno	
4.a Selecciona la opción <i>Buscar</i>		

Capítulo 2: Descripción de la solución propuesta

	4. a.1 Se ejecuta la Vista Resultados_Búsqueda y culmina la vista actual.		
4.b Selecciona la opción <i>Música</i>			
	4. b.1 Muestra al usuario dos opciones para tratar la música de fondo, permitiendo que seleccione una de ellas.		
4. b.2 Selecciona una de las opciones brindadas.			
	4. b.3 Si selecciona la opción “Sin Música”, se desactiva la música de fondo de la aplicación.		
	4. b.4 Si selecciona la opción “Con Música”, se activa la música de fondo de la aplicación.		
6. a Selecciona la opción de imprimir el tópico actual			
	6. a.1 Imprime el texto correspondiente al tópico actual del libro.		
6. b Selecciona la opción de imprimir el libro actual.			
	6. b.1 Imprime el texto correspondiente al libro.		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	-		
Sonido	Sonido_01	Tema Musical “La Internacional”	Existente
Texto	-		
Reglas Pedagógicas			
-			

Tabla 9: Descripción textual de la Vista Opciones_Principales

Descripción textual de la Vista de Presentación <i>Opciones_ Principales</i>	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario conocer y acceder a las opciones principales de la aplicación.
Objetivos Pedagógicos	-

Capítulo 2: Descripción de la solución propuesta

Resumen	La vista se inicia inmediatamente después de que se ha terminado de ejecutar la Vista Inicio, se muestran las opciones principales que brinda la aplicación, y culmina cuando se accede a alguna de ellas.	
Vistas asociadas	Vista "Inicio" (<i>Subsistema Aplicación</i>), Vista "Salir" (<i>Subsistema Opciones Principales</i>), Vista "Información" (<i>Subsistema Contenido</i>), Vista "Común" (<i>Subsistema Común</i>), Vista "Tópicos" (<i>Subsistema Contenido</i>)	
Referencias	R2, R9	
Precondiciones	Debe haberse ejecutado la Vista Inicio.	
Poscondiciones	Son visualizadas las opciones principales brindadas por la aplicación.	
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
	1. Muestra las opciones principales de trabajo de la aplicación.	
		1.1 Cada una de las opciones estará representada con un color diferente, que identificará a la misma en la aplicación.
2. Selecciona una de las opciones de trabajo en la aplicación.		
	3. Si selecciona alguna de las opciones entre la 1 y 6, o sea, El Vuelo de la Mariposa Amarilla, La Gaviota Humanocrática, Rumbo al Socialismo, Un Grano de Maíz I, Un Grano de Maíz II, o Un Grano de Maíz III, se ejecutan las Vistas Portada y Tópico y culmina la vista actual.	
	4. Si selecciona la opción Salir se ejecuta la Vista Salir y	

Capítulo 2: Descripción de la solución propuesta

	culmina la vista actual.		
	5. Si selecciona la opción de cambiar el tamaño de visualización de la aplicación en la pantalla, se ajusta el tamaño de la misma en dependencia del estado en el que se encuentra, si está en su máxima capacidad, se minimiza y si está minimizado, se restaura hasta ocupar toda la pantalla.		
Curso alternativo de los eventos			
Acción	Curso alternativo		
Prioridad	Crítica		
Mejoras	-		
Medias a utilizar			
Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	-		
Sonido	-		
Texto	-		
Reglas Pedagógicas			
-			

Tabla 10: Descripción textual de la Vista Salir

Descripción textual de la Vista de Presentación <i>Salir</i>	
Actores de la Vista de Presentación	Usuario
Propósito	Permitir al usuario confirmar si realmente desea salir de la aplicación.
Objetivos Pedagógicos	-
Resumen	La vista inicia cuando el usuario selecciona la opción de salir de la aplicación, el sistema

Capítulo 2: Descripción de la solución propuesta

	muestra las opciones de confirmación y culmina cuando el usuario selecciona una de estas opciones brindadas.	
Vistas asociadas	Vista "Opciones Principales" (Subsistema Opciones_Principales), Vista "Créditos" (Subsistema Opciones_Principales)	
Referencias	R9	
Precondiciones		
Poscondiciones		
Curso normal de eventos		
Acciones del actor	Respuesta del sistema	Elementos de la vista
1. Selecciona la opción de salir.		
	2. Muestra una pregunta para confirmar la opción de salir, y dos opciones de confirmación de salida, una opción de aceptar y otra de cancelar.	
3. Selecciona una de las opciones brindadas.		
	4. Si selecciona la opción positiva se muestra la animación correspondiente a los créditos, y una vez finalizada la misma, se cierra la aplicación, culminando así la vista actual.	
	5. Si selecciona la opción negativa se cierra la vista y regresa a la vista anterior.	
Curso alternativo de los eventos		
Acción	Curso alternativo	
Prioridad	Crítica	
Mejoras	-	
Medias a utilizar		

Capítulo 2: Descripción de la solución propuesta

Tipo de media	Nombre	Descripción	Estado
Imagen	-		
Video o Animación	A_Créditos	Animación que muestra el nombre completo de nuestra universidad y el nombre de la Institución para la que fue creado el producto.	Existente
Sonido			
Texto	-		
Reglas Pedagógicas			
-			

Conclusiones del capítulo

Se definieron los conceptos fundamentales asociados al negocio que se estudió, los cuales fueron utilizados para el modelamiento del mismo a través del Modelo de Dominio, y se especificaron los requisitos funcionales y no funcionales que debe cumplir el sistema a desarrollar. Por otra parte se definieron los subsistemas en los que funcionalmente estará dividida la aplicación, las vistas de presentación que conformarán las pantallas y sus descripciones textuales, y se especificó la navegabilidad entre ellas a través de los diagramas de estructura de navegación.

Capítulo 3

Construcción de la solución propuesta

3.1 Introducción

Una vez finalizada la primera iteración del flujo de trabajo de Requerimientos que establece RUP para el desarrollo de software, se tienen los elementos más importantes para realizar el diseño y la implementación del mismo, y es precisamente sobre estos flujos de trabajo que trata este capítulo. Se representarán los diagramas de estructura de presentación, los diagramas de clases del diseño, de interacción y el modelo de implementación, que constituyen los artefactos fundamentales de estos flujos de trabajo y que permitirán construir la solución propuesta.

3.2 Diagramas de Estructura de Presentación (DEP)

Estos diagramas aparecen formando parte de la Vista de Presentación definida en ApEM-L.

Los DEP abordan un aspecto esencial en las aplicaciones con tecnología multimedia: la definición de la estructura que tendrán las futuras interfaces de comunicación con el usuario en cuanto a sus componentes.

Permiten representar la estructura que tendrán dichas interfaces, estableciendo una organización lógica de los elementos conformantes de las mismas y dejándoles a los diseñadores gráficos la función de decidir dónde y cómo serán en términos visuales dichos elementos. [14]

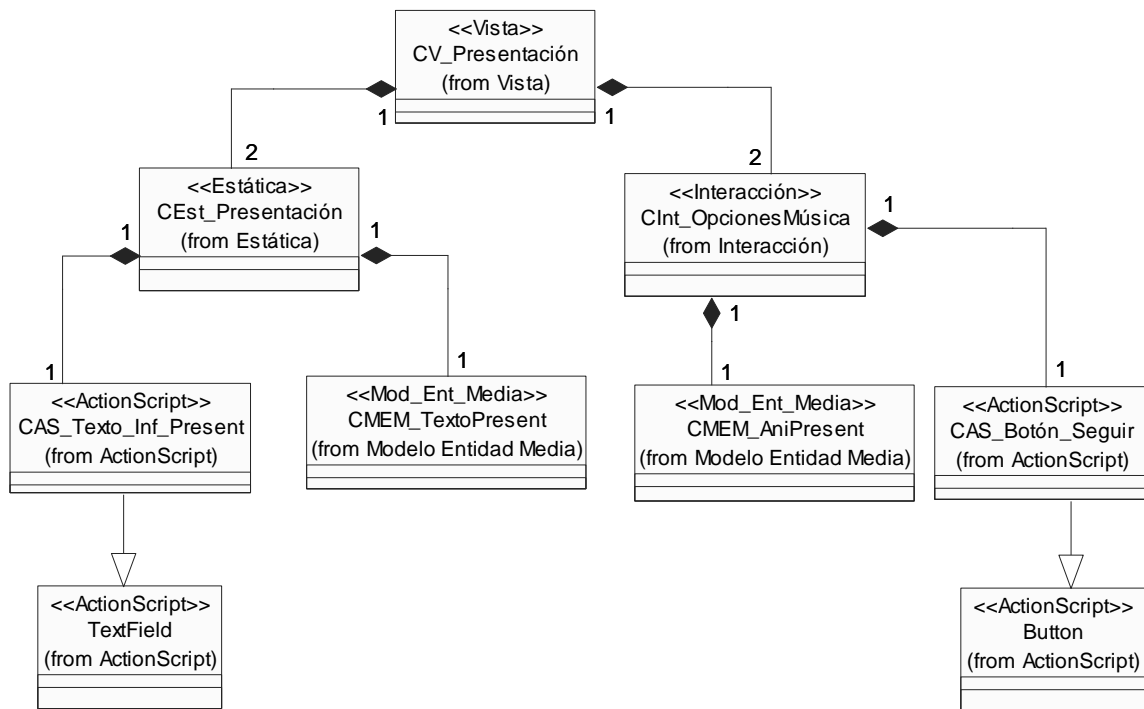


Ilustración 11: DEP Vista Presentación

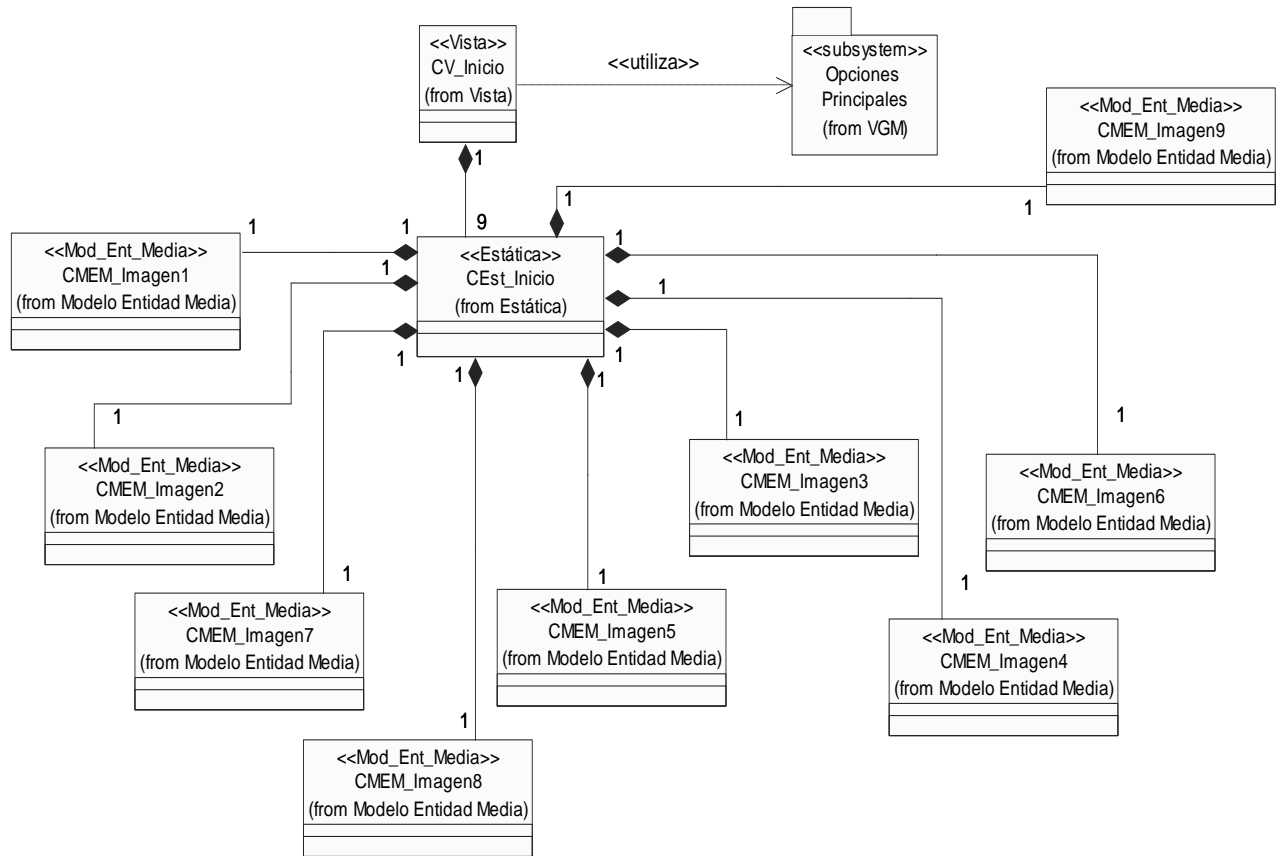


Ilustración 12: DEP Vista Inicio

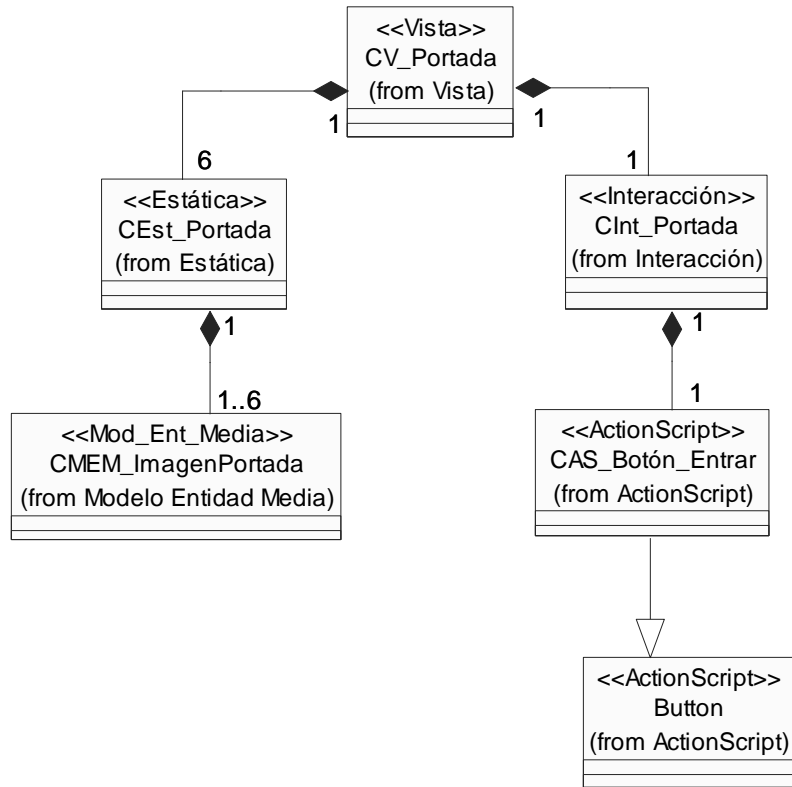


Ilustración 13: DEP Vista Portada

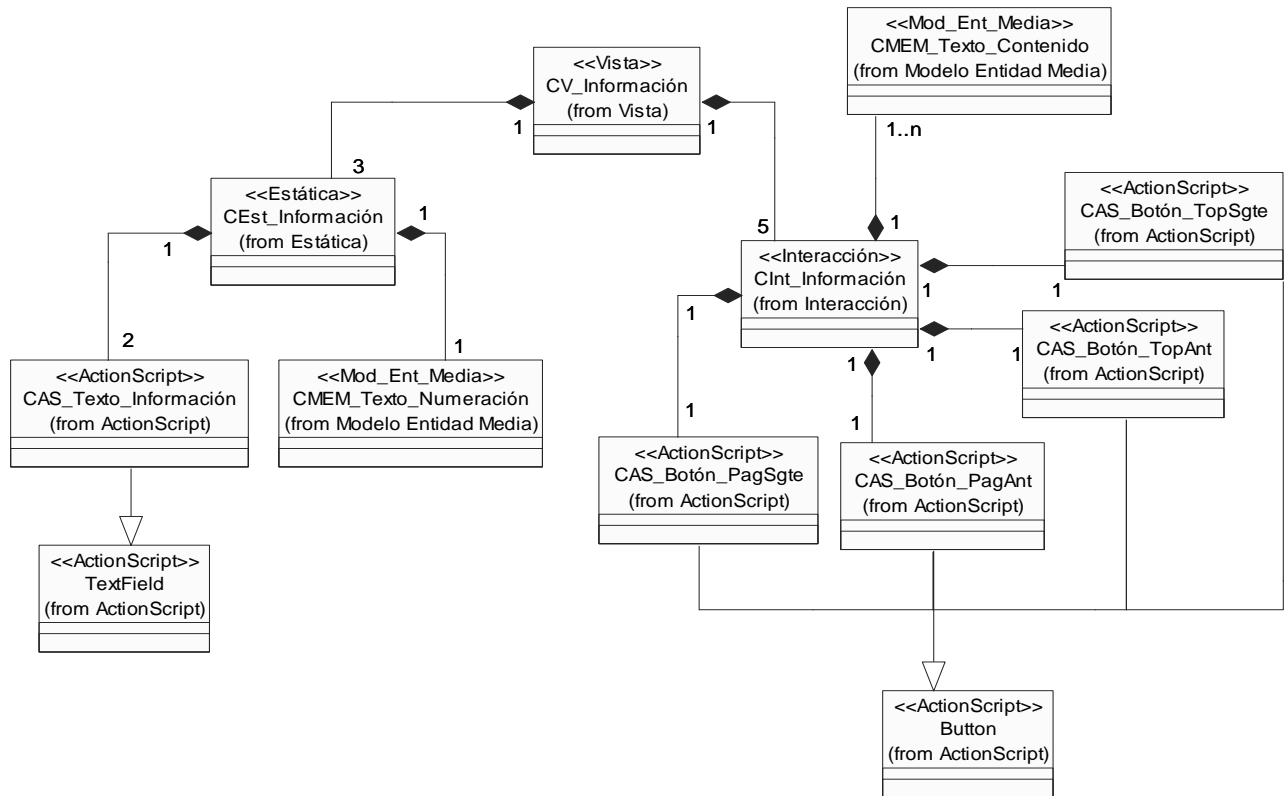


Ilustración 14: DEP Vista Información

- ✓ DEP Vista Común [Ver Anexo 1]
- ✓ DEP Vista Opciones_Principales [Ver Anexo 2]

3.3 Modelo de Diseño

Puede definirse diseño como el proceso de aplicar distintas técnicas y principios con el propósito de definir un producto con los suficientes detalles como para permitir su realización física.

El diseño del software es un proceso iterativo mediante el cual los requisitos se traducen en un “plano” para construir el software. Es una secuencia de pasos que hacen posible que el diseñador describa todos los aspectos del software que se va construir. [26]

En la etapa de diseño se modela el sistema y se encuentra su forma para que soporte todos los requisitos, incluyendo los requisitos no funcionales y otras restricciones que se establezcan. El modelo

Capítulo 3: Construcción de la solución propuesta

de diseño es uno de los artefactos resultantes de esta etapa de desarrollo y Pressman lo define en su libro *Ingeniería de Software. Un enfoque práctico (Quinta Edición)*, como el equivalente a los planos de un arquitecto para una casa. Es un modelo de objetos que describe las relaciones entre las clases definidas a través de los diagramas de clases y diagramas de interacción.

3.3.1 Diagramas de clases

El diagrama de clases constituye uno de los diagramas de estructura estática que define UML para el modelado del software, es la representación de las clases identificadas y sus relaciones. En la etapa de diseño las clases que componen estos diagramas son descritas con un mayor nivel de detalle, concebidas para satisfacer los requisitos funcionales y no funcionales, teniendo en consideración la tecnología en la cual se implementará dicho diseño.

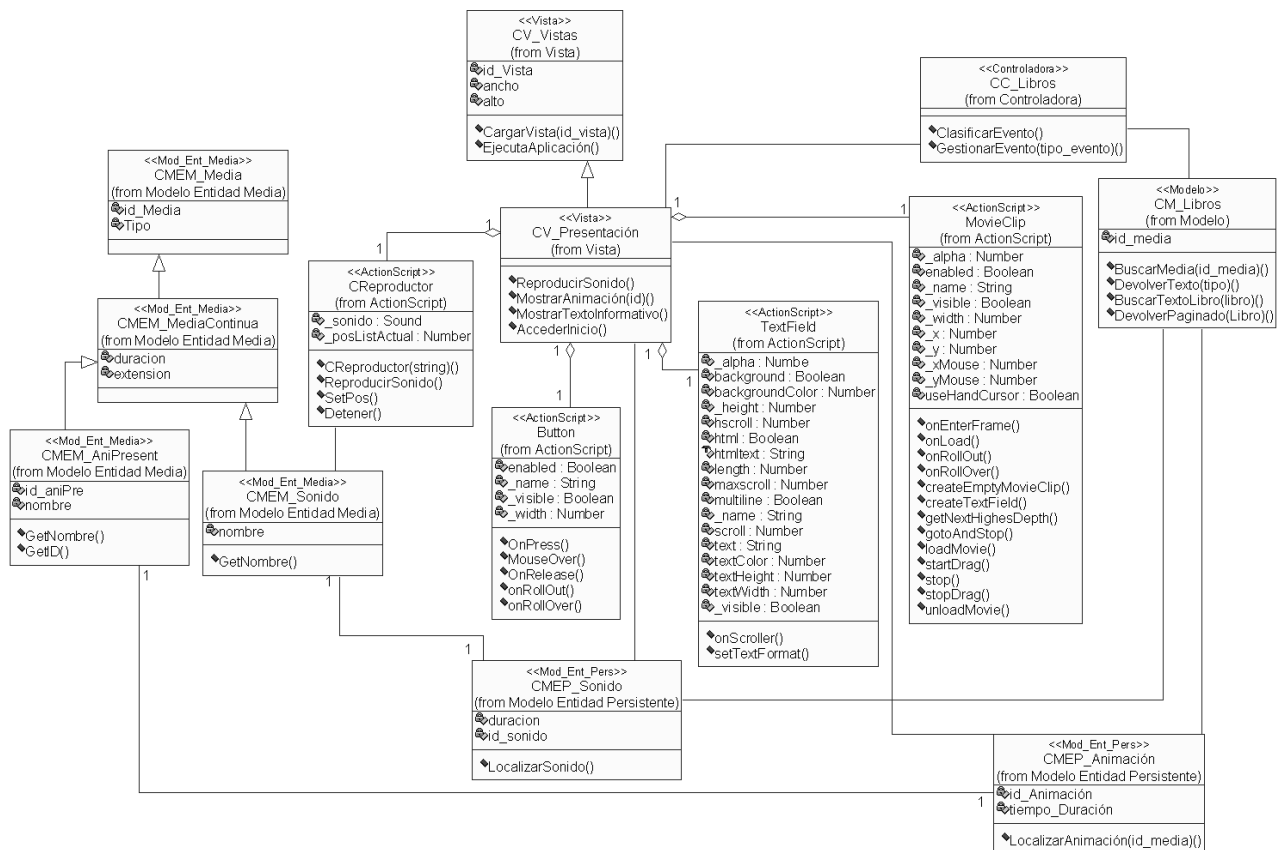


Ilustración 15: Diagrama de Clases de la Vista Presentación

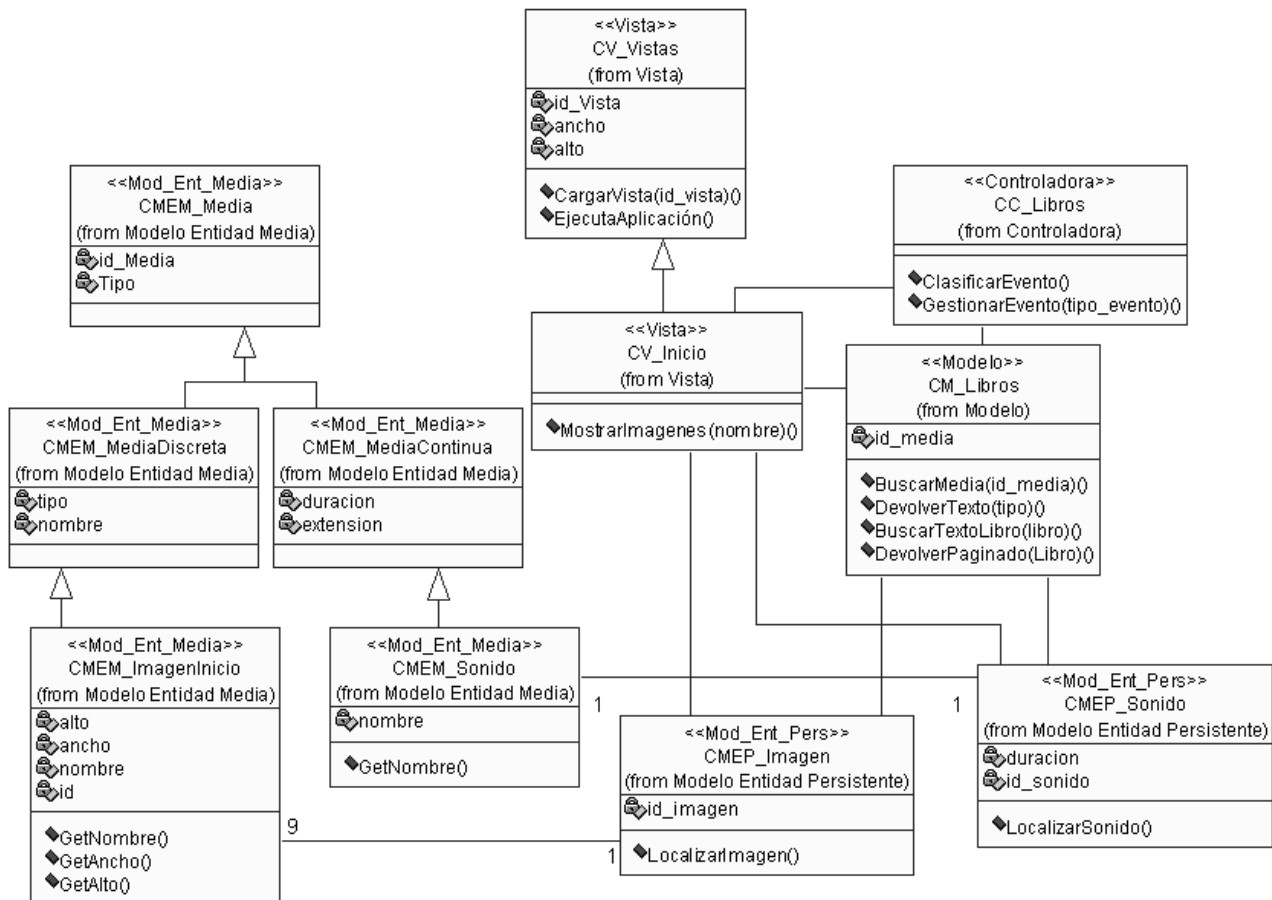


Ilustración 16: Diagrama de Clases de la Vista Inicio

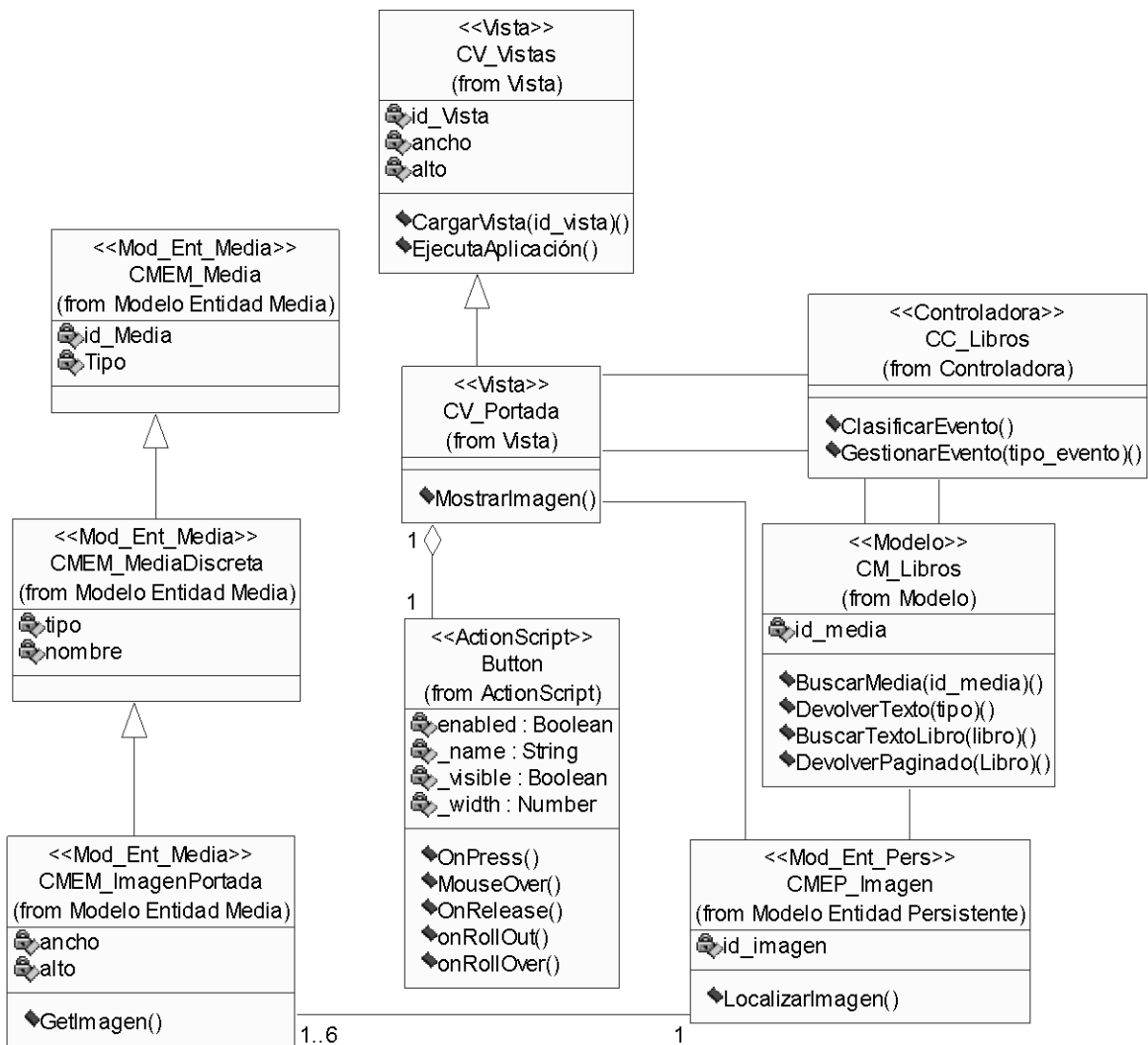


Ilustración 17: Diagrama de Clases de la Vista Portada

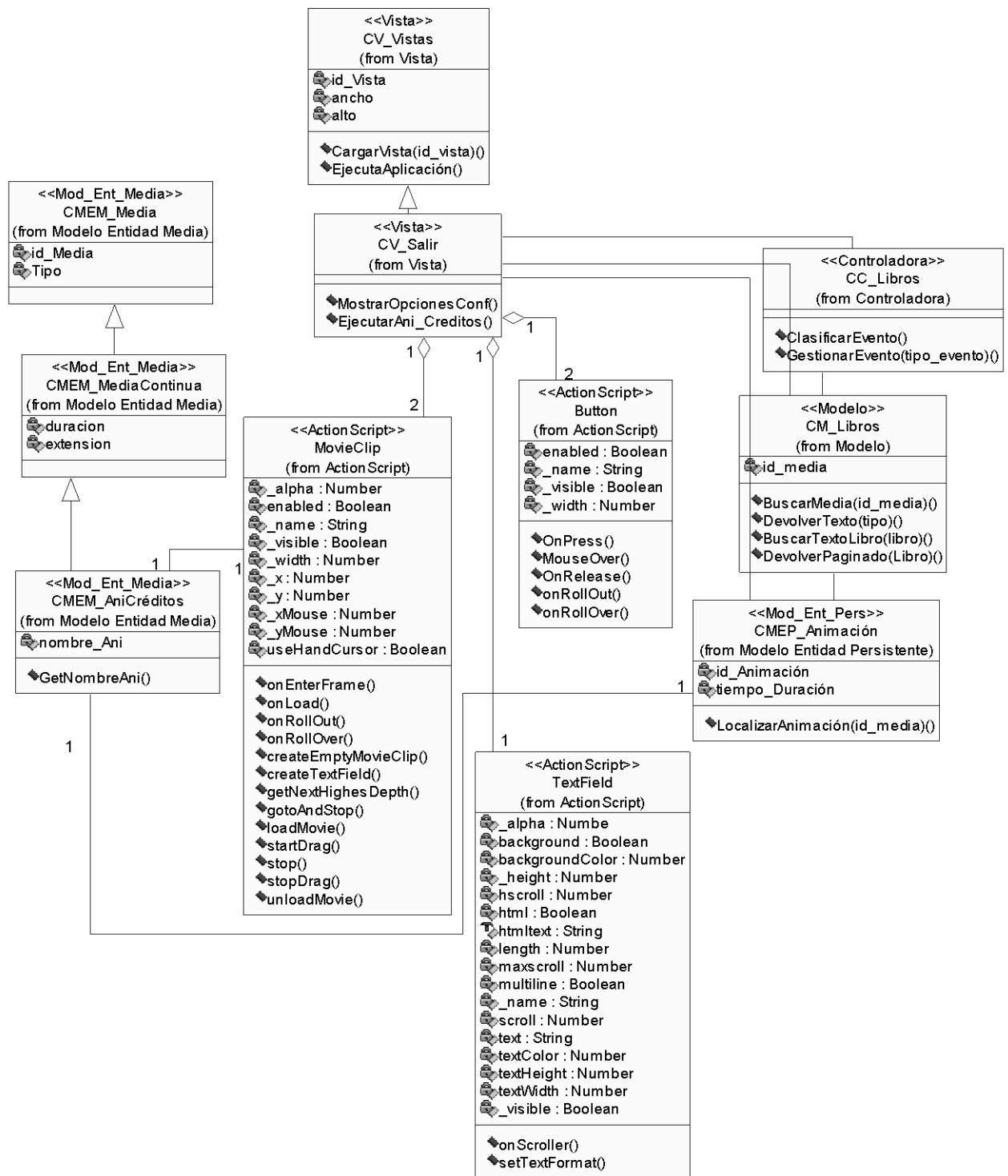


Ilustración 18: Diagrama de Clases de la Vista Salir

3.3.2 Diagramas de Interacción

Los diagramas de interacción del UML se utilizan para modelar los aspectos dinámicos de un sistema. Explican gráficamente, a través de mensajes, las interacciones existentes entre las instancias de las clases. El UML define dos tipos de estos diagramas, los diagramas de colaboración y los diagramas de secuencia.

Para este caso de estudio en particular se representarán únicamente los diagramas de secuencia, que muestran los mensajes entre los objetos a lo largo del tiempo. (Ver Anexos 3-5)

3.4 Modelo de Implementación

El modelo de implementación puede considerarse como el artefacto más significativo del Flujo de trabajo de Implementación, precisamente por la importancia que tiene para los desarrolladores entender bien el funcionamiento del sistema desde el punto de vista de componentes y sus relaciones, antes de escribir líneas de código. Este modelo está conformado por los Diagramas de Componentes y de Despliegue, los cuales describen los componentes a construir, y su organización y dependencia entre nodos físicos en los que funcionará la aplicación.

Su objetivo fundamental es describir cómo las clases del modelo de diseño se implementan en términos de componentes, tales como ejecutables, ficheros de código fuentes o tablas de una base de datos.

3.4.1 Diagrama de Componentes

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes software, ya sean componentes de código fuente binarios o ejecutables. Desde el punto de vista del diagrama de componentes se tienen en consideración los requisitos relacionados con la facilidad de desarrollo, la gestión del software, la reutilización, y las restricciones impuestas por los lenguajes de programación y las herramientas utilizadas en el desarrollo.

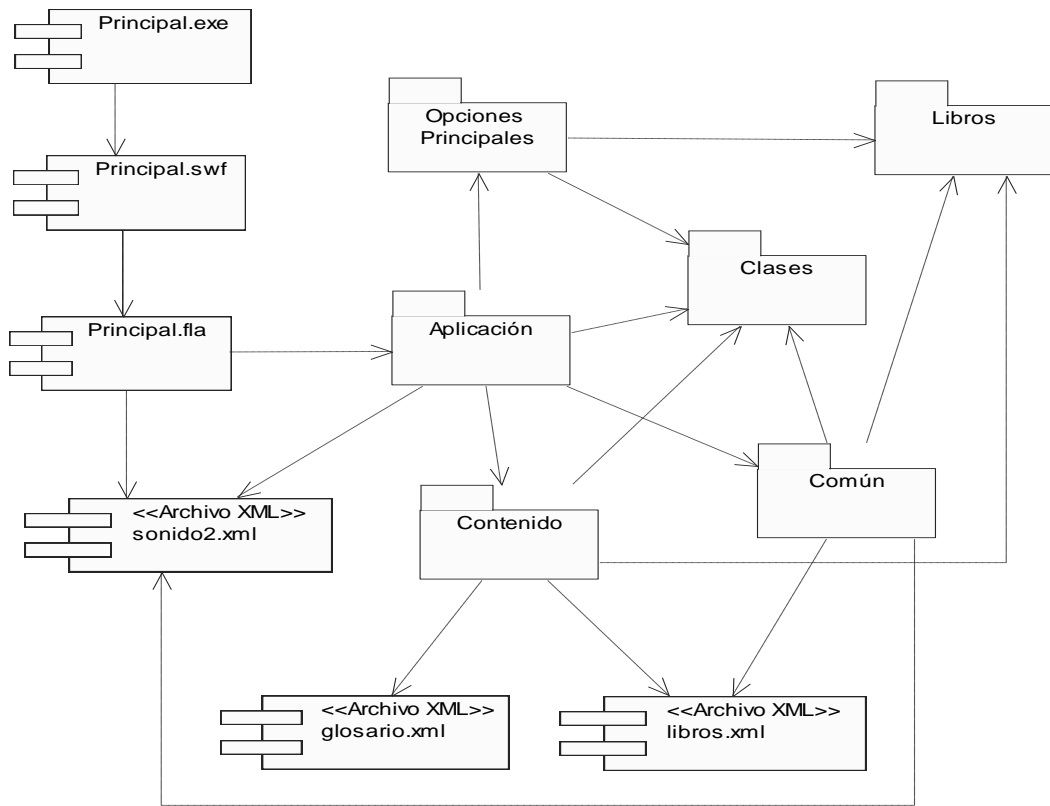


Ilustración 19: Diagrama General de Componentes

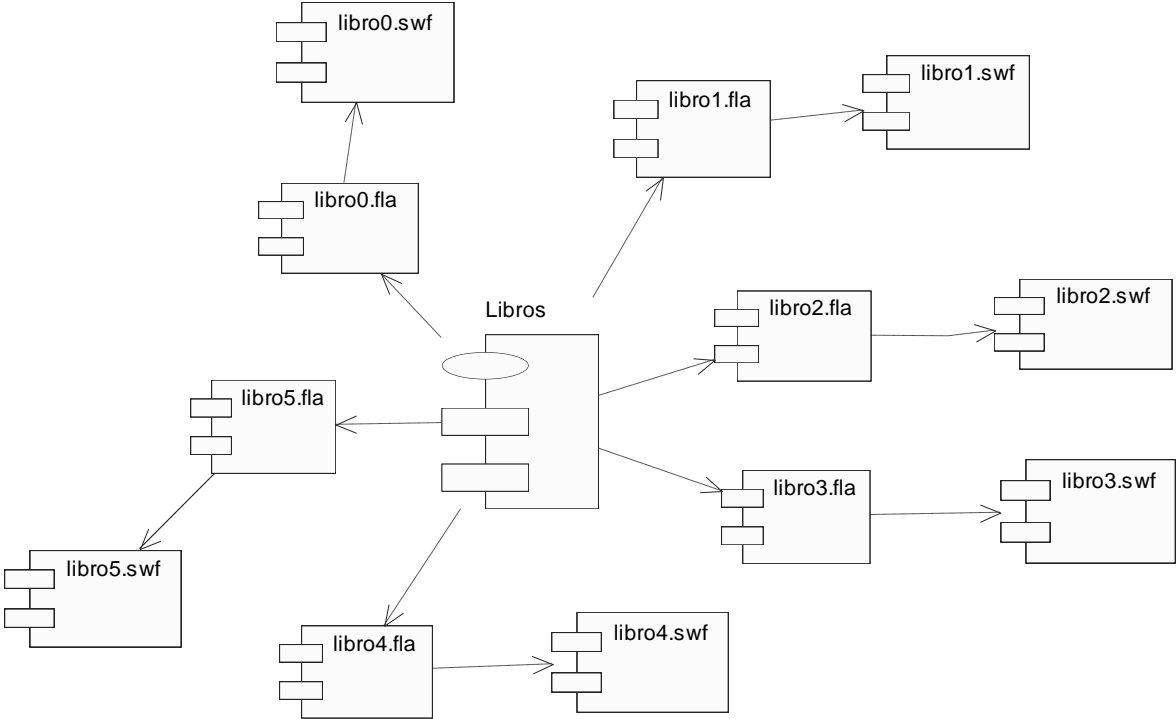


Ilustración 20: Diagrama de Componentes del Paquete Libros

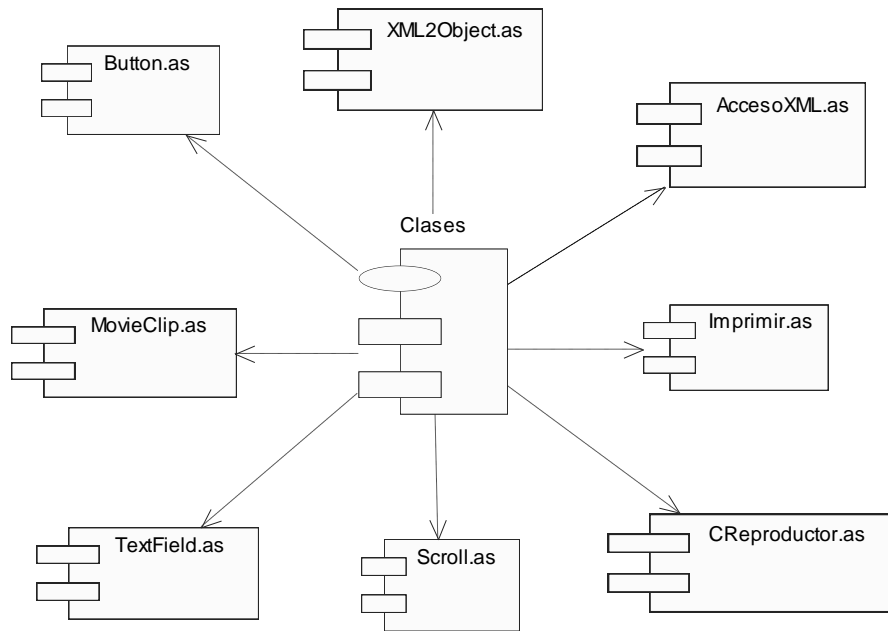


Ilustración 21: Diagrama de Componentes del Paquete Clases

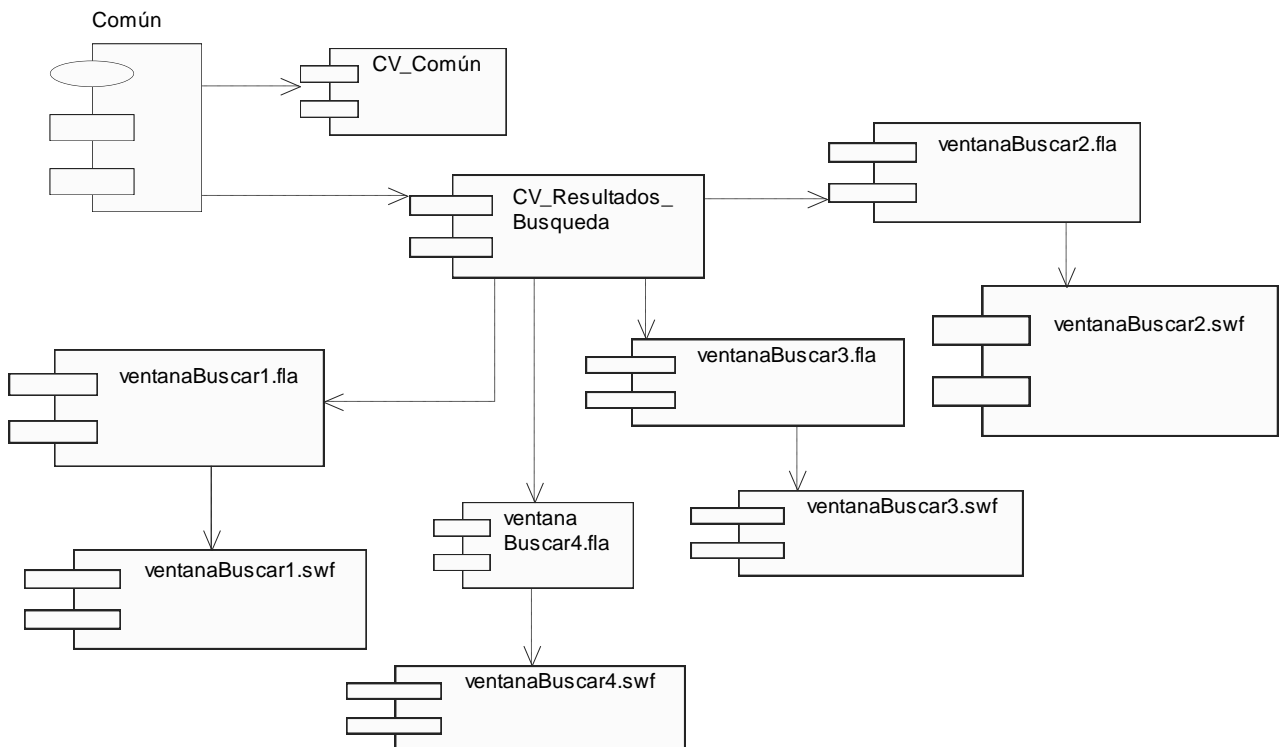


Ilustración 22: Diagrama de Componentes del Paquete Común

- ✓ Diagrama de componentes Paquete Aplicación [Ver Anexo 6]
- ✓ Diagrama de componentes Paquete Contenido [Ver Anexo 7]

3.4.2 Diagrama de Despliegue

El modelo de despliegue es un modelo de objetos que describe la distribución física del sistema, es decir, cómo se distribuyen las funcionalidades entre los nodos de cómputo. Se utiliza como entrada fundamental en las actividades de diseño e implementación debido a que la distribución del sistema tiene una influencia principal en su diseño.

El Modelo de Despliegue en sí mismo representa una correspondencia entre la arquitectura software y la arquitectura del sistema (el hardware). [7]

En el presente caso de estudio, el diagrama está compuesto únicamente por dos nodos, uno representando la computadora donde se ejecutará la aplicación, la cual solo requiere de un lector de CD y cumplir con los requisitos mínimos expuestos como especificaciones no funcionales, y otro representando un dispositivo externo, en este caso la impresora, que permitirá dar cumplimiento al deseo del cliente de poder imprimir el contenido abordado en la aplicación.

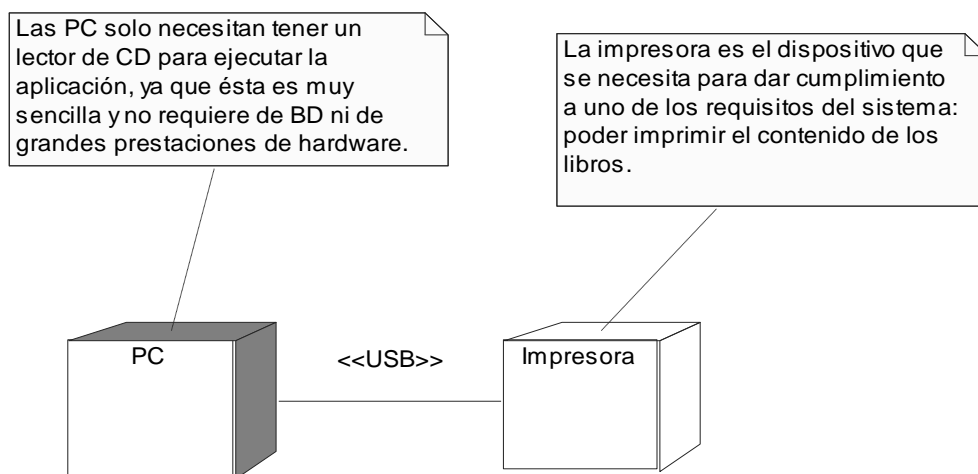


Ilustración 23: Diagrama de Despliegue

3.5 Descripción de archivos XML

Teniendo en cuenta las ventajas que brinda el lenguaje XML y las facilidades que ofrece para el trabajo con Flash, se decidió almacenar toda la información ofrecida en la aplicación en archivos de este tipo.

Se utilizaron únicamente tres archivos XML, uno para tratar el contenido de los libros, o sea, para mostrar la información recogida en los 6 libros que conforman la colección, otro para almacenar la información del Glosario de términos de los mismos y un tercero para los datos del sonido.

3.5.1 Archivo XML Libros

<libros>

<libro id= Identificador único asociado a cada libro autor=Autor del libro titulo= Título del libro cantPaginas= Cantidad total de páginas que conforman el libro tipo= Parámetro para definir si el libro está dividido por capítulos o por artículos>

<contenido>

<item id=Identificador de los elementos del contenido, en este caso, los capítulos o artículos que conformen el libro>

<titulo>Título del tópico (Artículo o capítulo)</titulo>

<autor>Autor del tópico/autor>

<paginas>

<pagina id= Identificador asociado a cada página del elemento, comienza siempre desde cero para cada nuevo elemento que conforme el libro numPagina= El número de página asociado a la página actual, es consecutivo desde el inicio, por lo que es único para cada libro >

<texto>Texto perteneciente a la página actual del contenido del libro</texto>

</pagina>

</paginas>

</item>

</contenido>

</libro>

</libros>

3.5.2 Archivo XML Glosario

<terminos>

<termino id=Identificador de cada término que conforman el glosario, es único>

<palabra>Título del término, puede ser una palabra o una frase </palabra>

<descripcion>Descripción asociada al término, es información adicional de la que aparece en el contexto del libro</descripcion>

</termino>

</terminos>

3.5.3 Archivo XML Sonido2

<lista>

<cancion id= identificador de la canción> URL donde se encuentra la música a reproducir</cancion>

</lista>

3.6 Conclusiones del capítulo

Se obtuvieron los artefactos fundamentales correspondientes a los flujos de trabajo de diseño e implementación: Modelo de Diseño y Modelo de Implementación, así como los Diagramas de Estructura de Presentación definidos por el lenguaje de modelado para aplicaciones con tecnología multimedia y la descripción de los archivos XML que se utilizan para el tratamiento de datos en la aplicación.

Capítulo 4

Estudio de factibilidad

4.1 Introducción

Frecuentemente la estimación de costo de un software se realiza muy tempranamente, sin tener una visión global del proyecto en cuestión, trayendo consigo estimaciones de costo no siempre exactas. Un estudio gubernamental americano en los proyectos de desarrollo del software reveló su magnitud: el 60 por ciento de los proyectos estaban atrasados (o sea, fuera de la fecha planificada), el 50 por ciento de los proyectos estaban por encima del costo estimado, y el 45 por ciento de los proyectos entregados eran inutilizables. De la experiencia viene la necesidad real de estimar correctamente los esfuerzos de un proyecto.

Esta estimación no solo conlleva los costos sino que tiene además la actividad de estimación de los resultados del proyecto y los valores de tiempo y recursos requeridos. En su confección se incluyen además selección de métodos, procesos, herramientas, normas, organización, etc.

El primer objetivo de la estimación es, sin dudas, la determinación de la posibilidad de ejecutar el proyecto o lo que es lo mismo el estudio de la factibilidad de acuerdo a las diferentes restricciones. Estas restricciones están condicionadas por las características propias del proyecto y/o grupo de trabajo, como son:

- ✓ Organizativas: por la estructura, los procesos y las personas.
- ✓ Económicas: se analiza la relación costos-beneficios.
- ✓ Técnicas: muy propias del grupo de trabajo pues tienen en cuenta las habilidades, experiencia y los recursos disponibles.
- ✓ Tiempo: el tiempo es un elemento fundamental cuando existen requerimientos de fecha de cumplimiento.

Para definir la factibilidad de un proyecto existen varios métodos de estimación, siendo los más conocidos, Análisis por puntos de casos de uso y Análisis de puntos de función y COCOMO II (Constructive Cost Model).

El presente capítulo abordará el estudio de factibilidad del proyecto en curso utilizando el método Análisis de puntos de función y COCOMO II.

4.2 Análisis de puntos de función y COCOMO II

La especificación de los requerimientos mediante Casos de Uso ha probado ser uno de los métodos más efectivos para capturar la funcionalidad de un sistema. Este hecho se puede apreciar en algunas metodologías actuales ampliamente difundidas, como el Proceso Unificado de Rational (RUP), en la cual se propone especificar la funcionalidad de los sistemas mediante la utilización de Casos de Uso.

Si bien los Casos de Uso permiten especificar la funcionalidad de un sistema bajo análisis, no permiten por sí mismos efectuar una estimación del tamaño que tendrá el sistema o del esfuerzo que tomaría implementarlo. Para la estimación del tamaño de un sistema a partir de sus requerimientos, una de las técnicas más difundidas es el Análisis de Puntos de Función. Ésta técnica permite cuantificar el tamaño de un sistema en unidades independientes del lenguaje de programación, las metodologías, plataformas y/o tecnologías utilizadas, denominadas Puntos de Función.

Por otro lado, el SEI (del inglés, Software Engineering Institute) propone desde hace algunos años un método para la estimación del esfuerzo llamado COCOMO II. Éste método está basado en ecuaciones matemáticas que permiten calcular el esfuerzo a partir de ciertas métricas de tamaño estimado, como el Análisis de Puntos de Función y las líneas de código fuente (en inglés SLOC, Source Line Of Code).

En relación a los Puntos de Función, las transacciones se clasifican de la siguiente manera:

- ✓ Entradas Externas (EI)
- ✓ Salidas Externas (EO)
- ✓ Consultas Externas (EQ)

En relación a los Puntos de Función, los archivos se clasifican de la siguiente manera:

- ✓ Archivos Lógicos Internos (ILF)
- ✓ Archivos de Interfaz Externos (EIF)

Entradas Externas: No hay

Tabla 11: Salidas Externas

Nombre de la salida externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (Baja, Media, Alta)
Mostrar el contenido del libro “El Vuelo de la Mariposa Amarilla”.	1	1	Baja
Mostrar el contenido del libro “Rumbo al Socialismo”	1	1	Baja
Mostrar el contenido del libro “La Gaviota Humanocrática”	1	1	Baja
Mostrar el contenido del libro “Un Grano de Maíz I”	1	1	Baja
Mostrar el contenido del libro “Un Grano de Maíz II”	1	1	Baja
Mostrar el contenido del libro “Un Grano de Maíz III”	1	1	Baja
Mostrar el contenido referente a un tópico seleccionado de un libro.	1	2	Baja
Mostrar contenido relacionado con la palabra caliente seleccionada.	1	2	Baja
Reproducir música de fondo	1	1	Baja

Tabla 12: Consultas Externas

Nombre de la consulta externa	Cantidad de archivos referenciados	Cantidad de elementos de datos	Clasificación (Baja, Media, Alta)
Realizar búsqueda de datos.	1	2	Baja

Tabla 13: Ficheros lógicos internos

Nombre del fichero interno	Tipos de registro	Cantidad de elementos de datos	Clasificación (Baja, Media, Alta)
Música	1	1	Baja
Libros	2	10	Baja
Glosario	1	3	Baja

La información asociada al fichero interno libros es: número, título, autor, clasificación, cantidad de tópicos y cantidad de páginas. Cada tópico a su vez tiene como información asociada un nombre, la cantidad de páginas, autor y si tiene o no fecha. Se tienen entonces dos tipos de registro, la

información de los libros y la información de los tópicos. El total de elementos de datos es 10, 6 pertenecientes al registro de la información de los libros y 4 al registro de la información de los tópicos. Para el fichero música se tiene un único registro con un solo elemento de dato, el nombre de la canción. El Glosario tiene un único registro: la información asociada a los términos, y 3 elementos de datos, el identificador del término, el título y la descripción.

Tabla 14: Puntos de Función

	Total	Complejidad						Aporte	
		Simple	Valor	Media	Valor	Compleja	Valor		
Salidas Externas	9	9	4	-		-		36	
Consultas Externas	1	1	3	-		-		3	
Archivo lógicos internos	3	3	3	-		-		9	
								48	Total

Con la suma total de los aportes de todos los elementos se obtienen los Puntos de Función sin ajustar:

$$\text{UFP (Puntos de Función sin ajustar)} = 36 + 3 + 9 = 48$$

Una vez que se han obtenido los Puntos de Función sin ajustar del sistema se puede estimar el esfuerzo, para esto se utilizará el método COCOMO II. Éste método es el preferido en la actualidad para la estimación del esfuerzo cuando no se tiene información histórica a la cual recurrir. Está basado en dos modelos: uno aplicable al comienzo de los proyectos (Diseño preliminar, en inglés Early Design) y otro aplicable luego del establecimiento de la arquitectura del sistema (Post arquitectura, en inglés Post Architecture).

Se utilizará el modelo Diseño preliminar, el cual contempla la exploración de las arquitecturas alternativas del sistema y los conceptos de operación. En esta etapa no se sabe lo suficiente del proyecto como para hacer una estimación fina. Ante ésta situación, el modelo propone la utilización de Puntos de Función como medida de tamaño y un conjunto de 7 factores (cost drivers) que afectan al esfuerzo del proyecto. Estos 7 factores son agrupaciones de los factores que se utilizan en la otra variante del modelo (Post Arquitectura).

Características	Valor
Puntos de función desajustados	48
Lenguaje	ActionScript 2.0
Instrucciones fuentes por puntos de función(SLOC/UFP)	66
Instrucciones fuentes	3168

4.3 Estimación del Esfuerzo, de la cantidad de hombres, del tiempo de desarrollo y del costo.

$$PM_{\text{nominal}} = A \times (\text{Size})^B$$

Donde:

PM_{nominal}: es el esfuerzo nominal requerido en meses-hombre.

Size: es el tamaño estimado del software, en miles de líneas de código (KSLOC) o en Puntos de Función sin ajustar (convertibles a KSLOC mediante un factor de conversión que depende del lenguaje y la tecnología).

A: es una constante que se utiliza para capturar los efectos multiplicativos en el esfuerzo requerido de acuerdo al crecimiento del tamaño del software. El modelo la calibra inicialmente con un valor de 2.94.

B: es una constante denominada Factor escalar, la cual tiene un impacto exponencial en el esfuerzo y su valor está dado por la resultante de los aspectos positivos sobre los negativos que presenta el proyecto.

Cálculo del Factor Escalar (B)

$$B = 0.91 + 0.01 \times \Sigma (SF_i)$$

SF_i= valor de la variable escalar

Variables

- ✓ PREC, variable de precedencia u orden secuencial del desarrollo.
- ✓ FLEX, variable de flexibilidad del desarrollo.
- ✓ RSEL, indica la fortaleza de la arquitectura y métodos de estimación y reducción de riesgos.

- ✓ TEAM, esta variable refleja la cohesión y madurez del equipo de trabajo.
- ✓ PMAT, relaciona el proceso de madurez del software

Tabla 15: Variables escalares

Variable	Descripción	Nivel de cuantificación(Muy bajo, Bajo, Nominal, Alto, Muy alto y Extra alto)	Valores
PREC	Se tiene algo de precedencia en el trabajo con este tipo de sistemas.	Nominal	3.72
FLEX	Algo de relajación en cuanto a la flexibilidad del desarrollo.	Nominal	3.04
RSEL	No todos los riesgos se han mitigado	Nominal	4.24
TEAM	La interacción del equipo de desarrollo es altamente cooperativa.	Muy Alto	1.10
PMAT	La madurez del proceso de software es baja.	Baja	6.24

$$\Sigma SF_i = 3.72 + 3.04 + 4.24 + 1.10 + 6.24 = 18.34$$

$$B = 0.91 + 0.01 * 18.34 = 1.0934$$

$$A = \text{Se toma el valor por defecto del modelo} = 2.94$$

Size: se calcula como el producto de los puntos de función sin ajustar por un factor de conversión que depende del lenguaje a utilizar en el desarrollo del sistema.

$$\text{Size} = 48 * 66 = 3168 \text{ SLOC} = 3.2 \text{ KSLOC}$$

$$PM_{\text{nominal}} = 2.94 * (3.2)^{1.0934}$$

$$PM_{\text{nominal}} = 2.94 * 3.567 = 10.4876$$

Para completar la estimación, hay que ajustar el esfuerzo nominal de acuerdo a las características del proyecto.

$$PM = PM_{\text{nominal}} \times \prod (ME_i)$$

Donde:

Los ME_i (multiplicadores de esfuerzo) varían en función del modelo de estimación seleccionado (Diseño Preliminar o Post arquitectura). Representan las características del proyecto y expresan su impacto en el desarrollo total del producto de software. En este caso se utiliza el modelo Post arquitectura.

Multiplicadores de esfuerzo:

- ✓ PERS: Capacidad del personal.
- ✓ RCPX: Complejidad del producto.
- ✓ RUSE: Reusabilidad.
- ✓ PDIF: Dificultad de la plataforma.
- ✓ PREX: Experiencia del personal.
- ✓ SCED: Calendario.
- ✓ FCIL: Facilidades.

Tabla 16: Multiplicadores de esfuerzo

Multiplicador	Descripción	Nivel de cuantificación(Extra bajo, Muy bajo, Bajo, Nominal, Alto, Muy alto y Extra alto)	Valores
PERS	Se tienen analistas y programadores con alta capacidad de trabajo en equipo	Alto	0.83

Capítulo 4: Estudio de factibilidad

	y eficientes.		
RCPX	Las exigencias de confiabilidad, documentación y volumen de datos son moderados, y la complejidad del producto es simple.	Nominal	1
RUSE	Se pretende reutilizar el código para futuros productos en el proyecto.	Nominal	1
PDIF	No existen restricciones en cuanto al tiempo del CPU o al consumo de memoria, la plataforma es muy estable.	Baja	0.87
PREX	El personal del equipo de trabajo, dígame analistas y programadores tienen aproximadamente 1 año de experiencia en el trabajo con las herramientas y el lenguaje utilizado.	Nominal	1
SCED	Se requiere terminar el proyecto en el tiempo estimado.	Alto	1
FCIL	Se tienen herramientas CASE simples e infraestructura de comunicación básica	Bajo	1.10

$$ME = 0.83 * 1 * 1 * 0.87 * 1 * 1 * 1.10 = 0.79431$$

$$\pi (ME_i) = 0.79431$$

$$PM = 10.4876 * 0.79431 = 8.33$$

PM= 8.33 meses- hombres

Ya encontrado el esfuerzo (PM), se aplican algunas fórmulas de Boehm para calcular el tiempo de desarrollo de la aplicación (TDEV).

$$TDEV = 3.67 * (PM)^{0.28 + 0.002 * \sum SFi}$$

$$TDEV = 3.67 * (8.33)^{0.28 + 0.002 * 18.34} = 3.67 * (8.33)^{0.31668}$$

$$TDEV = 1.9568 = 2 \text{ meses}$$

Para estimar cuántas personas requiere el desarrollo:

$$CH = PM / TDEV$$

$$CH = 8.33 / 2 = 4.165$$

$$CH = 5 \text{ hombres}$$

Costo

$$\text{Costo} = CHM * PM$$

$$CHM = CH * \text{Salario mínimo}$$

Considerando que el producto es desarrollado por estudiantes de 1ro a 5to año de la Universidad de las Ciencias Informáticas, el salario mínimo a considerar es de \$75.

$$CHM = 5 * 75 = 375$$

$$\text{Costo} = 375 * 8.33 = 3123.75$$

$$\text{Costo} = \$ 3123.75$$

4.4 Beneficios tangibles

- ✓ Documentación necesaria para implementar la aplicación.
- ✓ Reusabilidad de código.

- ✓ Multimedia Colección Antonio Guiteras y Carlos Aponte: seis libros para la formación de conciencia política.
- ✓ Aporte económico de 3.332 839 dólares a nuestro país.

4.5 Beneficios intangibles

Los beneficios intangibles asociados al desarrollo del software con tecnología multimedia “Colección Antonio Guiteras y Carlos Aponte: Seis libros para la formación de conciencia política” serán expuestos a continuación:

- ✓ Motivar a los dirigentes de parroquias venezolanos a estudiar el acontecer político, tanto nacional como internacional.
- ✓ Proporcionar contenidos que tributen a la formación de valores morales y sociales.
- ✓ Crear una conciencia política en estos dirigentes de parroquias, acorde con el proceso revolucionario que hoy se vive en Venezuela.
- ✓ Contribuir a la formación de dirigentes honestos, revolucionarios, comprometidos con su patria venezolana y con la Revolución.
- ✓ Promover el Socialismo como el modo de vida y sistema económico más factible y necesario para Venezuela y los pueblos de nuestra América.

4.6 Análisis de costos y beneficios

El desarrollo de la aplicación Colección Antonio Guiteras y Carlos Aponte: seis libros para la formación de conciencia política, no necesita de grandes gastos de recursos, ni de computadoras de última generación con grandes prestaciones y capacidad de almacenamiento. Es una aplicación realizada para usuarios totalmente inexpertos en el trabajo con una computadora, por lo que tiene una interfaz amigable, de fácil entendimiento y comunicación. Su mayor aporte es la formación política de los dirigentes venezolanos que aún no están acordes con el momento histórico que vive hoy la hermana república.

Según los resultados obtenidos a través de la estimación, y analizando la relación costo-beneficios del producto, se decidió que si era factible desarrollarlo, ya que el mismo abonará al país una suma de 3.332 839 dólares, cifra que cubre el costo de la producción (\$3123) y además aporta ganancias significativas.

4.7 Conclusiones del capítulo

Es factible desarrollar el producto, el cual, según el tiempo estimado debe estar listo en 2 meses, con el trabajo de 5 hombres y un costo aproximado de 3123.75 pesos en moneda nacional. Se puntualizaron los beneficios que se obtendrán de la comercialización de esta aplicación con Venezuela, beneficios para Cuba, y para el pueblo y el gobierno venezolano, y se definió como mayor aporte, la propagación y estimulación del Socialismo en Venezuela, y a través de ella, en América.

CONCLUSIONES

- ✓ Para un exitoso desarrollo se hizo un estudio sobre la ingeniería de software aplicada a productos multimedia, donde se escogió como metodología el Proceso Unificado de Rational (RUP) unido al lenguaje de modelado ApEM-L. Mientras el primero establecía las actividades y los criterios para conducir el sistema por todo el proceso, el otro ofrecía la notación gráfica necesaria para representar los sucesivos modelos que se obtenían en el proceso de refinamiento.
- ✓ Se investigó sobre las posibles herramientas para el desarrollo de aplicaciones con tecnología multimedia y se seleccionó Macromedia Flash en su versión 8.0.
- ✓ Se realizó el análisis, el diseño y la implementación de la multimedia *Colección electrónica: Seis libros para la formación de conciencia política*, obteniendo la documentación necesaria para garantizar el futuro mantenimiento del sistema.
- ✓ Se obtuvo un producto interactivo que muestra la información solicitada por el cliente y cumple con los requisitos establecidos por el mismo.
- ✓ Con la realización de este proyecto, el país y específicamente la Universidad de las Ciencias Informáticas cumplieron con el contrato establecido con el MENPET, el cual reportará beneficios económicos considerables una vez le sea entregado el producto al cliente.

RECOMENDACIONES

- ✓ Difundir el contenido de los libros presentes en la colección a otros sectores de la población venezolana que también necesitan formación política, específicamente las clases más bajas.
- ✓ Promover el producto obtenido en otros países de América que intentan, de la misma manera, hacer una Revolución, contribuyendo con esto a formar una conciencia política en estos pueblos y a difundir la verdad sobre el Socialismo.
- ✓ Ampliar las funcionalidades del producto obtenido de manera que puedan brindarse más opciones al usuario, ejemplo de ello pudiera ser, mostrar varias canciones y permitirle que seleccione la que desea escuchar, así como, realizar búsquedas por fechas.

REFERENCIAS BIBLIOGRÁFICAS

- [1]: ORTIZ, L.A. *El libro electrónico, nueva versión de un antiguo medio*, 2007
- [2]: CAMARGO, J. *Libro impreso VS libro electrónico* [En línea]. Disponible en <http://libroselectronicos.wordpress.com/category/libro-impreso-vs-libro-electronico/>. 2007 [citado el: 23/04/2008]
- [3]: PÉREZ, Y. *Metodologías para el desarrollo de software educativo: Un estudio comparativo*. Universidad de las Ciencias Informáticas (UCI), Ciudad de la Habana, Cuba. 2006.
- [4]: PIATTINI, P. *Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión*. 1996
- [5]: VALDÉS, S. *Propuesta del proceso de producción para el departamento de multimedia educativa de la Universidad de Ciencias Informáticas*. Universidad de las Ciencias Informáticas, Ciudad de la Habana. 2005. Disponible en: http://bibliodoc.uci.cu/TD/TD_0134_05.pdf
- [6]: SCHWABE, D., Rossi, G. *Building Hypermedia Applications as Navigational Views of Information Models*. Hawái
- [7]: BOOCH, G., JACOBSON, I. y RUMBAUGH, J. *El Proceso Unificado de Desarrollo de Software*, Madrid, Editorial Pearson Educación, S. A., 2000.
- [8]: GSInnova. *Ayuda de Rational Rose*. 2007.
- [9]: BECK, K. *Extreme Programming Explained: Embrace Change*. Editorial Addison Wesley Longman, 2000.
- [10]: CALERO, M. *Una explicación de la programación extrema (XP)*. V Encuentro usuarios xBase 2003. MADRID.
- [11]: RAMOS, Y. *Conferencia: Metodologías Ágiles de Desarrollo de Software*. Universidad de las Ciencias Informáticas. Ciudad de la Habana. 2008
- [12]: BOOCH, G., JACOBSON, I. y RUMBAUGH, J. *El Lenguaje Unificado de Modelado. Manual de Referencia* (1ra ed.). Madrid, Editorial Addison-Wesley. 2000.

Referencias Bibliográficas

- [13]: SAUER, S. y ENGELS, G. *Extending UML for Modeling of Multimedia Applications* [En línea]. Disponible en <http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf> . 2001. [citado el: 14/03/2008]
- [14]: MSc. CIUDAD, F. *ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UC*. Universidad de las Ciencias Informáticas (UCI), Ciudad de la Habana. 2007.
- [15]: MSc. CIUDAD, F. *Conferencia sobre nueva versión de ApEM-L*, Universidad de las Ciencias Informáticas, Ciudad de la Habana, 25 de marzo del 2008 y 24 de abril del 2008.
- [16]: MALAGÓN, M. y FRÍAS, Y. *El libro electrónico: el caso de Tecnologías de información y las comunicaciones aplicadas a la Educación*. [En línea]. Disponible en: <http://www.monografias.com/trabajos31/libro-multimedial/libro-multimedial.shtml> [citado el 23/02/2008]
- [17]: FELIPE, R. *COMPARATIVA DE SISTEMAS MULTIMEDIA EN EL DISEÑO DE UNIDADES DIDÁCTICAS EN CIENCIAS*. [En línea]. Disponible en: <http://www.sociedadelainformacion.com/fisica/multimedia/multimedia.htm>. 1999. [citado el 03/12/2007]
- [18]: ZAMBRANO, D. *MULTIMEDIA, ToolBook (Windows)*. [En línea]. Disponible en: <http://www.monografias.com/trabajos10/mmedia/mmedia.shtml#toolbook> . 1997 [citado el 23/03/2008]
- [19]: PROGRAMADORES, C. D. *DICCIONARIO INFORMÁTICO 2000*. [En línea]. Disponible en: <http://www.lawebdelprogramador.com/diccionario/mostrar.php?letra=T&pagina=4> [citado 2/2008]
- [20]: VAN DER HENST, C. *Flash, la tecnología multimedia para el web*. [En línea]. Disponible en: <http://www.maestrosdelweb.com/editorial/flash/>. 1999. [citado el 14/03/2008]
- [21]: AULACLIC: *Introducción a Flash 8 (I), De Flash MX 2004 a Flash 8*. [En línea]. Disponible en: http://www.aulaclit.es/flash8/t_1_1.htm. 2006 [citado el 21/01/2008]
- [22]: WIKIPEDIA 2008 [Disponible en: <http://es.wikipedia.org/wiki/XML>]

Referencias Bibliográficas

- [23]: *Curso Online Programación con ActionScript 2.0*. [En línea] Disponible en: http://www.virtual-formac.com/informatica/programacion/curso_programacion_con_actionscript_2_0-c6408.html. [citado el: 18/05/2008]
- [24]: PAVÓN, M. *Rational Rose*, Universidad de Madrid. [En línea] Disponible en: <http://www.fdi.ucm.es/profesor/jpavon/is2/Lab01RationalRose.pdf> . [citado el 16/06/2008]
- [25]: BAUMEISTIER, H., KOCH, N. y MANDEL, L. *Towards a UML Extension for Hypermedia Design*. [En línea]. Disponible en: <http://citeseer.ist.psu.edu/cache/papers/cs/31048/http:zSzzSzwww.fast.dezSzProjektezSzforsoftzSzuml99zSzuml99.pdf/baumeister99towards.pdf>. 2001. [citado el 10/01/2008]
- [26]: PRESSMAN, R. *Ingeniería del Software. Un enfoque práctico*. (5ta ed.). New York, 2002.

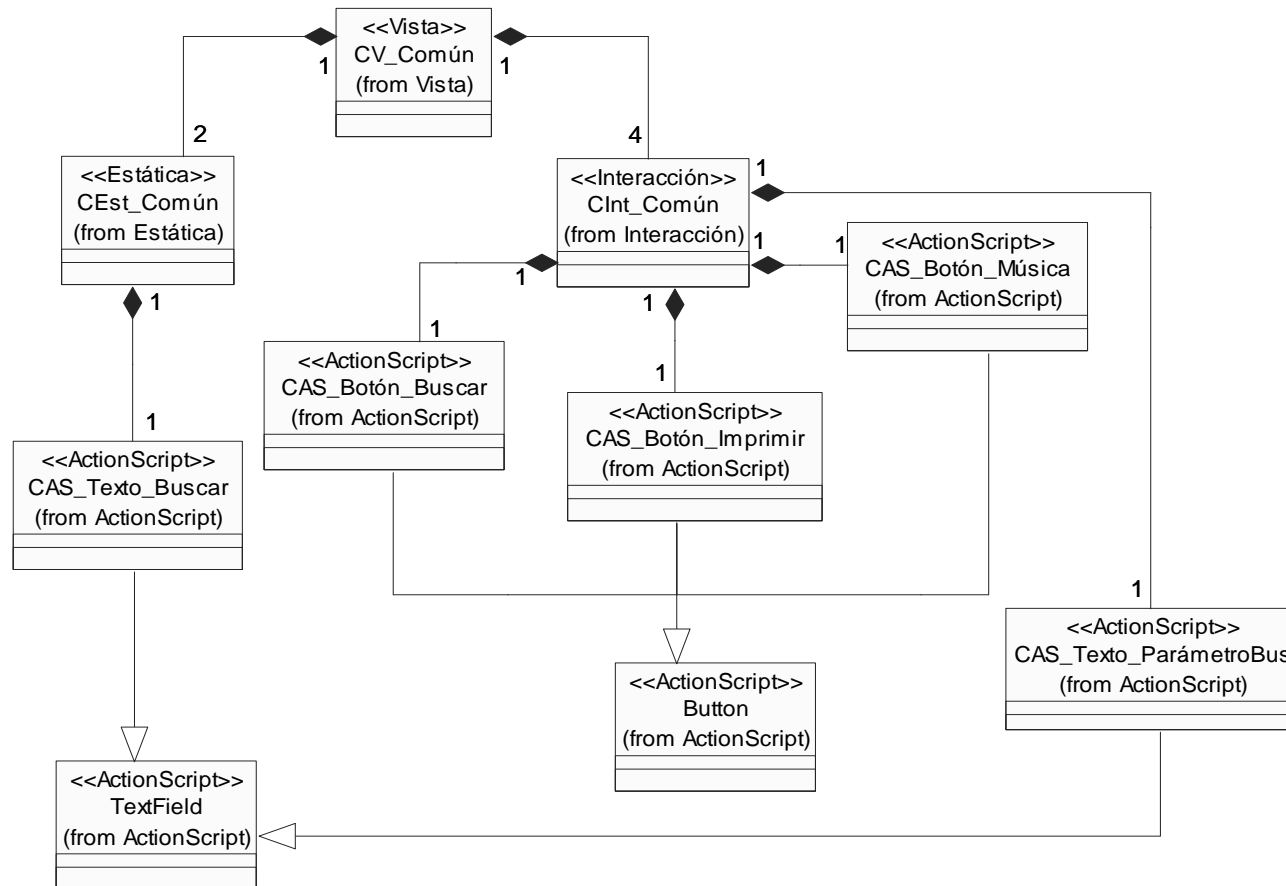
BIBLIOGRAFÍA

1. BOOCH, G., JACOBSON, I. y RUMBAUGH, J. *El Proceso Unificado de Desarrollo de Software*. Madrid, Editorial Pearson Educación, S. A. 2000.
2. CALDERÓN, A. y otros. *Metodologías Ágiles*. Universidad Nacional de Trujillo, Perú. 2007
3. CALERO, M. *Una explicación de la programación extrema (XP)*. 2003
4. CIUDAD, F. *ApEM-L como una nueva solución a la modelación de aplicaciones educativas multimedia en la UC*", Universidad de las Ciencias Informáticas (UCI), Ciudad de la Habana. 2007.
5. CORRALES, C. *LA TECNOLOGIA MULTIMEDIA: Una Nueva Tecnología de Comunicación e Información. Características, concepciones y aplicaciones*. [En línea]. Disponible en: <http://iteso.mx/%7Ecarlosc/pagina/documentos/multidef.htm>. 1994. [citado el 2/04/08].
6. *Curso Online Programación con ActionScript 2.0*. [En línea] Disponible en: http://www.virtualformac.com/informatica/programacion/curso_programacion_con_actionscript_2_0-c6408.html. [citado el: 18/05/2008]
7. DÍAZ, S. y otros, *Ingeniería para productos con tecnología multimedia*, II Taller de Software Educativo y Multimedia, julio 2007.
8. *Diccionario de la lengua española, vigésima segunda edición*. [en línea] Disponible en: <http://www.rae.es/rae.html>
9. ESTRADA DE LÓPEZ, P. *Diagramas de colaboración*, Universidad de El Salvador. 2006
10. FERNÁNDEZ, G. *Introducción a Extreme Programming*, diciembre 2002.
11. GACITUÁ, RA. *METODOS DE DESARROLLO DE SOFTWARE: EL DESAFIO PENDIENTE DE LA ESTANDARIZACION*, Universidad del Bío-Bío, Chile. 2003.
12. GARCÍA, J. *UML: Diagramas UML. ¿Qué es UML?* [En línea] Disponible en: <http://www.ingenierosoftware.com/analisisydiseno/uml.php>. 2005. [citado el 26/01/2008].
13. BECK, K. *Extreme Programming Explained: Embrace Change*. Editorial Addison Wesley Longman, 2000.

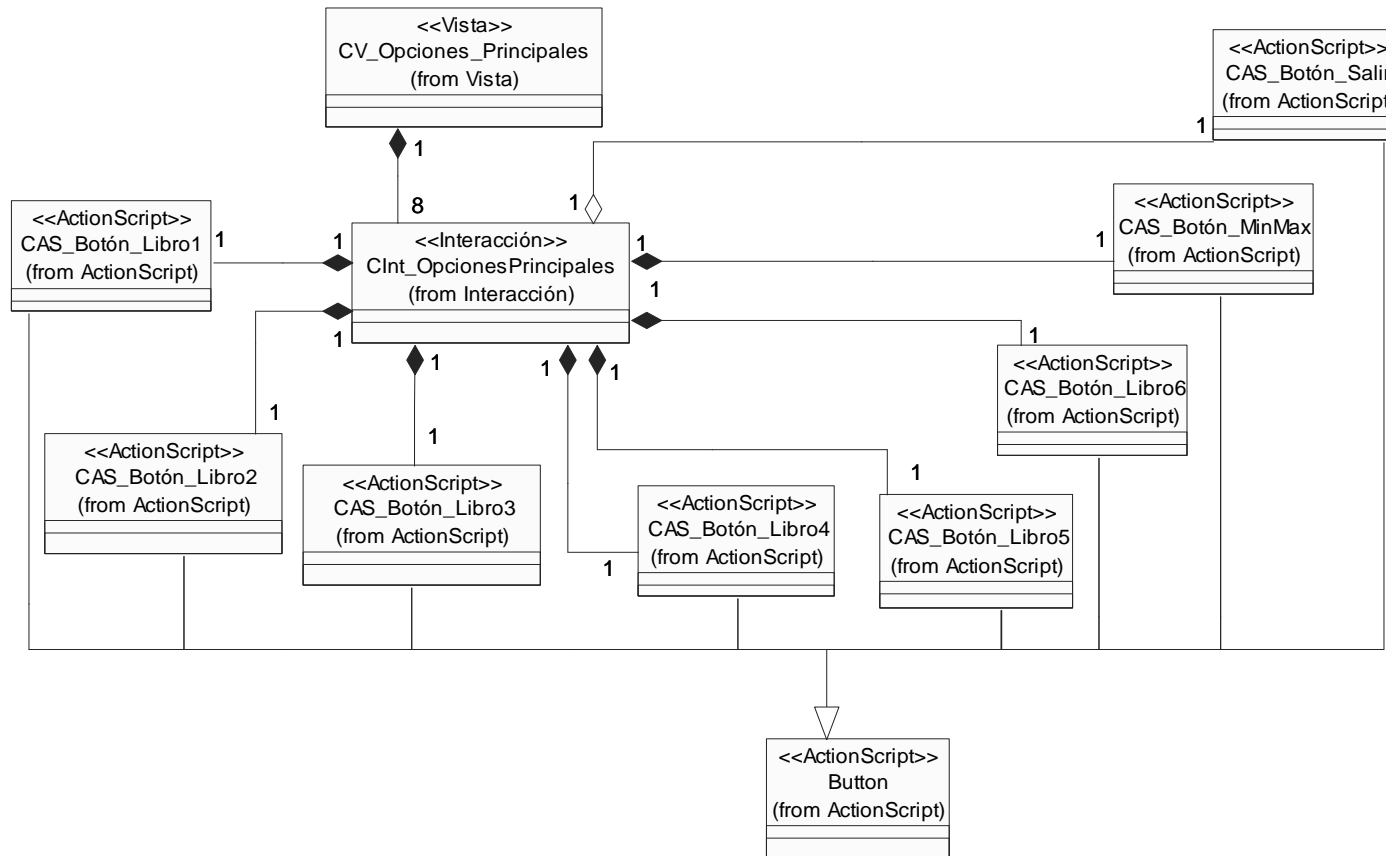
14. MARQUÉS, P. *Multimedia Educativa: Clasificación, Funciones, Ventajas e Inconvenientes*. [En línea]. Disponible en: <http://dewey.uab.es/pmarques/funcion.htm> 2004. [citado el: 22/11/2007]
15. MARQUÉS, P. *ENTORNOS FORMATIVOS MULTIMEDIA: ELEMENTOS, PLANTILLAS DE EVALUACIÓN/CRITERIOS DE CALIDAD*. [En línea]. Disponible en: <http://dewey.uab.es/pmarques/calidad.htm>. 2005. [citado el 18/11/2008].
16. MENDOZA, MA. *Metodologías de Desarrollo de Software*. [En línea] Disponible en: http://www.informatizate.net/articulos/metodologias_de_desarrollo_de_software_07062004.html 2004. [citado el 22/01/2208]
17. PERALTA, M. *Estimación del esfuerzo basada en Casos de Uso*. Centro de Ingeniería del Software e Ingeniería del Conocimiento (CAPIS), Buenos Aires.
18. PÉREZ, Y. *Metodologías para el desarrollo de software educativo: Un estudio comparativo*. Universidad de las Ciencias Informáticas (UCI), Ciudad de la Habana. 2006.
19. PRESSMAN, R. *Ingeniería del Software. Un enfoque práctico*. (5ta ed.). New York, 2002.
20. SAUER, S. *UML & OMMMA-L, 2001*
21. TERUEL, A. *COCOMO II: Una Familia de Modelos de Estimación*. [En línea]. Disponible en: <http://www ldc.usb.ve/~teruel/ci4713/clases2001/cocomo2.html> . 2001. [citado el 06/05/2008]
22. VAN DER HENST, C. *Flash, la tecnología multimedia para el web*. [en línea]. Disponible en: <http://www.maestrosdelweb.com/editorial/flash/> . 1999. [citado el 20/03/08].
23. FELIPE, R. *COMPARATIVA DE SISTEMAS MULTIMEDIA EN EL DISEÑO DE UNIDADES DIDÁCTICAS EN CIENCIAS*. [En línea]. Disponible en: <http://www.sociedadelainformacion.com/fisica/multimedia/multimedia.htm>. 1999. [citado el 07/12/2008]

Anexos

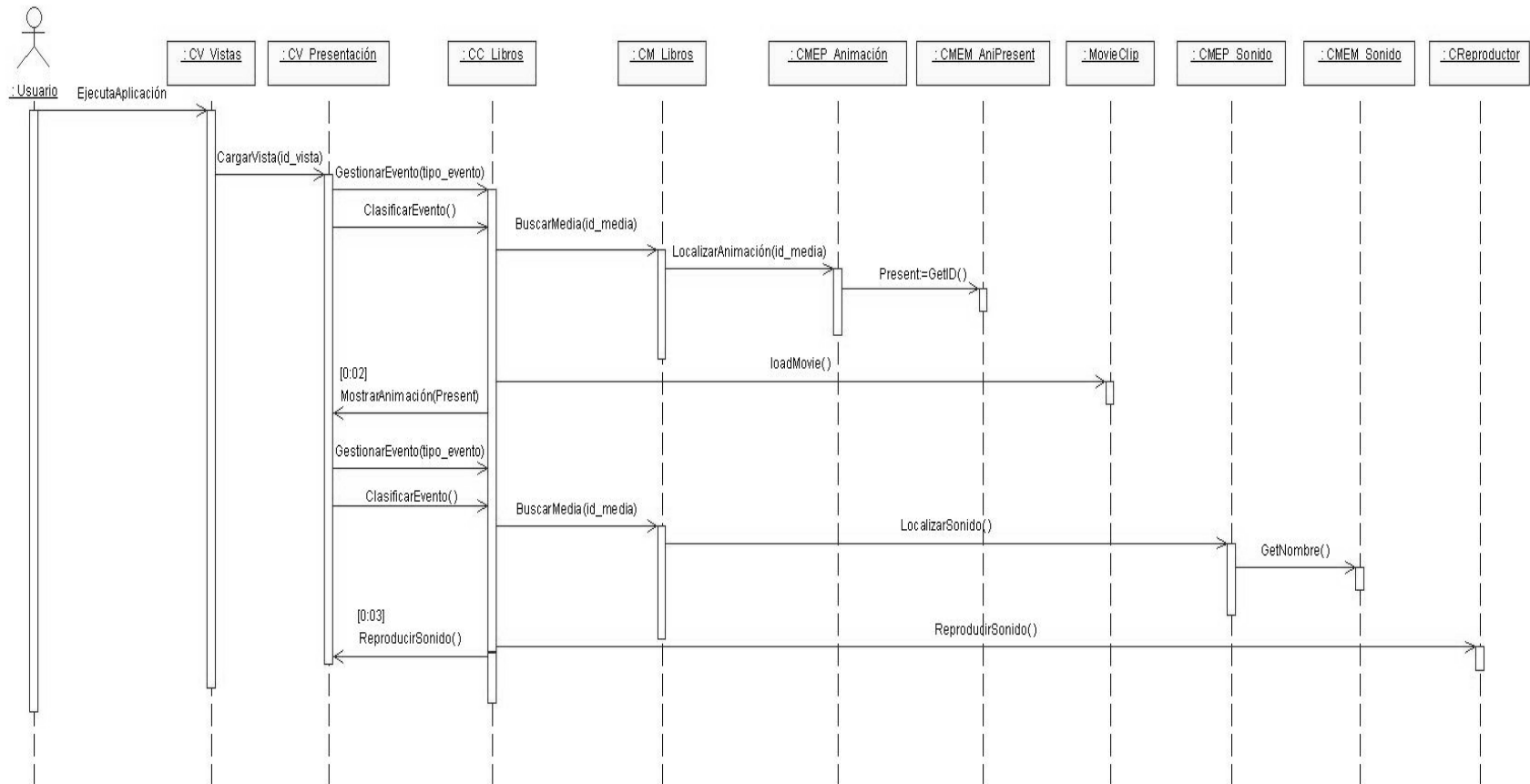
Anexo 1: DEP Vista Común



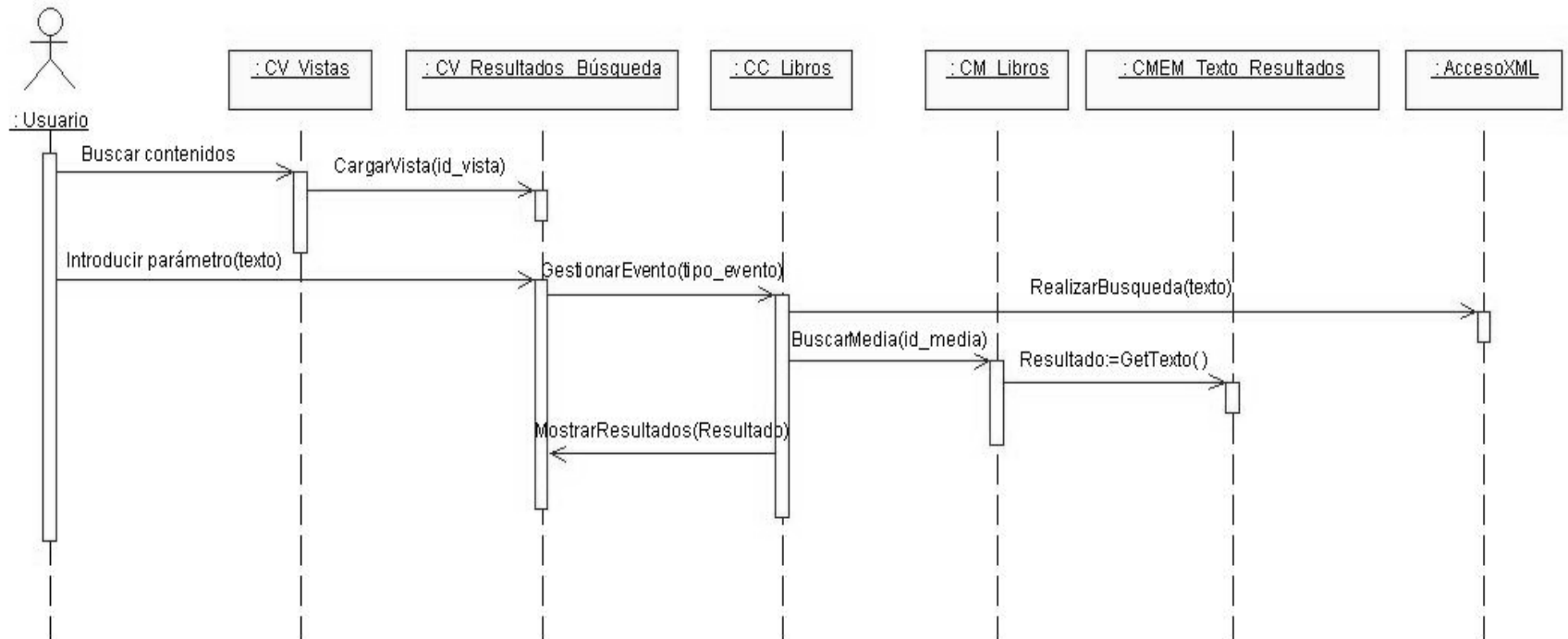
Anexo 2: DEP Vista Opciones_Principales



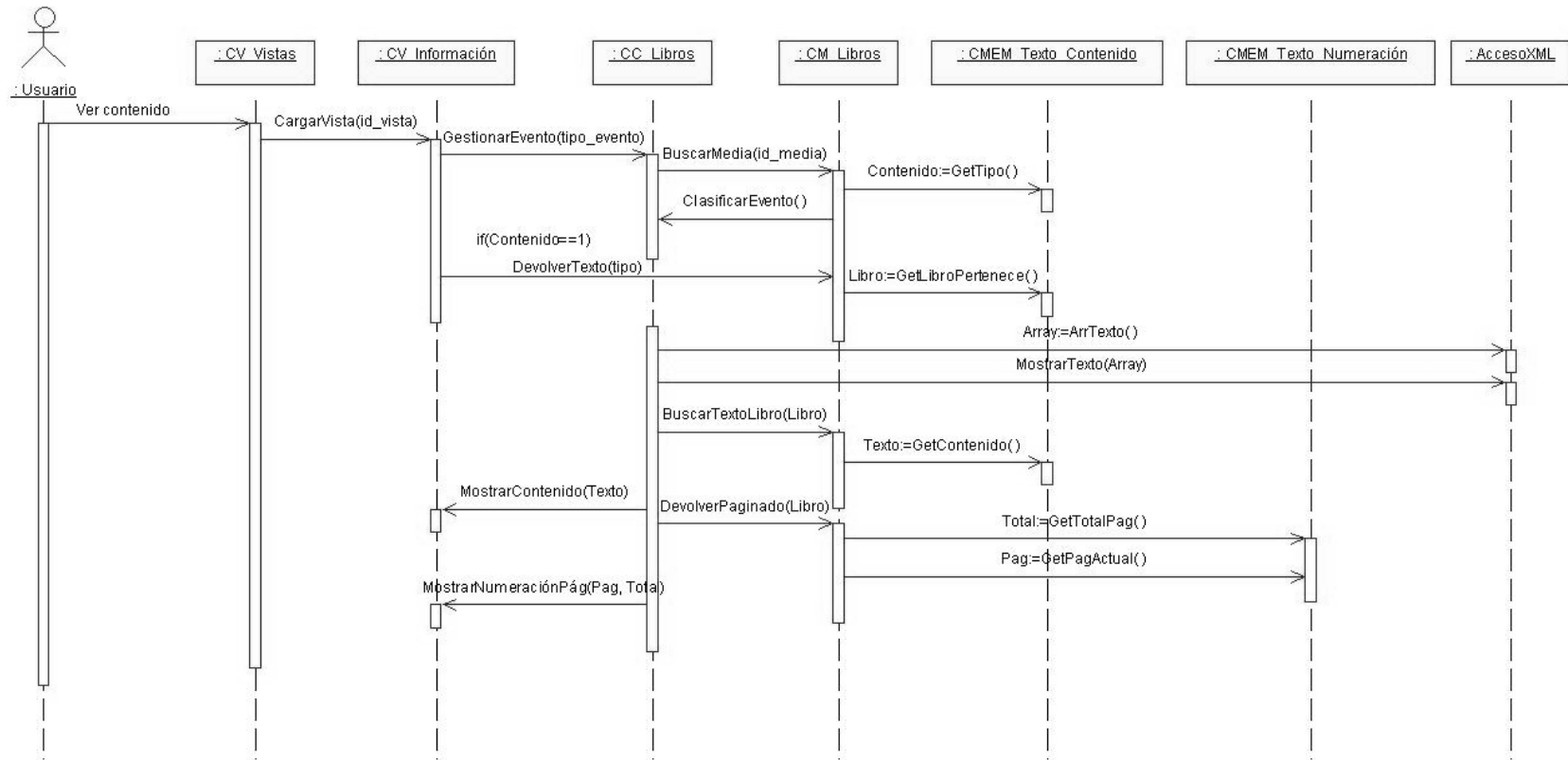
Anexo 3: Diagrama de Secuencia de la Vista Presentación



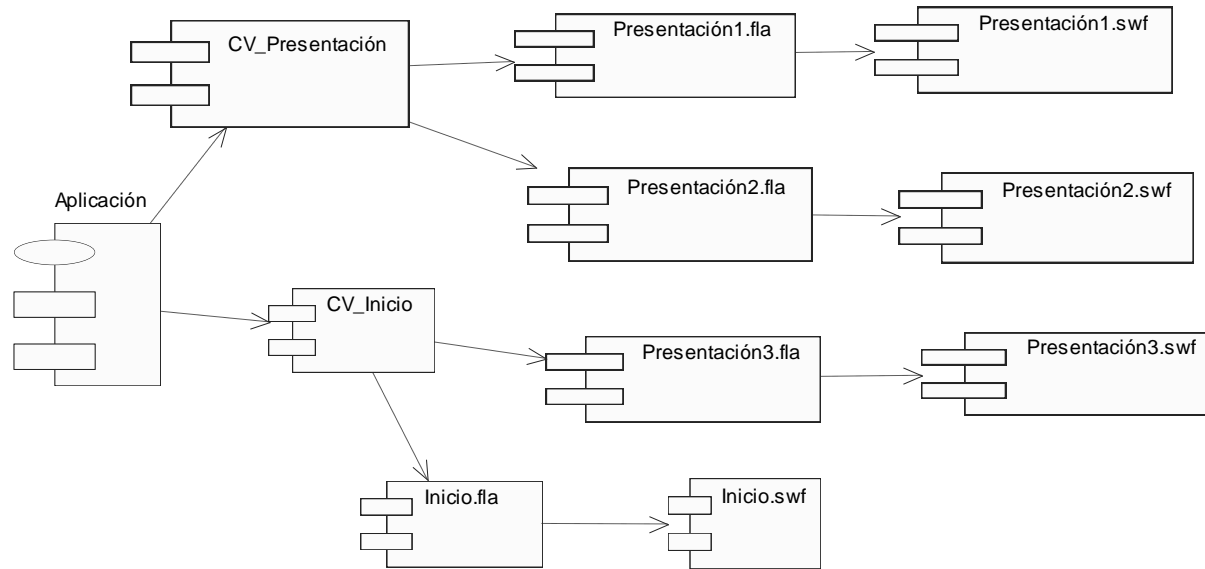
Anexo 4: Diagrama de Secuencia de la Vista Resultados_Búsqueda



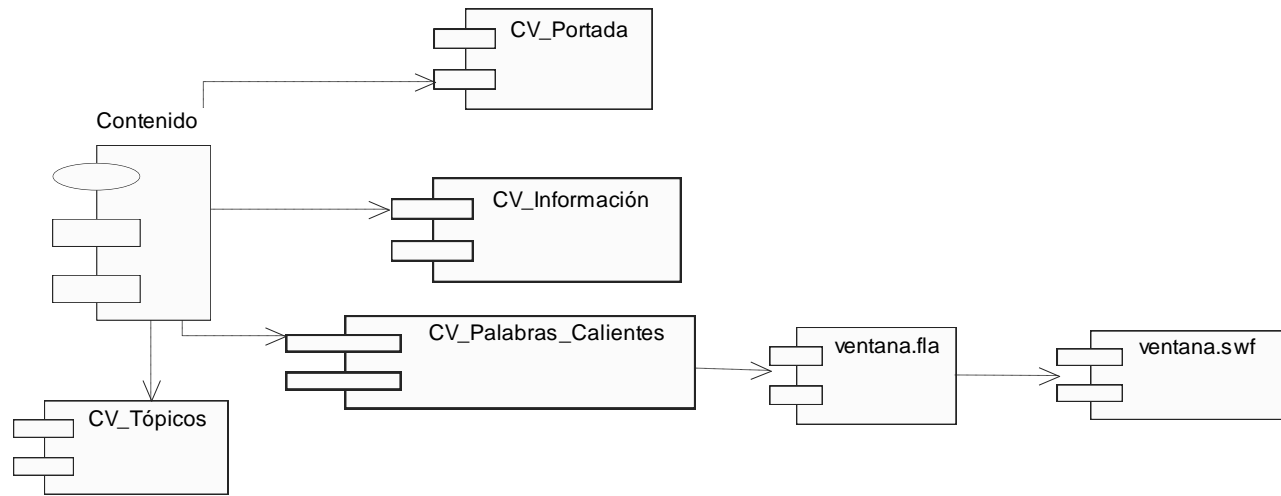
Anexo 5: Diagrama de Secuencia de la Vista Información



Anexo 6: Diagrama de Componentes del Paquete Aplicación



Anexo 7: Diagrama de Componentes del Paquete Contenido



GLOSARIO DE TÉRMINOS

Actor: Representa el rol que desempeñan en el sistema descrito, una o varias personas, un equipo o un sistema automatizado.

Archivos Lógicos Internos (ILF): Grupo de datos relacionados lógicamente e identificables por el usuario, que residen enteramente dentro de los límites del sistema y se mantienen a través de las Entradas Externas.

Archivos de Interfaz Externos (EIF): Grupo de datos relacionados lógicamente e identificables por el usuario, que se utilizan solamente para fines de referencia. Los datos residen enteramente fuera de los límites del sistema y se mantienen por las Entradas Externas de otras aplicaciones.

CASE: Siglas que corresponden a las iniciales de: **C**omputer **A**ided **S**oftware **E**ngineering, que en su traducción al español significa Ingeniería de Software Asistida por Computación. Es la aplicación de tecnología informática a las actividades, las técnicas y las metodologías propias de desarrollo, su objetivo es acelerar el proceso de automatizar o apoyar una o más fases del ciclo de vida del desarrollo de sistemas.

Clase Botón: Denota la existencia de un elemento interactivo del tipo botón para producir un camino en la navegación hacia una clase vista, y utilizando como intermediarias a clases de tipo consulta o índice.

Clase Consulta: Permite el enlace con una clase vista a y través de un valor directo (variable de consulta) asignado a la búsqueda para la visualización del elemento a mostrar, que se sitúa como identificador en la relación entre las clases.

Clase Índice: Denota el valor de direccionamiento hacia una clase vista a partir de una opción determinada.

Clase Modelo Entidad Media: Representan los recursos mediáticos de la aplicación (imágenes, texto, sonido, video o animación)

Clase Menú: Es el elemento de composición de una clase vista, desde donde se puede llegar a otras diferentes clases vistas con las cuales se conecta este menú, pues contiene una lista de las opciones de movimiento siguiente, pero no los valores de esas opciones, los cuales serán ofrecidos por el resto de los tipos de clases. Dicho menú no tiene que necesariamente llevar a una clase vista directamente, sino que puede ser a través de otro tipo de clase de navegación (guía, consulta o índice).

Componente: Parte modular de un sistema, desplegable y reemplazable que encapsula implementación y un conjunto de interfaces y proporciona la realización de los mismos. Un componente típicamente contiene clases y puede ser implementado por uno o más artefactos (archivos ejecutables, binarios, etc.).

Consulta Externa (CE): Proceso elemental con componentes de entrada y de salida donde un Actor del sistema rescata datos de uno o más Archivos Lógicos Internos o Archivos de Interfaz Externos. Los datos de entrada no actualizan ni mantienen ningún archivo (lógico interno o de interfaz externo) y los datos de salida no contienen datos derivados (es decir, los datos de salida son básicamente los mismos que se obtienen de los archivos).

Diagrama: Representación gráfica en el que se muestran las relaciones entre las diferentes partes de un conjunto o sistema.

Entrada Externa (EI): Proceso elemental mediante el cual ciertos datos cruzan la frontera del sistema desde afuera hacia adentro.

Mensaje: Comunicación entre objetos que contiene información y espera la realización de una acción.

Modelo: Es una vista de un sistema del mundo real, es decir, una abstracción de dicho sistema considerando un cierto propósito.

OCL: (Object Constraint Language, lenguaje de restricciones de objetos). Es un lenguaje notacional, un subconjunto del UML estándar industrial, que permite a los desarrolladores de software escribir restricciones sobre modelos de objetos (pre y Poscondiciones, guardas, invariantes, valores derivados, restricciones sobre operaciones, etc.). Estas restricciones son particularmente útiles, en la medida en que permiten a los desarrolladores crear un amplio conjunto de reglas que rigen el aspecto de un objeto individual. Es un lenguaje para la descripción textual precisa de restricciones que se aplican a los modelos gráficos UML.

Requerimiento: Una característica, propiedad o comportamiento que se desea para el sistema.

Salida Externa (SE): Proceso elemental con componentes de entrada y de salida mediante el cual datos simples y datos derivados (datos que se calculan a partir de otros datos) cruzan la frontera del sistema desde adentro hacia afuera. Adicionalmente, las Salidas Externas pueden actualizar un Archivo Lógico Interno.

SWF: ShockWave Flash. Extensión de archivo de animación digital creado con Macromedia Flash y exportado con Macromedia Shockwave que puede ser visualizado independientemente, o desde una obra hecha con Director, o por un visor o browser de páginas Web en Internet.