

Universidad de las Ciencias Informáticas

Facultad 8



“ANÁLISIS Y DISEÑO DEL PRODUCTO ARTEFACTOS ELECTRODOMÉSTICOS VOL. 1.”

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

Autor

José Andrés Noriega García

Tutor

Ing. Yobannys Cabrera González

Ciudad de la Habana

Julio 2007

Año 49 de la Revolución

DECLARACIÓN DE AUTORÍA

Declaro ser autor de la presente tesis y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

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

José Andrés Noriega García Ing. Yobannys Cabrera González

Firma del Autor Firma del Tutor

*“... lo que da al hombre el poder no es el mero conocimiento que viene del uso de los sentidos,
sino, ese otro conocimiento más profundo que se llama Ciencia.”*

José Martí

Agradezco la realización de esta investigación a aquellas personas que fueron capaces de ayudarme con la búsqueda de información, a Yerena Sosa Calderón, a Juan José Solano Pérez, a todos mis compañeros y compañeras de aula (y de año) por hacer mas fáciles todos estos años, profesores del mismo centro, a mi familia por su apoyo y confianza, a nuestra Revolución y nuestro Comandante por darnos todo para poder ser alguien en la vida.

....a todos muchas gracias

Dedico este trabajo a mi madre Regla, a mi padre José Andrés, a mis hermanos Lansana y Alseni, a mi novia Betty y a todas las personas que colaboraron en la realización de este trabajo.

En el presente trabajo se pretende abordar la creación de una multimedia para proporcionar conocimientos básicos acerca de los tipos de artefactos electrodomésticos existentes, conocer sus componentes y funciones. Se realiza un estudio de la actualidad en las diferentes comunidades venezolanas teniendo presente el grado de desconocimiento en cuanto al mantenimiento y reparación de los artefactos electrodomésticos de uso común en el hogar. Se analizan las posibles herramientas para su desarrollo teniendo en cuenta el conocimiento existente en proyectos anteriores y las necesidades del cliente de la utilización de una única herramienta para facilitar el soporte y en un futuro el mantenimiento del producto resultante de la misma. Utilizando RUP (Proceso Unificado de Desarrollo) se desarrolla el levantamiento de los requerimientos del cliente, luego el análisis y diseño de la solución propuesta. Se describen todas las clases para una mejor comprensión de los programadores que asumen la tarea de implementación, realizándose también el diagrama de componentes para una mejor división de las tareas para el producto a desarrollar. Se realiza el estudio de factibilidad para la realización de la misma a través del método de puntos de caso de uso para la estimación del esfuerzo y el tiempo de desarrollo.

INTRODUCCIÓN.....	1
CAPÍTULO 1 FUNDAMENTACIÓN TEÓRICA.....	5
1.1 Introducción.....	5
1.2 Estado del arte.....	5
1.2.1 Ventajas de la hipermedia.....	7
1.2.2 Inconvenientes de la hipermedia.....	8
1.3 Descripción del objeto de estudio.....	8
1.3.1 Descripción General.....	8
1.3.2 Identificación de la audiencia.....	10
1.5 Análisis del modelo de arquitectura de información utilizada.....	11
1.5.1 Principios y normas de diseño.....	11
1.5.2 Estándares de la Interfaz de la aplicación.....	13
1.5.3 Estándares de codificación.....	21
1.6 Conclusiones.....	23
CAPÍTULO 2 TENDENCIAS Y TECNOLOGÍAS.....	24
2.1 Introducción.....	24
2.2 Tendencias y Tecnologías actuales a considerar.....	25
2.2.1 Multimedia, principales aspectos.....	25
2.2.2 Principales conceptos de multimedia.....	25
2.2.3 Tendencias de las aplicaciones multimedia.....	27
2.2.4 Herramienta seleccionada: Macromedia Flash.....	29
2.2.5 Información básica de Macromedia Flash 8.....	30
2.3 Metodologías de desarrollo de software.....	33
2.3.1 MultiMet como proceso de producción.....	33
2.3.2 El Proceso Unificado de Desarrollo de Software (RUP).....	35
2.3.3 Características del Proceso Unificado de Software.....	36
2.3.4 Rational Rose y El Lenguaje Unificado de Modelado (UML).....	37
2.3.5 Objetivo de UML como lenguaje de modelado.....	38
2.3.6 Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA – L).....	39
2.3.7 Relationship Management Methodology (RMM).....	42
2.3.8 XTRAS.....	44
2.4 Conclusiones.....	45
CAPÍTULO 3 DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA.....	46
3.1 Introducción.....	46
3.2 Especificación del Contenido.....	46

3.3 Descripción del Sistema Propuesto.....	46
3.4 Modelo Conceptual.	47
3.4.1 Diagrama de Clases del Modelo del Dominio.	47
3.4.2 Análisis de los Conceptos del Dominio.	48
3.4.3 Diagrama de Navegación.	49
3.5 Modelo de Análisis.	49
3.5.1 Requisitos Funcionales.	49
3.5.2 Requisitos no Funcionales.	50
3.5.4 Navegación.	51
3.5.5 Servicios generales.	51
3.6 Modelo de casos de uso del sistema.	52
3.6.1 Determinación y justificación de los actores del sistema.	52
3.6.2 Descripción y expansión de los casos de uso.	52
3.6.3 Módulo Principal.	53
3.6.4 Módulo Electricidad.	55
3.6.5 Módulo Artefactos Térmicos.	58
3.6.7 Módulo Ejercitación.	60
3.7 Conclusiones.....	63
CAPÍTULO 4 CONSTRUCCIÓN DE LA SOLUCIÓN PROPUESTA.	64
4.1 Introducción.....	64
4.2 Modelo de Diseño.	65
4.2.1 Diagramas de Presentación.	65
4.2.2 Diagrama de componentes.....	77
4.3 Conclusiones.....	79
CAPÍTULO 5 ESTUDIO DE FACTIBILIDAD.....	80
5.1 Introducción.....	80
5.2 Planificación.	80
5.3 Costos.	80
5.3.1 Cálculo de puntos de Casos de Uso sin Ajustar.	80
5.3.2 Factor de Peso de los Actores sin ajustar.	81
Tabla 5. 1 Factor de Peso de los Actores sin ajustar.....	81
5.3.3 Factor de Peso de los Casos de Uso sin ajustar.	82
Tabla 5. 2 Factor de Peso de los Casos de Uso sin ajustar.	82
5.3.4 Cálculo de Puntos de Casos de Uso Ajustados.....	83
5.3.5 Factor de Complejidad Técnica.	83
Tabla 5. 3 Factor de Complejidad Técnica.	84
5.3.6 Factor Ambiente.	85
Tabla 5. 4 Factor Ambiente.	85
5.3.7 De los puntos de casos de uso a la estimación del esfuerzo.	86

Tabla 5. 5 Distribución del Esfuerzo.....	88
5.4 Beneficios Tangibles e Intangibles.....	89
5.5 Análisis de Costos y Beneficios.....	89
5.6 Conclusiones.....	90
CONCLUSIONES GENERALES.....	91
RECOMENDACIONES.....	92
REFERENCIAS BIBLIOGRÁFICAS.....	93
BIBLIOGRAFÍA.....	94
GLOSARIO DE TÉRMINOS.....	96

Introducción

Fidel con su visión ha definido que: “Batalla de Ideas no significa solo principios, teorías, cultura, argumentos, replicas y contrarreplicas, destruir mentiras y sembrar verdades, significa hechos y verdades concretas”. (Castro F: 2000). En los tiempos de la información y el conocimiento que vive el mundo hoy, la informática surge como uno de los recursos estratégicos vitales para el desarrollo económico y social del ser humano, la eliminación de las desigualdades sociales; el acceso y libertad a la información y a la cultura.

La rápida y frecuente producción de softwares educativos en el país para el uso interno, ha beneficiado sustancialmente la enseñanza cubana; hoy en día no existe medio docente donde no se utilicen estas aplicaciones como herramienta de aprendizaje, proporcionándoles a los estudiantes habilidades y conocimientos en el uso de las computadoras. La Informatización de la Sociedad Cubana a partir de 1996 llevó consigo la creación de Centros de Estudio de Software Educativo en los diferentes Institutos Superiores Pedagógicos del país y la instauración de bancos de software para organizar las producciones existentes. El estado cubano ha llevado a todos los centros de estudios los medios necesarios para el aprendizaje mediante la computación. Cuba cuenta con el mejor sistema de educación de América Latina, es hora de comenzar a producir conocimiento.

El Ministerio de Educación y Deportes de la República Bolivariana de Venezuela, a través de la Dirección Nacional de Formación Permanente y Supervisión pretende preparar un material didáctico sobre los artefactos electrodomésticos y sus componentes, que facilite a las comunidades conocer cada uno de los elementos que integra estos equipos que se encuentran en el hogar y su funcionamiento. Todo lo anterior está fundamentado en el estudio de conceptos básicos de electricidad y la aplicación del Efecto Joule en el funcionamiento de los electrodomésticos.

Como **situación** tenemos que las diferentes comunidades venezolanas cuentan ya con diversos equipos de uso doméstico, en la mayoría de los casos presentan total desconocimiento en cuanto al mantenimiento y reparación de dichos artefactos de uso común en el hogar, considerando que el uso continuo y frecuente origina un natural desgaste o deterioro, por lo que es necesario un mantenimiento periódico para evitar que dejen de funcionar. Esto por supuesto, redundaría en beneficio para la familia que los posee y asegura una fuente de trabajo permanente y creciente, debido al incremento en su uso y a las innovaciones que se producen, estos datos se obtuvieron a través de la realización de encuestas, entrevistas y observaciones en las diferentes comunidades venezolanas.

De esta manera se deriva como **problema científico**: ¿Cómo disminuir la desinformación existente de las comunidades venezolanas acerca del uso y mantenimiento de los artefactos electrodomésticos con la propuesta de una aplicación multimedia?

El **objeto de estudio** está orientado al proceso de desarrollo de software del tipo tutorial informativo.

Dentro del **campo de acción** se encuentra el desarrollo de un material didáctico sobre los artefactos electrodomésticos.

Como **ideas a defender** se puede plantear que si se diseña un software para proporcionar conocimientos básicos acerca de los tipos de artefactos electrodomésticos existentes, con sus componentes y funcionamiento, a través de la combinación de elementos multimedia, se contará con un medio de información para educar a la comunidad venezolana en el uso y cuidado de los equipos domésticos.

El **objetivo general** de este trabajo es realizar el análisis y diseño de la multimedia informativa Artefactos Electrodomésticos y sus Componentes Vol. 1.

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

- Diseñar un módulo que facilite la clasificación de los artefactos electrodomésticos que trabajan con calor (térmicos) y su importancia.
- Diseñar un módulo que facilite la identificación de los componentes y funcionamiento de estos equipos, así como la introducción al concepto básico de electricidad, al circuito eléctrico y a la aplicación del efecto Joule en el funcionamiento de los equipos térmicos.

Con vistas al cumplimiento de los objetivos se propone la realización de las siguientes **tareas**:

- Revisión bibliográfica sobre la fundamentación teórica referente a la multimedia educativa.
- Búsqueda de una adecuada arquitectura de sistema.
- Implementación del producto de forma modular, siguiendo la metodología RUP y apoyándose en la extensión OMMMA-L.
- Análisis de cómo se encuentran en la arena internacional las tecnologías que se utilizan para llevar a cabo aplicaciones como la que se pretende desarrollar.

Con el diseño de este trabajo se pretende finalmente obtener una multimedia que cumpla con las exigencias actuales de producción de aplicaciones multimedia en nuestra sociedad, acorde con los estándares internacionales de producción y catalogación y a las normas establecidas de calidad de diseño, además de proporcionar información, avivar el interés, orientar aprendizajes, proponer aprendizajes a partir de los errores, facilitar la evaluación y el control, posibilitar el trabajo individual y también en grupo.

La Estructuración del Contenido se basa en cinco capítulos. En el **capítulo I** se aborda sobre la Fundamentación del tema que incluye un estado del arte a nivel internacional, nacional y de la Universidad, análisis de otras soluciones existentes, descripción del objeto de estudio y el análisis del modelo de arquitectura de información utilizada.

El **capítulo II** aborda sobre las tendencias y tecnologías actuales a considerar: En este capítulo se explican las metodologías, los lenguajes usados, las herramientas utilizadas para el diseño de la aplicación y el estado del arte de la multimedia.

En el **capítulo III** Se describe la solución propuesta: Modelado del negocio del sistema de la plantilla a través de un modelo de dominio. Descripción de la información que se maneja, de la aplicación que se propone así como de todos los requisitos funcionales y no funcionales que esta debe cumplir.

En el **capítulo IV** Se construye la solución propuesta: Incluye la definición del modelo de análisis del sistema para cada realización de los casos de uso. Muestra el diagrama de presentación y el diagrama de componentes.

Y en el **capítulo V** Se estudia la factibilidad: Incluye todo el estudio de la factibilidad del producto a desarrollar así como una evaluación de la utilización del mismo.

Capítulo 1 Fundamentación teórica.

1.1 Introducción.

El contenido de este capítulo constituye la base teórica del presente trabajo. En él se describen los principales conceptos y lineamientos, como resultado de la investigación realizada para la concepción de la Multimedia Artefactos Electrodomésticos Vol. I.

Se brinda una vista global de los temas relacionados con cada uno de los elementos que integra los artefactos electrodomésticos y su funcionamiento de los más conocidos que se encuentran en el hogar, así como los principales conceptos asociados al dominio del problema, que son necesarios para entender el modelo de negocio y la propuesta de la solución. Además se realiza una comparación de las herramientas existentes y se determina cuáles van a ser las utilizadas en el sistema.

1.2 Estado del arte.

En el universo audio visual donde vive el hombre en las sociedades desarrolladas modernas, las técnicas multimedia se convierten cada día en un instrumento eficaz de comunicación y acceso a la información.

La implementación de las capacidades multimedia en las computadoras es sólo el último episodio de una larga serie de avances en el desarrollo de la humanidad, que reflejan el deseo innato del hombre de crear herramientas para expresarse creativamente, de comunicarse más poderosamente y liberar ideas: pinturas rupestres, textos, manuscritos, imprenta, radio y televisión.

La hipermedia surge como resultado de la fusión de dos tecnologías, el hipertexto y la multimedia. El hipertexto es la organización de una determinada información en diferentes nodos, conectados entre sí a través de enlaces. Los nodos pueden contener subelementos con entidad propia. Un hiperdocumento estaría formado por un conjunto de nodos conectados y relacionados temática y estructuralmente.

La tecnología multimedia es la que permite integrar diferentes medios (sonido, imágenes, secuencias...) en una misma presentación. La hipermedia, por tanto, es la tecnología que nos permite estructurar la información de una manera no-secuencial, a través de nodos interconectados por enlaces. La información presentada en estos nodos podrá integrar diferentes medios tales como: Texto, sonido, gráficos, video, etc. A partir de esta descripción se puede afirmar de forma sintética que una obra hipermedia es una obra de comunicación audiovisual interactiva, la cual no es sólo un producto informático cuya realización es totalmente tecnológica e ingeniería, sino también una labor creativa en la que se integran elementos estéticos y funcionales capaces de actuar sobre las emociones humanas de forma intensa, como pueden ser la literatura, el cine o la televisión, potenciada a su vez por las posibilidades interactivas y de multiplicidad de canales de comunicación que posee tanto para la cultura como en la educación de las personas.

Estas hipermedias y multimedia pretenden resolver el problema del procesamiento lineal de la información por el receptor, como ocurre en el libro de texto. Por el contrario, la información se puede construir desde diferentes trayectorias y alternativas, y con diferentes tipos de códigos.

Estas trayectorias pueden limitarse por el autor del programa, para evitar problemas de desorientación en el usuario [1], pudiendo soportar la autoría de documentos complejos y el procesamiento de ideas, especialmente en entornos de trabajo colaborativos. Para la aplicación de esta facilidad los alumnos necesitan la capacidad de anotar nodos de información y ser capaces de colaborar con otros estudiantes y con el profesor sobre unidades específicas de

información. Aún más importante que esto es que deben ser capaces de construir su propio sistema de conocimiento a través de la investigación, abstracción, readaptación y adición de conocimientos a una base de datos existente.

Estos sistemas proporcionan a los estudiantes un alto grado de control sobre la secuencia y contenido para ajustarlo a sus conocimientos y a su estilo de aprendizaje, permiten al estudiante determinar la secuencia en la que acceder a la información, añadir o modificar la información en orden a hacerla más personalmente significativa, o construir y estructurar su propia base de conocimiento. El nivel de control del usuario varía según las prestaciones del sistema y los objetivos del programa.

En los últimos años se ha producido una gran difusión de materiales multimediales e hipermediales, producidos en gran escala, mediante la forma de cursos básicamente de idiomas, educativos, de aprendizaje de programas de computadora y centrado fundamentalmente al aprendizaje del alumno a estos. Se ha observado que en la mayoría, estos cursos han sido realizados de acuerdo a algunos de los paradigmas descritos en este capítulo con herramientas de autor sea Toolbook, Macromedia Director, Macromedia Flash o algún lenguaje de programación como lo es el Visual Basic.

1.2.1 Ventajas de la hipermedia.

- Hipermedia ofrece un medio adecuado para representar información poco o nada estructurada, que no puede ajustarse a los rígidos esquemas de las bases de datos tradicionales.
- Sus ergonómicos interfaces de usuario, muy intuitivos pues imitan el funcionamiento de la memoria humana.
- Los usuarios pueden hacer crecer el hiperdocumento o anotarlos sin modificarlos. Facilita la división en módulos y la consistencia de la información.

- Constituyen un marco idóneo para la autoría en colaboración.
- Facilitan diferentes modos de acceso a la información de manera que el usuario pueda elegir en cada momento el que más se ajuste a sus necesidades.

1.2.2 Inconvenientes de la hipermedia.

- La desorientación: incapacidad del usuario para controlar la información en un inextricable espacio interconectado. Éste problema está íntimamente ligado al diseño del hiperdocumento y las soluciones pasan por el uso de metáforas en sustitución de los iconos, uso de índices, posibilidad de vuelta atrás, uso de mapas de situación y navegadores gráficos.
- La sobrecarga: se ha comprobado que comprender y utilizar las técnicas de recuperación de información en un hiperdocumento puede suponer un gran esfuerzo para el usuario. Las soluciones para minimizar éste problema pasa por el uso de un interfaz lo más intuitivo posible, huir de cualquier tipo de exceso como el empleo masivo e innecesario de elementos multimedia así como la generación sin sentido de enlaces.

1.3 Descripción del objeto de estudio.

1.3.1 Descripción General

Dentro del grupo de los materiales multimedia, que integran diversos elementos textuales (secuenciales e hipertextuales) y audiovisuales (gráficos, sonido, vídeo, animaciones...), están los *materiales multimedia educativos*, que son los materiales multimedia que se utilizan con una finalidad educativa.

Atendiendo a su estructura los materiales didácticos multimedia se pueden clasificar en programas tutoriales, de ejercitación, simuladores, bases de datos, constructores, programas

herramienta..., presentando diversas concepciones sobre el aprendizaje y permitiendo en algunos casos (programas abiertos, lenguajes de autor) la modificación de sus contenidos y la creación de nuevas actividades de aprendizaje por parte de los profesores y los estudiantes. Con más detalle, la clasificación es la siguiente:

- *Materiales formativos directivos*. En general siguen planteamientos conductistas. Proporcionan información, proponen preguntas y ejercicios a los alumnos y corrigen sus respuestas.
- *Programas de ejercitación*. Se limitan a proponer ejercicios autocorrectivos de refuerzo sin proporcionar explicaciones conceptuales previas. Su estructura puede ser: lineal (la secuencia en la que presentan las actividades es única o totalmente aleatoria), ramificada (la secuencia depende de los aciertos de los usuarios) o tipo entorno (proporciona a los alumnos herramientas de búsqueda y de proceso de la información para que construyan la respuesta a las preguntas del programa).
- *Programas tutoriales*. Presentan unos contenidos y proponen ejercicios autocorrectivos al respecto. Si utilizan técnicas de Inteligencia Artificial para personalizar la tutorización según las características de cada estudiante, se denominan *tutoriales expertos*.
- *Programas herramienta*. Proporcionan un entorno instrumental con el cual se facilita la realización de ciertos trabajos generales de tratamiento de la información: escribir, organizar, calcular, dibujar, transmitir, captar datos...
- *Lenguajes y sistemas de autor*. Facilitan la elaboración de programas tutoriales a los profesores que no disponen de grandes conocimientos informáticos.

Los materiales multimedia educativos, como los materiales didácticos en general, pueden realizar múltiples funciones en los procesos de enseñanza y aprendizaje. Las principales

funciones que pueden realizar los recursos educativos multimedia son las siguientes: informativa, instructiva o entrenadora, motivadora, evaluadora, entorno para la exploración y la experimentación, expresivo-comunicativa, metalingüística, lúdica, proveedora de recursos para procesar datos, innovadora, apoyo a la orientación escolar y profesional, apoyo a la organización y gestión de centros.

1.3.2 Identificación de la audiencia.

Uno de los aspectos más importantes es la correcta identificación del usuario final del sistema que va dirigida a:

Nivel Educativo: Escuela Básica (tercera etapa de educación básica)

Edad: Mayor de quince años

Sexo: masculino y femenino

Conocimientos previos sobre el tema: No se requieren conocimientos previos.

Dentro de las posibilidades interactivas se encuentran:

Medio de transmisión de información, este software será diseñado para transmitir conocimientos sencillos y básicos, acerca de cómo distinguir los tipos de artefactos electrodomésticos existentes, conocer sus componentes y funciones, a través de la combinación de elementos multimedia como son: imágenes, animaciones, textos, música, efectos de sonido, locuciones, entre otros.

Medio para desarrollar estrategias, los usuarios, a lo largo del software, encontrarán ambientes con íconos que al hacer clic sobre ellos, podrán navegar de manera sistemática por él, donde obtendrán informaciones e instrucciones referidas a los temas que están en pantalla.

Medio para la aplicación de conocimientos en situaciones nuevas, en este Vol. 1 los usuarios podrán obtener conocimientos básicos acerca de los artefactos electrodomésticos, su funcionamiento y componentes, lo cual es fundamental que el usuario conozca antes de realizar el mantenimiento y reparación de los mismos, tema que se explica en el Vol. 2.

Medio para desarrollar el razonamiento, al finalizar cada tema el usuario podrá realizar las actividades interactivas, donde podrá autoevaluar el conocimiento adquirido, a través actividades interactivas, como son: identificación, arrastrar y soltar, clasificación y autoevaluaciones. Todas estas actividades incluyen: imágenes, sonidos y mensajes de retroalimentación, que le indicarán si domina o no los contenidos.

1.5 Análisis del modelo de arquitectura de información utilizada.

1.5.1 Principios y normas de diseño.

El Modelo Vista Controlador (MVC) es un patrón de arquitectura de software que separa los datos de una aplicación, la interfaz de usuario y la lógica de control en tres componentes distintos. El patrón MVC se ve frecuentemente en aplicaciones web y en aplicaciones multimedia, por ejemplo, la vista es la página HTML y el código que provee de datos dinámicos a la página, el controlador es el Sistema de Gestión de Base de Datos y el modelo es el modelo de datos.

- **Modelo:** Esta es la representación específica de la información con la cual el sistema opera. La lógica de datos asegura la integridad de estos y permite derivar nuevos datos; por ejemplo, no permitiendo comprar un número de unidades negativo, calculando si hoy es el cumpleaños del usuario o los totales, impuestos o portes en un carrito de la compra.
- **Vista:** Este presenta el modelo en un formato adecuado para interactuar, usualmente la interfaz de usuario.
- **Controlador:** Este responde a eventos, usualmente acciones del usuario e invoca cambios en el modelo y probablemente en la vista.

Muchos sistemas informáticos utilizan un Sistema de Gestión de Base de Datos para gestionar los datos. En MVC corresponde al controlador.

Aunque se pueden encontrar diferentes implementaciones de MVC, el flujo que sigue el control generalmente es el siguiente:

1. El usuario interactúa con la interfaz de usuario de alguna forma (por ejemplo, el usuario pulsa un botón, enlace)
2. El controlador recibe (por parte de los objetos de la interfaz-vista) la notificación de la acción solicitada por el usuario. El controlador gestiona el evento que llega, frecuentemente a través de un gestor de eventos (handler) o callback.
3. El controlador accede al modelo, actualizándolo, posiblemente modificándolo de forma adecuada a la acción solicitada por el usuario (por ejemplo, el controlador actualiza el carro de la compra del usuario). Los controladores complejos están a menudo estructurados usando un patrón de comando que encapsula las acciones y simplifica su extensión.
4. El controlador delega a los objetos de la vista la tarea de desplegar la interfaz de usuario. La vista obtiene sus datos del modelo para generar la interfaz apropiada para el usuario donde se refleja los cambios en el modelo (por ejemplo, produce un listado del contenido del carro de la compra). El modelo no debe tener conocimiento directo sobre la vista. Sin embargo, el patrón de observador puede ser utilizado para proveer cierta indirección entre el modelo y la vista, permitiendo al modelo notificar a los interesados de cualquier cambio. Un objeto vista puede registrarse con el modelo y esperar a los cambios, pero aun así el modelo en sí mismo sigue sin saber nada de la vista. El controlador no pasa objetos de dominio (el modelo) a la vista aunque puede dar la orden a la vista para que se actualice.
Nota: En algunas implementaciones la vista no tiene acceso directo al modelo, dejando que el controlador envíe los datos del modelo a la vista.
5. La interfaz de usuario espera nuevas interacciones del usuario, comenzando el ciclo nuevamente.

1.5.2 Estándares de la Interfaz de la aplicación.

La Interfaz de Usuario, en adelante IU, de un programa es un conjunto de elementos hardware y software de una computadora que presentan información al usuario y le permiten interactuar con la información y con el computadora. También se puede considerar parte de la IU la documentación (manuales, ayuda, referencia, tutoriales) que acompaña al hardware y al software. Si la IU está bien diseñada, el usuario encontrará la respuesta que espera a su acción. Si no es así puede ser frustrante su operación, ya que el usuario habitualmente tiende a culparse a sí mismo por no saber usar el objeto.

Los programas son usados por usuarios con distintos niveles de conocimientos, desde principiantes hasta expertos. Es por ello que no existe una interfaz válida para todos los usuarios y todas las tareas. Debe permitirse libertad al usuario para que elija el modo de interacción que más se adecue a sus objetivos en cada momento. La mayoría de los programas y sistemas operativos ofrecen varias formas de interacción al usuario.

Existen tres puntos de vista distintos en una IU: el del usuario, el del programador y el del diseñador (analogía de la construcción de una casa). Cada uno tiene un modelo mental propio de la interfaz, que contiene los conceptos y expectativas acerca de la misma, desarrollados a través de su experiencia.

El modelo permite explicar o predecir comportamientos del sistema y tomar las decisiones adecuadas para modificar el mismo. Los modelos subyacen en la interacción con las computadoras, de ahí su importancia.

Modelo del usuario: El usuario tiene su visión personal del sistema, y espera que éste se comporte de una cierta forma. Se puede conocer el modelo del usuario estudiándolo, ya sea realizando tests de usabilidad, entrevistas, o a través de una realimentación. Una interfaz debe facilitar el proceso de crear un modelo mental efectivo.

Para ello son de gran utilidad las metáforas, que asocian un dominio nuevo a uno ya conocido por el usuario. Un ejemplo típico es la metáfora del escritorio, común a la mayoría de las

interfaces gráficas actuales. Modelo del diseñador: El diseñador mezcla las necesidades, ideas, deseos del usuario y los materiales de que dispone el programador para diseñar un producto de software. Es un intermediario entre ambos.

El modelo del diseñador: describe los objetos que utiliza el usuario, su presentación al mismo y las técnicas de interacción para su manipulación. Consta de tres partes: presentación, interacción y relaciones entre los objetos.

La presentación es lo que primero capta la atención del usuario, pero más tarde pasa a un segundo plano, y adquiere más importancia la interacción con el producto para poder satisfacer sus expectativas. La presentación no es lo más relevante y un abuso en la misma (por ejemplo, en el color) puede ser contraproducente, distrayendo al usuario.

La segunda parte del modelo define las técnicas de interacción del usuario, a través de diversos dispositivos.

La tercera es la más importante, y es donde el diseñador determina la metáfora adecuada que encaja con el modelo mental del usuario. El modelo debe comenzar por esta parte e ir hacia arriba. Una vez definida la metáfora y los objetos del interfaz, los aspectos visuales saldrán de una manera lógica y fácil.

Estos modelos deben estar claros para los participantes en el desarrollo de un producto, de forma que se consiga una interfaz atractiva y a la vez efectiva para el trabajo con el programa.

Una interfaz no es simplemente una cara bonita; esto puede impresionar a primera vista pero decepcionar a la larga. Lo importante es que el programa se adapte bien al modelo del usuario, cosa que se puede comprobar utilizando el programa más allá de la primera impresión.

Modelo del programador: Es el más fácil de visualizar, al poderse especificar formalmente. Está constituido por los objetos que manipula el programador, distintos de los que trata el usuario (ejemplo: el programador llama base de datos a lo que el usuario podría llamar agenda). Estos objetos deben esconderse del usuario. Los conocimientos del programador incluyen la plataforma de desarrollo, el sistema operativo, las herramientas de desarrollo y especificaciones. Sin

embargo, esto no significa necesariamente que tenga la habilidad de proporcionar al usuario los modelos y metáforas más adecuadas. Muchos no consideran el modelo del usuario del programa, y sí sus propias expectativas acerca de cómo trabajar con la computadora.

Existen principios relevantes para el diseño e implementación de IU, ya sea para las IU gráficas, como para la Web.

Anticipación.

Las aplicaciones deberían intentar anticiparse a las necesidades del usuario y no esperar a que el usuario tenga que buscar la información, recopilarla o invocar las herramientas que va a utilizar.

Autonomía.

La computadora, la IU y el entorno de trabajo deben estar a disposición del usuario. Se debe dar al usuario el ambiente flexible para que pueda aprender rápidamente a usar la aplicación. Sin embargo, está comprobado que el entorno de trabajo debe tener ciertas cotas, es decir, ser explorable pero no azaroso.

Es importante utilizar mecanismos indicadores de estado del sistema que mantengan a los usuarios alertas e informados. No puede existir autonomía en ausencia de control, y el control no puede ser ejercido sin información suficiente. Además, se debe mantener información del estado del sistema en ubicaciones fáciles de visualizar.

Percepción del Color.

Aunque se utilicen convenciones de color en la IU, se deberían usar otros mecanismos secundarios para proveer la información a aquellos usuarios con problemas en la visualización de colores.

Valores por Defecto.

No se debe utilizar la palabra "Defecto" en una aplicación o servicio. Puede ser reemplazada por "Estándar" o "Definida por el Usuario", "Restaurar Valores Iniciales" o algún otro término

especifico que describa lo que está sucediendo. Los valores por defecto deberían ser opciones inteligentes y sensatas. Además, los mismos tienen que ser fáciles de modificar.

Consistencia.

Para lograr una mayor consistencia en la IU se requiere profundizar en diferentes aspectos que están catalogados en niveles. Se realiza un ordenamiento de mayor a menor consistencia:

1. Interpretación del comportamiento del usuario: la IU debe comprender el significado que le atribuye un usuario a cada requerimiento. Ejemplo: mantener el significado de los comandos abreviados (shortcut-keys) definidos por el usuario.
2. Estructuras invisibles: se requiere una definición clara de las mismas, ya que sino el usuario nunca podría llegar a descubrir su uso. Ejemplo: la ampliación de ventanas mediante la extensión de sus bordes.
3. Pequeñas estructuras visibles: se puede establecer un conjunto de objetos visibles capaces de ser controlados por el usuario, que permitan ahorrar tiempo en la ejecución de tareas específicas. Ejemplo: ícono y/o botón para impresión.
4. Una sola aplicación o servicio: la IU permite visualizar a la aplicación o servicio utilizado como un componente único. Ejemplo: La IU despliega un único menú, pudiendo además acceder al mismo mediante comandos abreviados.
5. Un conjunto de aplicaciones o servicios: la IU visualiza a la aplicación o servicio utilizado como un conjunto de componentes. Ejemplo: La IU se presenta como un conjunto de barras de comandos desplegadas en diferentes lugares de la pantalla, pudiendo ser desactivadas en forma independiente.
6. Consistencia del ambiente: la IU se mantiene en concordancia con el ambiente de trabajo. Ejemplo: La IU utiliza objetos de control como menús, botones de comandos de manera análoga a otras IU que se usen en el ambiente de trabajo.
7. Consistencia de la plataforma: La IU es concordante con la plataforma. Ejemplo: La IU tiene un esquema basado en ventanas, el cual es acorde al manejo del sistema operativo Windows.

La inconsistencia en el comportamiento de componentes de la IU debe ser fácil de visualizar. Se debe evitar la uniformidad en los componentes de la IU. Los objetos deben ser consistentes con su comportamiento. Si dos objetos actúan en forma diferente, deben lucir diferentes. La única forma de verificar si la IU satisface las expectativas del usuario es mediante testeos.

Eficiencia del Usuario.

Se debe considerar la productividad del usuario antes que la productividad de la máquina. Si el usuario debe esperar la respuesta del sistema por un período prolongado, estas pérdidas de tiempo se pueden convertir en pérdidas económicas para la organización. Los mensajes de ayuda deben ser sencillos y proveer respuestas a los problemas. Los menús y etiquetas de botones deberían tener las palabras claves del proceso.

Interfaces Explorables.

Siempre que sea posible se debe permitir que el usuario pueda salir ágilmente de la IU, dejando una marca del estado de avance de su trabajo, para que pueda continuarlo en otra oportunidad. Para aquellos usuarios que sean noveles en el uso de la aplicación, se deberá proveer de guías para realizar tareas que no sean habituales. Es conveniente que el usuario pueda incorporar elementos visuales estables que permitan, no solamente un desplazamiento rápido a ciertos puntos del trabajo que esté realizando, sino también un sentido de "casa" o punto de partida. La IU debe poder realizar la inversa de cualquier acción que pueda llegar a ser de riesgo, de esta forma se apoya al usuario a explorar el sistema sin temores. Siempre se debe contar con un comando "Deshacer". Este suprimirá la necesidad de tener que contar con diálogos de confirmación para cada acción que realice en sistema. El usuario debe sentirse seguro de poder salir del sistema cuando lo desee. Es por ello que la IU debe tener un objeto fácil de accionar con el cual poder finalizar la aplicación.

Objetos de Interfaz Humana.

Los objetos de interfaz humana no son necesariamente los objetos que se encuentran en los

sistemas orientados a objetos. Estos pueden ser vistos, escuchados, tocados o percibidos de alguna forma. Además, estos objetos deberían ser entendibles, consistentes y estables.

Uso de Metáforas.

Las buenas metáforas crean figuras mentales fáciles de recordar. La IU puede contener objetos asociados al modelo conceptual en forma visual, con sonido u otra característica perceptible por el usuario que ayude a simplificar el uso del sistema.

Curva de Aprendizaje.

El aprendizaje de un producto y su usabilidad no son mutuamente excluyentes. El ideal es que la curva de aprendizaje sea nula, y que el usuario principiante pueda alcanzar el dominio total de la aplicación sin esfuerzo.

Reducción de Latencia.

Siempre que sea posible, el uso de tramas (multi-threading) permite colocar la latencia en segundo plano (background). Las técnicas de trabajo multitarea posibilitan el trabajo ininterrumpido del usuario, realizando las tareas de transmisión y computación de datos en segundo plano.

Protección del Trabajo.

Se debe poder asegurar que el usuario nunca pierda su trabajo, ya sea por error de su parte, problemas de transmisión de datos, de energía, o alguna otra razón inevitable.

Auditoria del Sistema.

La mayoría de los navegadores de Internet (browsers), no mantienen información acerca de la situación del usuario en el entorno, pero para cualquier aplicación es conveniente conocer un conjunto de características tales como: hora de acceso al sistema, ubicación del usuario en el sistema y lugares a los que ha accedido, entre otros. Además, el usuario debería poder salir del sistema y al volver a ingresar continuar trabajando en lugar dónde había dejado.

Legibilidad.

Para que la IU favorezca la usabilidad del sistema de software, la información que se exhiba en ella debe ser fácil de ubicar y leer. Para lograr obtener este resultado se deben tener en cuenta alguna como: el texto que aparezca en la IU debería tener un alto contraste, se debe utilizar combinaciones de colores como el texto en negro sobre fondo blanco o amarillo suave. El tamaño de las fuentes tiene que ser lo suficientemente grande como para poder ser leído en monitores estándar. Es importante hacer clara la presentación visual (colocación/agrupación de objetos, evitar la presentación de excesiva información).

Interfaces Visibles.

El uso de Internet, ha favorecido la implementación de interfaces invisibles. Esto significa que el usuario siempre ve una página específica, pero nunca puede conocer la totalidad del espacio de páginas de Internet. La navegación en las aplicaciones debe ser reducida a la mínima expresión. El usuario debe sentir que se mantiene en un único lugar y que el que va variando es su trabajo. Esto no solamente elimina la necesidad de mantener mapas u otras ayudas de navegación, sino que además brindan al usuario una sensación de autonomía.

Niveles de Prototipado.

Se puede hacer una clasificación de los principales tipos de prototipos, variando su grado de complejidad, de acuerdo a las características que consideren y a su operabilidad para realizar simulaciones.

- Prototipos Estáticos: son aquellos que no permiten la alteración de sus componentes, pero sirven para identificar y resolver problemas de diseño. En esta categoría se incluyen las presentaciones sobre reproductores, papel u otro medio de visualización.
- Prototipos Dinámicos: permiten la evaluación de un modelo del sistema sobre una estación de trabajo o una terminal. Estos prototipos involucran aspectos de diseño mas detallados que los prototipos estáticos, incluyendo la validación del diseño del sistema en términos de requerimientos no funcionales, por ejemplo de performance.

- Prototipos Robustos: deben ser relativamente completos en la simulación de las características dinámicas de la interfaz (presentación de mensajes de error, entrada y edición de datos, etc.). Esta categoría puede ser utilizada para validar los objetivos de diseño.

El nivel de sofisticación del prototipo debería incrementarse a lo largo del proceso de diseño de interfaces de usuario. La información recolectada durante las tareas de análisis del sistema y la especificación de los requisitos del usuario constituyen los datos clave para el proceso de prototipación.

Las heurísticas ayudan a poder analizar las IU y localizar problemas que afecten la utilización de las mismas.

Algunas pautas para evaluar una IU son:

- Visibilidad del estado del sistema
- Semejanza del sistema al mundo real
- Control y libertad por parte del usuario
- Consistencia y estandarización
- Prevención de Errores
- Reconocimiento de acciones y opciones
- Flexibilidad y eficiencia en el uso
- Estética y diseño minimalista
- Reconocimiento de errores, diagnóstico y recuperación
- Ayuda y documentación

Para establecer medidas que indiquen la severidad de los problemas en el uso de las interfaces, se deben conocer los factores que determinan el grado de un problema:

- La frecuencia de ocurrencia.
- El impacto que causa la ocurrencia del problema.

- La persistencia del problema.
- El impacto en el mercado.

Medidas de severidad de un problema en la IU:

0: No puede llegar a considerarse un problema.

1: Es un problema "cosmético" que no necesita ser corregido a menos que se disponga tiempo extra en el proyecto.

2: Es un problema menor y su corrección puede tener baja prioridad.

3: Es un problema mayor y su corrección debería tener alta prioridad.

4: Es una catástrofe para la utilización de la aplicación y es imperativo corregir el error.

Para la evaluación de los problemas en las IU es conveniente que contar con más de un evaluador; de esta forma los resultados son más confiables. [6]

1.5.3 Estándares de codificación.

Los estándares de codificación son reglas específicas a una lengua que reducen perceptiblemente el riesgo de que los desarrolladores introduzcan errores. Los estándares de codificación no destapan problemas existentes, evitan más bien que los errores ocurran. Los bugs frecuentes en programas pueden ser detectados mucho anterior o pueden incluso ser evitados totalmente. Durante el desarrollo, estándares de codificación ayudan a los ingenieros a producir un código de alta calidad y a entender y a utilizar el código de sus colegas. Pero también realzan considerablemente la capacidad de mantenimiento y rehúso a largo plazo del producto final. Tal práctica del control de bugs en el proceso del desarrollo mejora la calidad mientras que reduce el tiempo de desarrollo, el coste, y el esfuerzo.

¿Por qué adoptar los estándares de codificación?

- Reducir la probabilidad de introducir errores.
- Precisar algunos errores ocultos o inesperados.

- Evitar que los bugs se desplieguen en su software.
- Hacer el código más uniforme y más fácil a leer.
- Hacer el software más fácil a mantener.
- Aumentar la robustez y la confiabilidad.
- Conformarse con los estándares y modelos del cliente.
- Revisar el código entregado por sus proveedores.
- Entrenar a los ingenieros jóvenes al buen desarrollo del software.

1.6 Conclusiones.

Con la conclusión de este capítulo se han demostrado los aspectos necesarios para la realización de este trabajo, ha servido para dar una explicación a la situación problemática y definiciones que ayudan a comprender el problema que se presenta, llegando al objetivo central por el que se rige este trabajo. Este capítulo fundamenta todo lo necesario para darle una solución eficaz al problema a resolver.

Capítulo 2 Tendencias y tecnologías.

2.1 Introducción.

En la actualidad con el desarrollo de las nuevas tecnologías de la información y las comunicaciones se hace posible un intenso intercambio de ideas, experiencias y conocimientos entre millones de seres humanos. La educación de la población mundial es una compleja y costosísima tarea para las instituciones de este sector. La educación a distancia constituye una alternativa para enfrentar este enorme problema.

Los estudiantes deberán moverse en un entorno rico en información, ser capaces de analizar y tomar decisiones, y dominar nuevos ámbitos del conocimiento en una sociedad cada vez más tecnológica. Deberán convertirse en estudiantes de por vida, colaborando con otros individuos para realizar tareas complejas y utilizando de modo efectivo los diferentes sistemas de representación y comunicación de conocimiento. El pasaje de un aprendizaje mayormente centrado en el docente hacia uno centrado en el alumno, puede crear un entorno de aprendizaje más interactivo y más motivador tanto para los alumnos como para los propios docentes, el profesor dejará de ser únicamente el transmisor de conocimiento para convertirse en un facilitador y orientador del mismo y en un participante del proceso de aprendizaje junto con el estudiante. Este nuevo rol no disminuye la importancia del docente, pero requiere de nuevos conocimientos y habilidades. Los alumnos serán más responsables de su propio aprendizaje en la medida en que busquen, encuentren, sinteticen y compartan su conocimiento con otros compañeros.

Las TICs constituyen una herramienta poderosa para apoyar este cambio y para facilitar el surgimiento de nuevos roles en docentes y alumnos. Aunque este nuevo entorno de aprendizaje puede crearse sin hacer uso de la tecnología, es claro que las TICs constituyen una herramienta decisiva para ayudar a los estudiantes a acceder a vastos recursos de conocimiento, a colaborar con otros compañeros, consultar a expertos, compartir conocimiento y resolver problemas

complejos utilizando herramientas cognitivas. Estas también ofrecen a los alumnos novedosas herramientas para representar su conocimiento por medio de texto, imágenes, gráficos y video, siendo esto el caso de las hipermedias de aprendizaje.

2.2 Tendencias y Tecnologías actuales a considerar.

2.2.1 Multimedia, principales aspectos.

Desde la década de los años 70 se comenzaron a dar los primeros pasos para la inserción de la computación en los procesos docentes. Con el de cursar de los años, se ha llegado a la conclusión de que el empleo de la computación en los procesos docentes de los distintos niveles de enseñanza, por si solo no resuelven todos los problemas.

Las tecnologías informáticas de última generación, pueden contribuir enormemente a solucionar un conjunto de problemas docentes que hoy en día existen, tales como, la escasez de bibliografía actualizada, la motivación de los estudiantes por una rama determinada del conocimiento, le ofrece al estudiante materiales didácticos de más calidad, así como la posibilidad de realizar cálculos con más rapidez y precisión, otro problema que nos pudiera ayudar a resolver la informática es la comunicación alumno profesor y entre los propios alumnos, puede contribuir a que tanto profesores y alumnos se mantengan actualizados, entre otros problemas. Ahora esto solo es posible si se utiliza adecuadamente, pues de no ser así puede crear el efecto inverso.

2.2.2 Principales conceptos de multimedia.

Si se habla de computación e informática no se puede dejar de hablar de software puesto que sin este la máquina no serviría de nada. Algunos desde el punto de vista de su función pueden tener la clasificación de software educativo.

Los software educativos los definimos como “Software diseñado y producido para la docencia o cualquier otro software que, aunque no haya sido diseñado con fines puramente educativos, se este empleando con el propósito de enseñar y aprender”.

El software educativo presenta varias clasificaciones pero en este capítulo solo abordaremos su clasificación según su función didáctica.

En este sentido abundan las clasificaciones, pero todas de una forma u otra consideran que existen:

- Tutoriales
- Entrenadores
- Simuladores
- Evaluadores

Los tutoriales consideramos que son los software educativos que cumplen con las funciones didácticas de motivar el estudiante, introducir y desarrollar la nueva materia.

En los tutoriales, la materia se le presenta a los estudiantes combinando imágenes, sonidos, videos y controlando el nivel de conocimientos alcanzado a través de preguntas, ejercicios. etc.

Algunos expertos plantean que tutorial es todo aquel software que suple al profesor pues tiene la función de presentar un conocimiento nuevo al estudiante. En esto no coincidimos plenamente, pues consideramos que por un lado es imposible sustituir al profesor, debido a que su función no es solamente presentar una información al estudiante para que este la convierta en conocimiento, sino que esta matizada de un componente afectivo que, “hasta ahora”, no hay máquina que capaz de suplirlo. Y por otro lado no solamente los software tienen la “capacidad” de presentar una información nueva al estudiante, los libros, las revistas, la televisión y otros medios, también pueden cumplir esta función.

Algunos autores plantean la existencia de tutoriales inteligentes, estos software realmente, parten de los fundamentos de la inteligencia artificial, que la definimos como la ciencia

cibernética, que estudia los sistemas que simulan los procesos lógicos del pensamiento humano, esta ciencia tiene como bases la lógica matemática, la psicología cognitiva, la gramática, y las ciencias de la computación.

Con respecto a los entrenadores se puede plantear que su función didáctica es la ejercitar lo aprendido por el alumno, también conocido como repasador. Las simulaciones son programas que intentan dar una representación lo más realista posible de un dominio complejo. Este tipo de software educativo intenta explicar un fenómeno mediante una representación controlada del mismo en la computadora con apariencia lo más cercana posible al fenómeno real. Los programas de simulación, como su nombre lo indica, reproducen en forma simplificada y a través de un modelo una situación o fenómeno (físico, químico, matemático, social), permitiéndole al estudiante explorar y actuar sobre dichos eventos. Los evaluadores tienen la función de, como bien su nombre lo indica evaluar al estudiante. Esto lo puede hacer de diferentes maneras, mediante distintos tipos de preguntas, de enlace, de verdadero o falso, de selección, de completamiento de frases o párrafos, etc. En el caso del software dirigido a los alumnos primarios esta evaluación se puede realizar mediante un juego.

La tendencia actual, en la producción de software educativos, en Cuba, no esta orientada ha hacer software puros tutoriales, simuladores o evaluadores, sino a producir software que lo integren todo, es decir que presente el nuevo contenido, que lo ejercite, lo evalúe y si tiene que simular o modelar, también lo haga. Nuestra aplicación tampoco fue una excepción, esta desarrollada principalmente como tutorial.

2.2.3 Tendencias de las aplicaciones multimedia.

El término multimedia se refiere a una integración o agrupación de diferentes medios audiovisuales. Pero la acepción actual más habitual del mismo es la que hace referencia a aquellos programas que se desarrollan a través del ordenador, de tal modo que todo el sistema multimedia se apoya en un solo soporte. La multimedia se convierte así en un entorno de

aprendizaje que combina las posibilidades educativas que ofrecen diferentes medios de comunicación interconectados y controlados a través de un ordenador. Se comienza a hablar entonces de la informática multimedia como un intento de combinar la capacidad autoexplicativa de los medios audiovisuales con el texto y fotografías, para crear un nuevo medio de comunicación único en la pantalla del ordenador.

Estamos pasando de una era industrial a una era donde la supremacía de individuos, organizaciones y naciones pasa por el manejo inteligente de la información. Por lo que se refiere a las tecnologías de los sistemas de información, entendiendo como tales las tecnologías de ordenadores, telecomunicaciones y automatización de oficinas, cabe decir que 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 para introducirnos en el campo de la comunicación audiovisual, de la transmisión de sensaciones y de innumerables novedades más. En cuanto a las tecnologías de la comunicación, se ha pasado de la tradicional división entre diferentes medios que compiten por un espacio educativo a un nudo de conexión único. Ahora el alumno no necesita desplazarse físicamente para consultar un texto, puesto que a través de su ordenador podrá acceder a cualquier clase de información, sin importar el lugar en que esté.

La característica principal de los sistemas multimedia es su gran flexibilidad así como la alta interactividad que poseen, pues permiten un aprendizaje autoguiado y autoiniciado, en el cual cada persona va construyendo su conocimiento, bien sea de manera individual o colectiva.

Entre los múltiples sistemas multimedia, destaca el hipermedia, basado en hipertexto. Debido a que el hipertexto no es un sistema cerrado permite que el formador sea el que decida en cada momento a qué tipo de información desea acceder. El alumno podrá construir sus propios caminos de lectura saltando aquellos que no considere de su interés. El hipertexto rompe la idea tradicional de que el formador es el depositario de toda la información que ha de recibir el alumno, el cual tendría como misión tratar de aprender lo más fielmente posible todo lo que aquél le transmitiera.

Como características principales y distintivas de la multimedia se encuentran en:

- La integración o mezcla de al menos tres de los diversos datos o información manejados por la computadora: texto, imagen, sonido, voz y video.
- La digitalización de esos diversos datos o tipos de información.
- La interactividad que propicia la relación del usuario con el programa y la interacción con la máquina, así como la posibilidad de colaboración o de trabajo en equipo.

2.2.4 Herramienta seleccionada: Macromedia Flash.

En este caso se seleccionó el Flash ya que es en la actualidad la herramienta más extendida para la producción de sistemas multimedia en el mundo profesional. Sus diferentes versiones, la facilidad de manejo y la inclusión de gran número de herramientas de producción son las bases fundamentales con las que cuenta este programa. Sus principales ventajas sobre los demás programas de autoría son:

1. *La programación con un lenguaje de script*, la cual presenta un nivel de programación poderoso, incluso, mas poderoso que el de programación visual por iconos, ya que amplia el campo de acción del paquete.
2. *Macromedia Flash 8 presenta niveles de integración nunca antes vistos con la familia de aplicaciones creadas por Macromedia*. Ahora, Director y la amplia gama de herramientas de desarrollo del programa Macromedia Director MX 2004 pueden combinar las fuerzas de este poderoso producto, para así construir presentaciones interactivas altamente efectivas, aplicaciones, juegos y medios de instrucción. También Flash 8 nos permite añadir hipervínculos a plataformas Web, contenido para aplicaciones tridimensionales y aplicaciones simples, tanto en Windows como en Macintosh.
3. *Sus herramientas están basadas en el tiempo*. Cada una de ellas usa su propia interfaz de usuario para manejar los eventos en el tiempo; muchas utilizan una línea de tiempo para darle

secuencia a los eventos que suceden durante una presentación multimedia y con frecuencia despliegan niveles de elementos de multimedia o eventos a lo largo de una escala con incrementos altamente precisos.

2.2.5 Información básica de Macromedia Flash 8.

Action Script es el lenguaje propio de Flash, herramienta específica diseñada para la creación de contenidos multimedia. Como lenguaje, Action Script ofrece diferentes características de alto nivel:

- * Gramática: comandos en idioma inglés, muy parecidos a los lenguajes BASIC.
- * Sintaxis: construcción sencilla de comandos, se escriben casi como el inglés.
- * Funciones: conjunto de comandos que realizan tareas complejas.
- * Programación estructurada.
- * Orientación a objetos.
- * Flash ofrece un entorno de programación “amigable”, permitiendo organizar y depurar el código fácilmente.

Flash es una herramienta para la creación de contenidos multimedia, está planteado entorno a la metáfora del teatro (o cine). En una obra de teatro se encuentra:

- * La obra
- * El reparto
- * El guión
- * El escenario

En Flash hay: Película (Película), Biblioteca (Reparto), Línea de tiempo (Guión), Escenario (Escenario), [figura 1, anexo 1].

Película.

Toda “obra” creada en Flash se llama Película. Una película puede tener diferentes formas de distribución:

*”Work in progress”: cuando aún se esta modificando la película. Se pueden modificar todos sus elementos. Se almacena como un fichero x fla, y solamente puede abrirse con Flash.

* Distribución off-line: Flash originalmente fue diseñado para la creación de CD-ROMs. Para distribuir una película terminada en CD-ROM, se “cierra” convirtiéndola en un fichero x.exe.

* Para los lenguajes compilados: el fichero que contiene el código es diferente al fichero ejecutable. Compilar implica convertir el código (texto) en algo ejecutable.

* Distribución on-line: Actualmente, Flash puede generar contenido para Internet. Cualquier película de Flash puede transformarse a formato Shockwave.

Cuando se está creando una película, se puede manipular, modificar, añadir o eliminar elementos. Las propiedades de una película pueden modificarse utilizando la herramienta “Propiedades”, [figura 2, anexo 1].

Principales tipos de miembros del reparto.

* **Bitmaps:** Gráficos o imágenes. Se pueden importar los formatos BMP, GIF, JPEG, LRG, PhotoShop, MacPaint, PNG, PICT, Targa, TIFF.

- * **Texto:** En Flash existen campos que contienen caracteres con formato. Se pueden crear en Flash mediante la ventana de herramientas o importar ficheros creados con procesadores de texto.

- * **Optimización de fuentes:** Incorpora también opciones de legibilidad para fuentes pequeñas, haciendo la lectura de nuestros 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.

- * **Sonidos:** Se pueden importar diferentes formatos de ficheros de sonido: MP3, AIFF, WAV y Shockwave.

- * **Animación:** Flash 8 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 nuestras interpolaciones.

- * **Vectores:** figuras complejas formadas por líneas capaces de doblarse o curvarse. Se crean en Flash utilizando en la ventana Herramienta la Herramienta de Subselección.

- * **Director:** Director permite incluir películas de Flash dentro del reparto y ser modificadas con otras herramientas que este posee.

- * **Vídeo:** Para facilitar el resultado con formatos de vídeo, Flash 8 incluye un códec independiente de calidad superior capaz de competir con los mejores códecs de vídeo actuales con un tamaño de archivo mucho más pequeño. Además de una gran posibilidad de revestimientos para los controles de éste en nuestra película.

- * **3D:** Modelos tridimensionales creados con algún programa 3D.

* **Scripts:** En estos miembros se almacena el código escrito en Action Scripts de la película.

* **Java Script:** Debemos destacar además, que Flash puede trabajar conjuntamente con Java Script y ejecutar códigos y funciones sin problemas.

* **Diseño:** Permite el uso de efectos visuales que nos facilitarán la creación de animaciones, presentaciones y formularios más atractivos y profesionales, así mismo, pone a nuestra disposición mecanismos para hacer este trabajo más cómodo y rápido, tales como la existencia de filtros y modos de mezcla.

2.3 Metodologías de desarrollo de software.

Uno de los temas más comunes en el mundo de la informática hoy en día es el de las metodologías de desarrollo de software: cómo trabajar eficientemente evitando las catástrofes que conllevan al fracaso de un gran porcentaje de proyectos. Una metodología tiene como objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo, por medio de una mayor transparencia y control sobre el proceso; “producir lo esperado en el tiempo esperado y con el coste esperado”.

Durante los últimos años se han desarrollado dos corrientes en lo referente a las metodologías de desarrollo de software, las llamadas “pesadas” y las llamadas “ligeras o ágiles”. Las primeras se basan en la idea de conseguir el objetivo común por medio de orden y documentación, mientras que las segundas tratan de lograrlo por medio de la comunicación directa e inmediata entre aquello que intervienen en el proceso. Analizaremos a continuación tres de las más conocidas, sus características, ventajas y desventajas.

2.3.1 MultiMet como proceso de producción.

Como propuesta para sentar los pilares del proceso, se estudió Multimet, una metodología de diseño nacional que describe etapas generales de la organización de un proyecto informático de multimedia. Su objetivo es que cada especialista componente 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, como las necesidades 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. Se especifican los temas que serán tratados, su orden de aparición y teniendo en cuenta el nivel de detalle individual, la forma en que será estructurado pautando cada elemento de media a utilizarse y las restricciones de diseño. La etapa de especificación de contenidos recopila toda la información referente a los objetos media a utilizar, mostrando un diagrama de flujo que tipifica la composición y navegación a través de módulos de pantallas; elabora el guión de contenidos donde describe cada media incorporada en una pantalla y la descripción de los eventos de interacción del usuario o propios del sistema a un nivel muy general. No modela la arquitectura del producto, sino la idea de su funcionamiento.

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 señalamos 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. Cabe reflexionar qué sucedería si luego de varios meses de trabajo no aparece una herramienta que se ajuste al diseño propuesto, o resulta más factible hacerlo por una que emplea una filosofía diferente a la planificada; se necesitaría reestructurar, perdiendo en tiempo.

Como se señaló con anterioridad, describe la navegación del producto a través de un diagrama de flujo y la utilización de media y sus tipos en tablas. No se centra en la especificación de la estructura al nivel de programación, llevando un nivel elemental el análisis y diseño; con la obtención de medias y la descripción de un proceso verbal fuerza al programador a llevar adelante la confección de módulos que pueden resultar complejos. Carece de herramientas de sostén para la descripción del proceso de implementación. Visto de esta forma, Multimet no es el proceso ideal a desarrollar, aunque aporta una visión inicial de cómo proyectarse.

2.3.2 El Proceso Unificado de Desarrollo de Software (RUP).

Con el desarrollo que está teniendo en la actualidad la industria del software, el avance y complejidad que alcanzan los nuevos sistemas informáticos, debido principalmente al auge de las computadoras, el aumento del rigor del usuario y el rápido crecimiento en el uso de Internet para el intercambio de todo tipo de información, James Rumbaugh, Grady Booch e Ivar Jacobson, autores de “El proceso unificado de desarrollo de software”, opinan que “El problema del software se reduce a la dificultad que afrontan los desarrolladores para coordinar las múltiples cadenas de

trabajo de un gran proyecto de software”. Para desarrollar un software se necesita una forma coordinada de trabajo, un proceso que integre las múltiples facetas del desarrollo [figura 6 anexo 1], cuyo objetivo será producir software de alta calidad, es decir, que cumpla con los requerimientos de los usuarios dentro de una planificación y presupuesto establecido, cubriendo el ciclo de vida y desarrollo de software [2].

El proceso unificado de desarrollo, RUP, es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. RUP permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Esto permite a los grupos de desarrollo producir aplicaciones informáticas más robustas y flexibles que se adaptan a las necesidades de los usuarios. La correcta aplicación de RUP permite reducir los tiempos de desarrollo, aumentar la calidad de las aplicaciones y disminuir los costes de mantenimiento. “Está basado en componentes, lo cuál quiere decir que el sistema software en construcción está formado por componentes software interconectados a través de interfaces bien definidas” RUP es un proceso de desarrollo de software que contiene un conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software de forma eficiente. Es el resultado de la experiencia de más de 30 años de trabajo y los autores [James Rumbaugh, Grady Booch e Ivar Jacobson] confirman que es la solución al problema del software.

2.3.3 Características del Proceso Unificado de Software.

Los aspectos definatorios y a la vez que lo convierten en único al Proceso Unificado, se resumen en tres fases: dirigido por casos de uso, centrado en la arquitectura, e iterativo e incremental.

- *Dirigido por casos de uso:* La razón de ser de un sistema es brindar servicios a los usuarios, RUP define caso de uso como el conjunto de acciones que debe realizar un sistema para dar un resultado de valor a un determinado usuario y los utiliza tanto para especificar los requisitos funcionales del sistema, como para guiar todos los demás pasos de su desarrollo, dígase diseño, implementación y prueba.

- *Estar centrado en la arquitectura:* La arquitectura es una vista del diseño completo con las características más importantes. Esta no sólo incluye las necesidades de los usuarios e inversores, sino también otros aspectos técnicos como el hardware, sistema operativo, sistema de gestión de base de datos, protocolos de red, con los que debe coexistir el sistema. La arquitectura representa la forma del sistema, la cual va madurando en su interacción con los casos de uso hasta llegar a un equilibrio entre funcionalidad y características técnicas.
- *Ser iterativo e incremental:* El alto nivel de complejidad de los sistemas actuales, hace que sea factible dividir el proceso de desarrollo en varios mini-proyectos. Cada uno de estos se les denomina iteración y pueden o no representar un incremento en el grado de terminación del producto completo. En cada iteración los desarrolladores seleccionan un grupo de casos de uso, los cuales se diseñan, implementan y prueban. La planificación de iteraciones hace que se reduzcan los riesgos de los costes de un solo incremento, no sacar al mercado un producto en el tiempo previsto, mantener la motivación del equipo pues puede ver avances claros a corto plazo y que el desarrollo pueda adaptarse a los cambios en los requisitos.

2.3.4 Rational Rose y El Lenguaje Unificado de Modelado (UML).

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, 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 y entregables. Esta herramienta propone la utilización de cuatro tipos de modelos para realizar un diseño del sistema, utilizando una vista estática y otra dinámica de los modelos del sistema, uno lógico y otro físico. Permite crear y refinar estas vistas creando de esta forma un modelo completo que representa el dominio del problema y el sistema de software. A continuación se muestran algunas de las características que tiene Rational:

- *Desarrollo Iterativo*: Utiliza un proceso de desarrollo iterativo controlado, donde el desarrollo se lleva a cabo en una secuencia de iteraciones. Cuando la implementación pasa todas las pruebas que se determinan en el proceso, ésta se revisa y se añaden los elementos modificados al modelo de análisis y diseño. Una vez que la actualización del modelo se ha modificado, se realiza la siguiente iteración.
- *Generador de Código*: Se puede generar código en distintos lenguajes de programación a partir de un diseño en UML.
- *Ingeniería Inversa*: Proporciona mecanismos para realizar la denominada Ingeniería Inversa, a partir del código de un programa, se puede obtener su diseño.
- *Trabajo en Grupo*: Permite varias personas trabajando a la vez en el proceso iterativo controlado, para ello posibilita que cada desarrollador opere en un espacio de trabajo privado que contiene el modelo completo y tenga un control exclusivo sobre la propagación de los cambios en ese espacio de trabajo.

UML oficialmente se presenta cuando Rumbaugh, Booch y Jacobson unifican sus estudios con una semántica y notación, para lograr compatibilidad en el análisis y diseño orientado a objetos, permitiendo que los proyectos se asentaran en un lenguaje de modelado maduro, enfocando a los constructores de herramientas en producir características más útiles.

El desarrollo de sistemas con UML siguiendo el proceso unificado incluye actividades específicas, cada una de ellas a su vez contienen otras subactividades las cuales sirven como una guía de cómo deben ser las actividades desarrolladas y secuenciadas con el fin de obtener sistemas exitosos; consecuentemente el desarrollo de los sistemas puede variar de desarrollador en desarrollador, de proyecto en proyecto, de empresa en empresa adoptando siempre un Proceso de Desarrollo.

2.3.5 Objetivo de UML como lenguaje de modelado.

- UML es un lenguaje de modelado de propósito general que pueden usar todos los modeladores. No tiene propietario y está basado en el común acuerdo de gran parte de la comunidad informática.
- UML no pretende ser un método de desarrollo completo. No incluye un proceso de desarrollo paso a paso. UML incluye todos los conceptos que se consideran necesarios para utilizar un proceso moderno iterativo, basado en construir una sólida arquitectura para resolver requisitos dirigidos por casos de uso.
- Ser tan simple como sea posible pero manteniendo la capacidad de modelar toda la gama de sistemas que se necesita construir. UML necesita ser lo suficientemente expresivo para manejar todos los conceptos que se originan en un sistema moderno, tales como la concurrencia y distribución, así como también los mecanismos de la ingeniería de software, como son la encapsulación y componentes.
- Debe ser un lenguaje universal, como cualquier lenguaje de propósito general.
- Imponer un estándar mundial.

2.3.6 Lenguaje Orientado a Objetos para el Modelado de Aplicaciones Multimedia (OMMMA – L).

En adición a las áreas de juegos interactivos, la educación y entretenimientos, las aplicaciones multimedia interactivas están ganando gran importancia en las áreas tradicionales de los sistemas de software. Como efecto, los investigadores de software multimedia abogan por el desarrollo de principios y métodos de ingeniería de software para la construcción de sistemas multimedia. Al mismo tiempo como profundización de estos anhelos, forma parte de la demanda de los constructores de multimedia el desarrollo de notaciones precisas semánticamente, y al mismo tiempo usables sintácticamente, que soporten las diferentes vistas y niveles de abstracción [3].

Muchos lenguajes de modelado han sido propuestos para la especificación del proceso de desarrollo de aplicaciones multimedia, aunque aún no existe un estándar que cubra todos los

aspectos relacionados con el comportamiento dinámico e interactivo asociado a las interfaces gráficas para una generalización de herramientas, productos y procesos. En medio de una búsqueda para una modelación adecuada, el Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia (OMMMA - L) se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos, y MVC (Modelo Vista Controlador) para la interfaz de usuario [figura 3 anexo 1] [4], siendo este un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. Esto es útil ya que los modelos típicamente tienen cierto grado de estabilidad (dependiendo de la estabilidad del dominio del problema que está siendo modelado), donde el código de la interfaz de usuario sea más robusto, debido a que el desarrollador esta menos propenso a "romper" el modelo mientras trabaja de nuevo en la vista.[5]

OMMMA-L está sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular. Estas vistas son: [6]

- Vista Lógica: modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación [figura 5 anexo 1].
- 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 diagrama tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización (texto, gráfico, video, animación) e interacción (scrolls, barras de menú, botones, campos de entrada y

salida, hipertextos con hipervínculos). Estos diagramas de presentación pueden ser divididos en capas virtuales de presentación donde en cada uno de ellas sólo se haga referencia a una clase específica de componentes (por ejemplo, una vista para los objetos de visualización y otra para los de interacción, u otro tipo de división para la representación de los intereses de los desarrolladores.).

- Vista de Comportamiento temporal predefinido: modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El Diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo. 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 que permite soportar su reusabilidad; marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media; activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.
- Vista de Control Interactivo: modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML, sin tácticamente igual a este último, mas con la diferencia semántica de que en el orden de unir los controles interactivos y predefinidos, no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados; queriendo esto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

Extendiendo el paradigma MVC para multimedia a las peculiaridades de comportamiento estático y dinámico identificadas anteriormente, obtenemos MVCMM [figura 4 anexo 1], sobre el que se basa las especificaciones de OMMMA – L **¡Error! No se encuentra el origen de la referencia..** Actualmente, OMMMA – L se evalúa en diferentes escenarios, como proyectos

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

2.3.7 Relationship Management Methodology (RMM).

El desarrollo de hipermedia, especialmente en una escala comercial, frecuentemente involucra equipos de desarrolladores quienes necesitan ser administrados y coordinados a través de un período extenso de tiempo. Sistemas formales de desarrollo y técnicas de administración de proyectos son necesarios para asegurar que los productos hipermediales alcancen sus objetivos y sean completados en el tiempo presupuestado. Sin embargo, las técnicas tradicionales en la industria del software deben ser modificadas para adecuarse a los nuevos requerimientos.

Los proyectos hipermediales difieren del desarrollo de software tradicional en varias dimensiones críticas. Primero, los proyectos hipermediales pueden involucrar personas con disciplinas muy distintas (autores, libretistas, diseñadores, artistas, músicos, como también programadores). Segundo, el diseño de aplicaciones hipermediales involucra captar y organizar la estructura de un dominio complejo para hacerla simple y accesible a los usuarios. Tercero, los aspectos multimediales de las aplicaciones hipermediales conllevan numerosas dificultades. El diseño hipermedia es por lo tanto un proceso desafiante que actualmente está más relacionado con el arte que con la ciencia. Finalmente la necesidad de prototipos y validaciones intensivas con usuarios es más pronunciada en desarrollos hipermediales que en el software tradicional, porque la tolerancia de los usuarios a errores en aplicaciones hipermediales es muy baja **¡Error! No se encuentra el origen de la referencia..**

El método RMM fue propuesto por primera vez en **¡Error! No se encuentra el origen de la referencia.**, fue la primera metodología para el diseño de multimedia; si bien se trataba ésta de una versión con múltiples limitaciones que al ser detectadas dieron lugar a una versión extendida, ERMM. Se trata, probablemente, del único método para hipermedia que parece cubrir todo el

ciclo de desarrollo, desde el estudio de factibilidad hasta la evaluación del sistema, aunque sólo propone actividades y productos concretos para las fases de análisis y de diseño.

La clase de aplicaciones para la cual RMM es más adecuada, corresponde a las que presentan una estructura regular para un dominio de interés, en donde hay clases de objetos, relaciones definibles entre éstas clases, y múltiples instancias de objetos dentro de cada clase. Muchas aplicaciones hipermediales satisfacen estos requerimientos, como por ejemplo, catálogos de productos, aplicaciones hipermediales frontales (front-end) para bases de datos tradicionales o aplicaciones legadas. Considerando que muchas aplicaciones hipermediales de este tipo poseen datos volátiles que requieren actualizaciones frecuentes, se hace necesario disponer de medios que permitan automatizar y agilizar los desarrollos iniciales y los subsecuentes procesos de actualización. **¡Error! No se encuentra el origen de la referencia.**

El análisis se realiza por medio de un diagrama entidad-relación en el que sólo se permiten relaciones con cardinalidades $1 \rightarrow 1$ o $1 \rightarrow N$ y tampoco es posible establecer relaciones reflexivas. Con respecto al diseño, una de las características más relevantes de este método es que éste se hace tanto de forma ascendente como descendente, ofreciendo una interesante manera de realizar una verificación. El diseño descendente empieza con la construcción de un diagrama de aplicación descendente, que es un esquema de las unidades de presentación (equiparables a ventanas) y de los enlaces que existen entre las mismas. A continuación, se compone cada una de esas unidades partiendo de las entidades del diagrama E-R, generando los denominados *m-slices*. En los *m-slices* se especifican los contenidos, enlaces, herramientas de navegación y funciones asociadas a cada unidad. El diseño ascendente toma como punto de partida los *m-slices* y genera un nuevo diagrama de aplicación ascendente que contrasta con el descendente. Para el resto de las fases, estudio de factibilidad, implementación, pruebas y evaluación, no se propone ningún tipo de guías.

RMM constituye una metodología tentadora para el desarrollo del proceso por el desglose de las fases de la producción y la incorporación de diagramas para el diseño de la presentación, el

comportamiento dinámico y la estructura de la navegación [figura 7 anexo 1]. No obstante, su uso óptimo se basa en las aplicaciones de catálogo de productos y aplicaciones hipermediales frontales para bases de datos tradicionales o aplicaciones legadas, por poseer una alta volatilidad de la información. En el lado opuesto del espectro, un trabajo artístico puede tener una estructura bastante difusa en la cual no se observen cambios frecuentes a través del tiempo, haciendo de RMM poco aplicable. Una multimedia educativa, entra en consideración de acuerdo al uso de los datos para su ejecución, fácilmente identificables son las plataformas de muestra de contenidos, mientras que las didácticas de contenido residente, se ven afectadas por el poco o nulo uso de esta metodología para su representación ingenieril. **¡Error! No se encuentra el origen de la referencia.**

2.3.8 XTRAS.

Los “XTRAS” son extensiones de código para aplicaciones de Macromedia, como Director y Authorware. A los usuarios el uso de Xtras les permite incorporar características personalizadas a las herramientas, conociendo como se utilizan. Xtras son módulos de objetos en código C que usa la Xtras API (Application Program Inteface) para ampliar la funcionalidad de las aplicaciones Macromedia.

Todas las Xtras API son implementadas usando la Arquitectura Abierta de Macromedia (MOA). MOA define un modelo objeto que provee caminos(o formas, o vías) para la interacción de aplicaciones y extensiones MOA proporciona mecanismos unificados para la implementación de extensiones de códigos (XTRAS) para acentuar la potencialidad de las aplicaciones de Macromedia, permite que los XTRAS se adapten a todas las aplicaciones en un conjunto único de tratados y términos de programación.

2.4 Conclusiones.

Una vez vista las características de OMMMA – L, se argumenta su aplicación exitosa partiendo de la idea de que no es un lenguaje nuevo, sino una extensión del UML, por lo que es necesario interpretar las características extendidas, centrados a la lógica de funcionamiento de una multimedia, que es por lo general, sencilla. Muestra análisis similares a otras metodologías potentes como RMM y no se especializa en una clasificación de producto, sino que generaliza a través del uso de la semántica original de UML. Es robusto y altamente descriptivo, refleja el proceso 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.

Por su parte, RUP modela, centrado en la arquitectura del software, una serie de artefactos que permite la división de responsabilidades dentro de la ingeniería, la construcción precisa de la estructura del producto y un mecanismo de producción iterativa e incremental que dividen en pasos dentro el flujo de trabajo garantizando el crecimiento. Por tanto se decide basar la propuesta de este trabajo en la descripción de un proceso basado en RUP, y utilizando OMMMA – L para la especificación del producto y la utilización de la herramienta Macromedia Flash.

Capítulo 3 Descripción de la solución propuesta.

3.1 Introducción.

En este capítulo se realiza un análisis del producto. Se plantean una serie de requisitos obtenidos de entrevistas con los clientes y del análisis del guión del proyecto. Se obtienen y describen los casos de uso que nos guiarán en la solución del sistema que se desarrolla centrándose en el Proceso Unificado de Desarrollo de Software, haciendo uso Lenguaje de Modelado Orientado a Objetos de Aplicaciones Multimedia (OMMMA-L) como lenguaje extendido.

Ha sido de gran utilidad la herramienta CASE Rational Rose para el modelado de la aplicación. Debido a la poca estructuración de los procesos de negocio se plantea un modelo de dominio ayudando a una mejor comprensión de los conceptos del sistema.

3.2 Especificación del Contenido.

El contenido esta dividido en cuatro módulos, en el módulo principal se hace referencia a la presentación y créditos de la multimedia, en el primer módulo se hace referencia a los conceptos básicos de la electricidad, en el segundo módulo se clasifican los artefactos electrodomésticos y sus componentes que trabajan con calor y en el tercer módulo se realizan los ejercicios correspondientes para ejercitar el contenido dado.

3.3 Descripción del Sistema Propuesto.

Los electrodomésticos son artefactos eléctricos diseñados para facilitar las tareas domésticas, tales como: planchar, cocinar, lavar, licuar frutas y verduras, moler y amasar, secar la ropa, aspirar el polvo, secar el cabello, tostar el pan, entre otros. Algunos de estos artefactos se

emplean todos los días; ninguno debe representar dificultades para su reparación si tenemos el ingenio suficiente y pensamos un poco el problema que nos plantea; si el reparador o su poseedor conoce los principios generales que rigen estas reparaciones y tiene algunas nociones básicas de electricidad.

3.4 Modelo Conceptual.

3.4.1 Diagrama de Clases del Modelo del Dominio.

Debido al bajo nivel de estructuración que presenta el negocio que se está estudiando se propone un modelo del dominio, ya que nos permite de manera visual mostrar al usuario los principales conceptos que se manejan en el dominio del sistema en desarrollo. Esto ayuda a los usuarios, desarrolladores e interesados, a utilizar un vocabulario común para poder entender el contexto en que se emplaza el sistema. Para capturar correctamente los requisitos y poder construir un sistema correcto se necesita tener un firme conocimiento del funcionamiento del objeto de estudio.

Este modelo va a contribuir posteriormente a identificar algunas clases que se utilizarán en el sistema. Se realiza la descripción del modelo de dominio mediante un diagrama de clases UML donde se especifican las principales clases conceptuales que pueden intervenir en este sistema, estos representarán los objetos que existen o eventos que suceden en el entorno en el que trabajará el sistema.

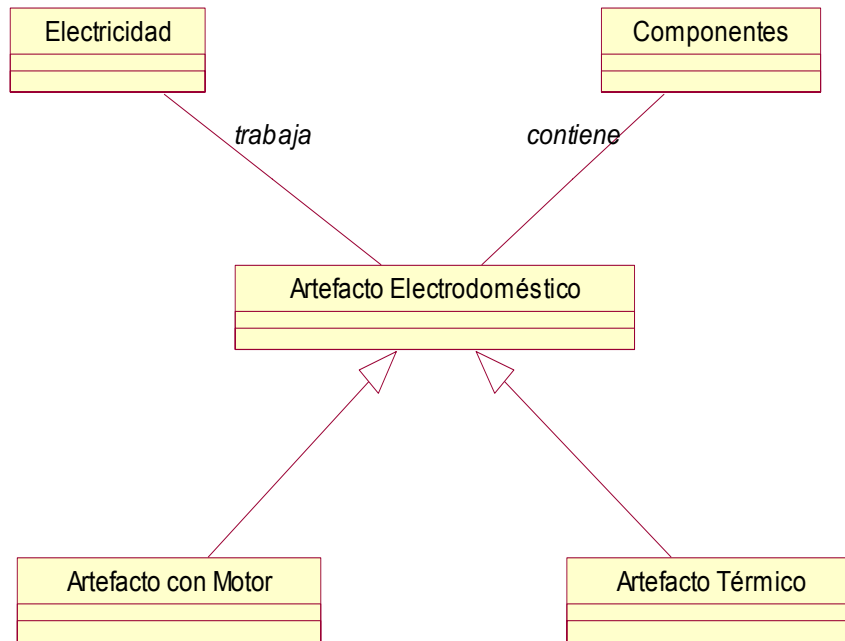


Figura 3. 1 Diagrama de clases del modelo de dominio.

3.4.2 Análisis de los Conceptos del Dominio.

Identificación de conceptos que se utilizarán en el diagrama, mediante un glosario de términos sobre los nombres:

- Se le llamará **artefacto electrodoméstico** al equipo que se utiliza para realizar alguna labor doméstica.
- Se le llamará **artefacto térmico** al equipo electrodoméstico que trabaja con el Efecto Joule (calor).
- Se le llamará **artefacto con motor** al equipo electrodoméstico que trabaja con motor eléctrico.
- Se le llamará **componentes** a las partes que integran un artefacto.
- Se denominará **electricidad** a la energía que utilizan los artefactos.

3.4.3 Diagrama de Navegación.

El diagrama de navegación brinda una visión de lo que se desea lograr, y nos ayuda a crear una idea del camino que se debe de seguir.

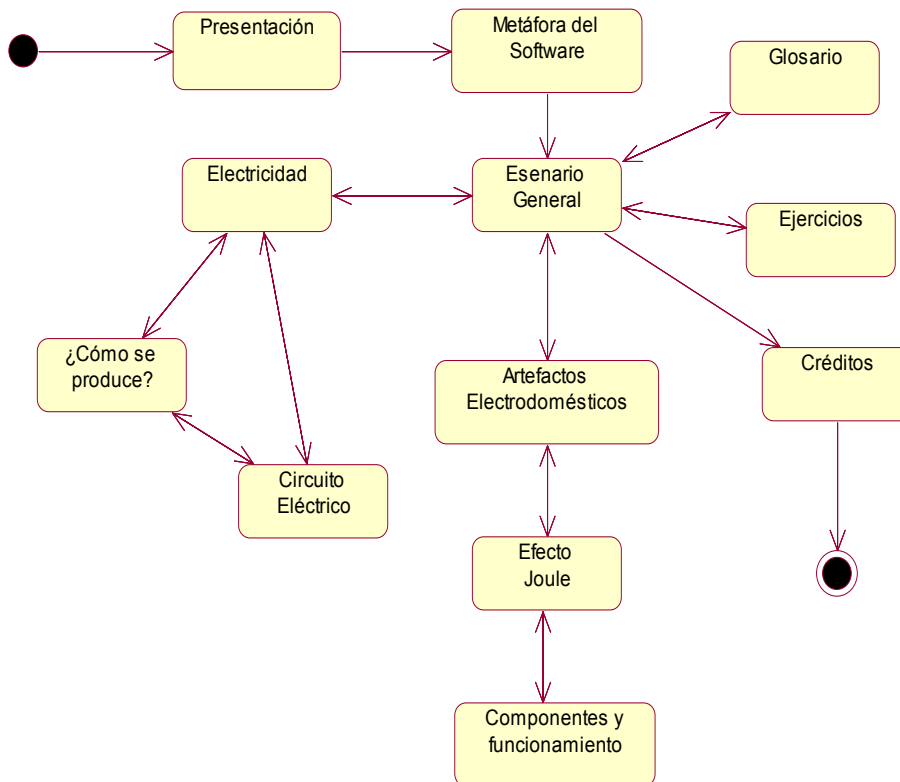


Figura 3. 2 Diagrama de Navegación.

3.5 Modelo de Análisis.

3.5.1 Requisitos Funcionales.

Ref.	Función
R1	Mostrar presentación.
R1.1	Representar la metáfora del software.
R2	Mostrar elementos de la electricidad.
R2.1	Mostrar como se produce la electricidad.
R2.1.2	Mostrar imágenes.
R2.2	Mostrar principios básicos del circuito eléctrico.
R3	Mostrar artefactos electrodomésticos térmicos.
R3.1	Mostrar aplicación del efecto Joule.
R3.2	Mostrar componentes de cada artefacto.
R5	Mostrar ejercicios.
R5.1	Comprobar respuesta.

Tabla 3. 1 Requisitos Funcionales.

3.5.2 Requisitos no Funcionales.

- Tiene que existir consistencia entre la presencia de elementos de la interfaz y la funcionalidad prevista en cada situación. (no deben existir elementos interactivos activos o visibles cuya funcionalidad no sea necesaria en un contexto dado Ej. Deslizador donde no haya texto que deslizar).
- Poseer un deslizador interactivo susceptible al evento Arrastrar y Soltar.
- Manipular el audio de fondo (On/Off).
- Permitir mostrar el Glosario del sistema.
- Permitir salir del sistema en cualquier momento.

3.5.3 Apariencia o Interfaz Externa.

- Diseño sencillo, permitiendo la utilización del sistema sin mucho entrenamiento.

3.5.4 Navegación.

- Desde cualquier pantalla se podrá acceder a cualquier módulo. (se exceptúan los casos en que por razones de diseño instruccional o gráfico se inhiba esta característica).
- Se podrá abandonar el programa desde cualquier pantalla.

3.5.5 Servicios generales.

Todas las pantallas presentarán los siguientes servicios generales con los botones interactivos siguientes: *música*, *inicio*, *glosario*, *salir*. Los cuales se activarán o desactivarán según el contexto.

- Especificaciones del servicio **música**.
 - El botón música funcionará como un conmutador ON/OFF activando o desactivando la música según corresponda.
 - Una vez desactivada la música, esta no se volverá a activar hasta que el usuario no lo vuelva a solicitar.

3.6 Modelo de casos de uso del sistema.

El modelo de casos de uso se forma mediante la utilización de factores tales como actores, descripción de sus acciones y relaciones fundamentales representando las principales funcionalidades del sistema.

3.6.1 Determinación y justificación de los actores del sistema.

Los actores de un sistema son agentes externos: aquellas personas o sistemas que interactúan con él. A continuación se describen en la siguiente tabla.

Actor del Sistema	Justificación
Usuario	Es la persona que inicia las acciones más importantes del sistema, y al mismo tiempo es el principal beneficiado con el resultado de dicha aplicación.

Tabla 3. 2 Determinación de los actores.

3.6.2 Descripción y expansión de los casos de uso.

Los casos de uso son fragmentos de funcionalidad del sistema. En ellos se describe la secuencia determinada de eventos que realiza un actor en interacción con la aplicación. A continuación se presentan mediante módulos.

Ref #	Casos de uso	Prioridad
CUS 1	Mostrar Presentación.	Crítico
CUS 2	Introducir Electricidad.	Crítico
CUS 3	Introducir Circuito Eléctrico.	Secundario
CUS 4	Mostrar Imágenes.	Crítico
CUS 5	Introducir Artefactos Térmicos.	Crítico
CUS 6	Mostrar Efecto Joule.	Crítico
CUS 7	Mostrar Componentes.	Crítico
CUS 8	Ejercitar Contenido.	Crítico
CUS 9	Verificar Respuesta.	Secundario

Tabla 3. 3 Casos de uso.

3.6.3 Módulo Principal.

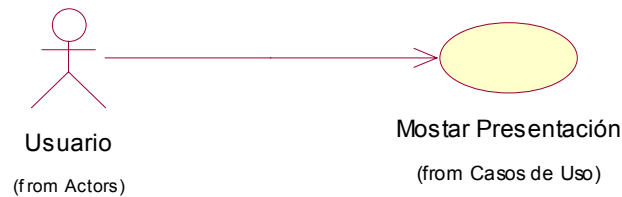


Figura 3. 3 Diagrama de casos de uso del Mostrar Presentación.

Caso de uso	
CUS 1	Mostrar Presentación.
Propósito	Que el usuario conozca la metáfora del software y el objetivo del mismo.
Actores: Usuario	
Resumen: Se inicia cuando el usuario ejecuta la aplicación. Se presentan los logos de los organismos patrocinantes, seguidamente se presenta la Metáfora del software en forma de vista aérea con elementos que forman parte de la conducción y el servicio de la corriente eléctrica, y en perspectiva la sala – comedor de una vivienda y seguidamente en el centro de la pantalla el título del software.	
Referencias	R1, R1.1
Acción del actor	Respuesta del sistema
1. El usuario ejecuta la aplicación.	1.1 El sistema carga la presentación 1.2 El sistema finaliza la ejecución del caso de uso y solicita la pantalla general.
Flujo alternativo: El usuario puede pasar la pantalla de presentación del producto.	
Acción del actor	Respuesta del sistema
2. El usuario acciona el clic del Mouse y después presiona la tecla ENTER.	2.1 El sistema solicita la pantalla general de la aplicación.
Puntos de extensión.	

Tabla 3. 4 Caso de uso Mostrar Presentación.

3.6.4 Módulo Electricidad.

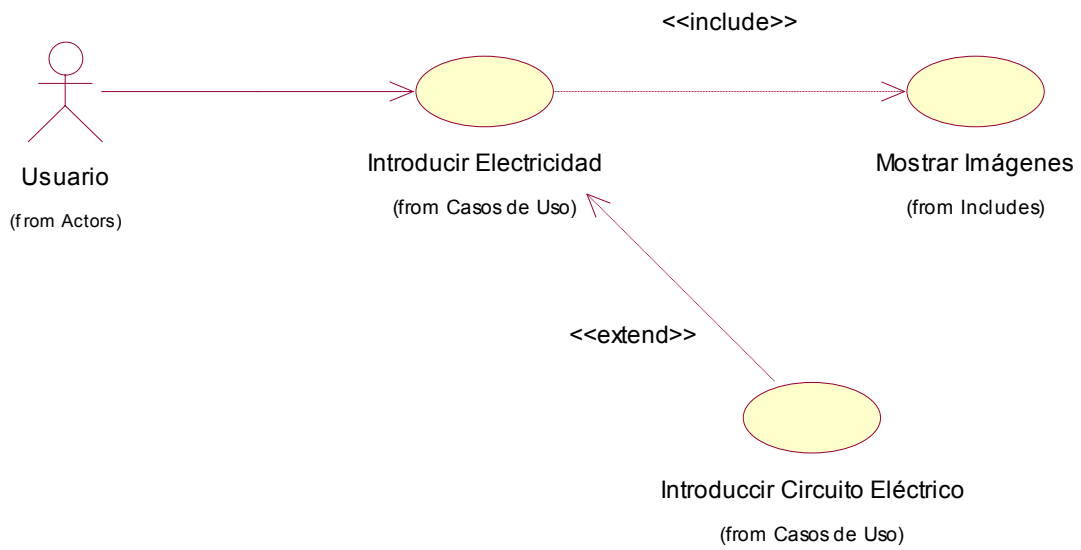


Figura 3. 4 Diagrama de caso de uso Introducir Electricidad.

Caso de uso	
CUS 2	Introducir Electricidad.
Propósito	Que el usuario conozca sobre el concepto básico de la energía que utilizan los artefactos electrodomésticos.
Actores: Usuario	
Resumen: Se explica cómo se genera la electricidad que llega a los hogares y se utiliza para el funcionamiento de los artefactos electrodomésticos.	
Referencias	R2, R2.1, CUS 3, CUS 4
Acción del actor	Respuesta del sistema
1. El usuario accede a la pantalla que aborda sobre el concepto básico sobre electricidad.	1.1 El sistema muestra un conjunto de imágenes y animaciones mostrando los diferentes tipos de energía eléctrica que existen y de que forma se produce.
Flujo alternativo: El usuario puede pasar la pantalla que aborda sobre la electricidad.	
Acción del actor	Respuesta del sistema
2. El usuario accede directamente a la pantalla que aborda sobre el circuito eléctrico simple.	2.1 El sistema muestra la pantalla correspondiente al circuito eléctrico.
Puntos de extensión.	

Tabla 3. 5 Caso de uso Introducir Electricidad.

Caso de uso	
CUS 3	Introducir Circuito Eléctrico.
Propósito	Que usuario conozca sobre el circuito eléctrico simple.
Actores: Usuario	
Resumen: Se explica el funcionamiento del circuito eléctrico y las partes que lo integran. Aparece la representación gráfica que escenifica un circuito eléctrico, con los siguientes elementos: toma corriente, interruptor, bombillo, fusible, conductores y fuente de energía, en cada caso se desplegará un texto con la información correspondiente.	
Referencias	R2.2
Acción del actor	Respuesta del sistema
1. El usuario accede a la pantalla que aborda sobre el circuito eléctrico.	1.1 El sistema muestra a través de una imagen los componentes de un circuito eléctrico.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Tabla 3. 6 Caso de uso Introducir Circuito Eléctrico.

Caso de uso	
CUS 4	Mostrar Imágenes.
Propósito	Permitirle al usuario obtener información visual sobre la electricidad y el circuito eléctrico.
Actores: Usuario	
Resumen: Se muestran imágenes de una represa, de una termoeléctrica, de unas hélices y de un conjunto residencial, en cada caso se escuchará un sonido y se desplegará un texto con la información correspondiente con respecto a la energía hidroeléctrica, termoeléctrica, eólica y residencial.	
Referencias	R2.1.2
Acción del actor	Respuesta del sistema
1. El usuario accede a la pantalla correspondiente a las imágenes.	1.2 El sistema muestra las imágenes.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Tabla 3. 7 Caso de uso Mostrar Imágenes.

3.6.5 Módulo Artefactos Térmicos.

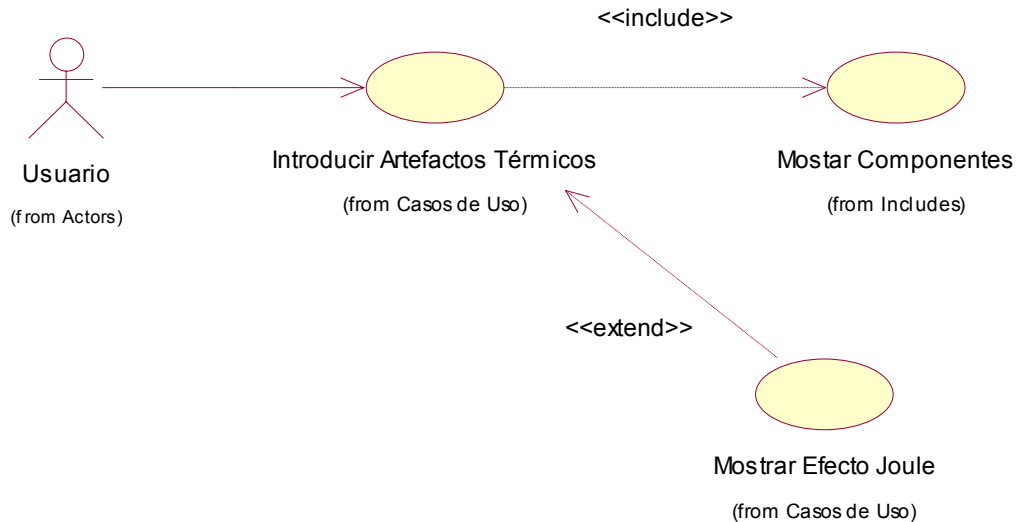


Figura 3. 5 Diagrama de caso de uso Introducir Artefactos Térmicos.

Caso de uso	
CUS 5	Introducir Artefactos Térmicos.
Propósito	Permitirle al usuario obtener información visual sobre la electricidad y el circuito eléctrico.
Actores: Usuario	
Resumen: Se clasifican los artefactos electrodomésticos que trabajan con calor (térmicos) en el hogar, entre los cuales se nombra algunos de los más usados: la plancha eléctrica de vapor, la tostadora eléctrica horizontal, la cocina eléctrica y la cafetera eléctrica.	
Referencias	R3
Acción del actor	Respuesta del sistema
1. El usuario accede a la pantalla correspondiente a los artefactos térmicos.	1.2 El sistema muestra a través de imágenes, textos y sonidos los artefactos térmicos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Tabla 3. 8 Caso de uso Introducir Artefactos Térmicos.

Caso de uso	
CUS 6	Mostrar Efecto Joule.
Propósito	Que el usuario conozca como se produce el efecto Joule en los artefactos térmicos.
Actores: Usuario	
Resumen: Se explica el proceso de aplicación del Efecto Joule en los artefactos electrodomésticos de uso común en el hogar.	
Referencias	R3.1
Acción del actor	Respuesta del sistema
1. El usuario accede a la pantalla correspondiente al efecto Joule.	1.1 El sistema muestra a través de imágenes y textos como se produce el efecto Joule en los artefactos térmicos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Tabla 3. 9 Caso de uso Mostrar Efecto Joule.

Caso de uso	
CUS 7	Mostrar Componentes.
Propósito	Que el usuario conozca sobre el funcionamiento y componentes de cada artefacto térmico.
Actores: Usuario	
Resumen: Se identifican los componentes y funcionamiento de la plancha eléctrica de vapor, cocina eléctrica, tostadora eléctrica horizontal y la cafetera eléctrica.	
Referencias	R3.2
Acción del actor	Respuesta del sistema
1. El usuario da clic sobre el botón que aborda sobre el funcionamiento de los artefactos térmicos. 2. El usuario da clic sobre el botón que aborda sobre los componentes de los artefactos térmicos.	1.1 El sistema muestra a través de un texto en que consiste el funcionamiento de los artefactos térmicos. 1.2 El sistema muestra a través de imágenes los componentes de los artefactos térmicos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Tabla 3. 10 Caso de uso Mostrar Componentes.

3.6.7 Módulo Ejercitación.

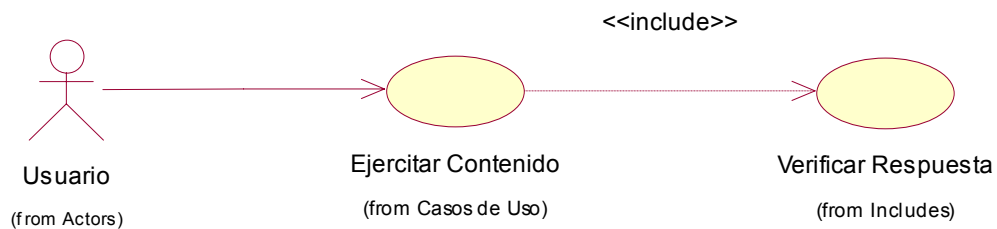


Figura 3. 6 Diagrama de caso de uso Ejercitar Contenido.

Caso de uso	
CUS 8	Ejercitar Contenido.
Propósito	Que el usuario aplique los conocimientos adquiridos mediante la aplicación.
Actores: Usuario	
Resumen: Se muestra una serie de cinco ejercicios con sus respectivos incisos donde se evalúa la identificación de los electrodomésticos que trabajan con el Efecto Joule, identificar los componentes de una plancha de vapor, identificar los componentes de la cocina eléctrica, identificar los componentes externos e internos de una tostadora eléctrica, identificar los componentes de una cafetera eléctrica, identificar los artefactos térmicos y los artefactos que trabajan con motor eléctrico y se presenta una auto evaluación con preguntas vinculadas a los contenidos de este módulo.	
Referencias	R5, CUS 9
Acción del actor	Respuesta del sistema
<ol style="list-style-type: none"> 1. El usuario accede al módulo de ejercicios y selecciona el ejercicio 1. 2. El usuario accede al ejercicio 2. 3. El usuario accede al ejercicio 3. 4. El usuario accede al ejercicio 4. 5. El usuario accede al ejercicio 5. 	<ol style="list-style-type: none"> 1.1 El sistema muestra el ejercicio 1 con sus Incisos. 2.1 El sistema muestra el ejercicio 2. 3.1 El sistema muestra el ejercicio 3. 4.1 El sistema muestra el ejercicio 4 con sus Incisos. 5.1 El sistema muestra el ejercicio 5 con sus Incisos.
Flujo alternativo	
Acción del actor	Respuesta del sistema
Puntos de extensión.	

Tabla 3. 11 Caso de uso Ejercitar Contenido.

Caso de uso	
CUS 9	Verificar Respuesta.
Propósito	Que el usuario verifique las respuestas seleccionadas.
Actores: Usuario	
Resumen: La aplicación tiene habilitado un botón para las verificaciones de las respuestas.	
Referencias	R5.1
Acción del actor	Respuesta del sistema
1. El usuario verifica manualmente la respuesta.	1.1 En caso que la respuesta sea correcta el sistema felicita al usuario. 1.2 En caso que la respuesta sea incorrecta el sistema advierte que lo intente de nuevo.
Flujo alternativo: El sistema verifica de forma automática la respuesta seleccionada por el usuario.	
Acción del actor	Respuesta del sistema
2. El usuario termina el ejercicio seleccionado.	2.1 En caso que la respuesta sea correcta el sistema felicita al usuario de forma automática. 2.2 En caso que la respuesta sea incorrecta el sistema advierte de forma automática que lo intente de nuevo.
Puntos de extensión.	

Tabla 3. 12 Caso de uso Verificar Respuesta.

3.7 Conclusiones.

Con el desarrollo de este capítulo se ha permitido una mejor comprensión del contexto a automatizar, realizando una descripción de la solución propuesta y definiendo los requisitos funcionales y no funcionales con que debe cumplir. Mediante los diagramas de casos de uso del sistema se representó la acción del actor sobre este. Se realiza una descripción de cada uno de los casos de uso. Gracias a la culminación de este flujo, ahora podemos empezar a construir el sistema, tratando de que se cumplan todos los requerimientos y las funciones que se han considerado necesarias en este capítulo.

Capítulo 4 Construcción de la solución propuesta.

4.1 Introducción.

La Programación Orientada a Objetos (POO u OOP según siglas en inglés) es un paradigma de programación que define los programas en términos de "clases de objetos", objetos que son entidades que combinan estado (es decir, datos), comportamiento (esto es, procedimientos o métodos) e identidad (propiedad del objeto que lo diferencia del resto). La programación orientada a objetos expresa un programa como un conjunto de estos objetos, que colaboran entre ellos para realizar tareas. Esto permite hacer los programas y módulos más fáciles de escribir, mantener y reutilizar [7].

El Proceso Unificado Racional o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje de Modelado Orientados a Objetos para Aplicaciones Multimedia (OMMMA - L) constituye la metodología más adecuada a utilizar para el análisis, implementación y documentación de aplicaciones multimedia [8].

En el capítulo 2 se hace referencia a estos conceptos pero era válido hacerle mención ya que la fase de análisis y diseño se desarrolla empleando los artefactos de OMMMA-L. OMMMA-L plantea una vista de presentación espacial modelada a través de los diagramas de presentación. Estos diagramas pueden ser divididos en capas virtuales de presentación donde en cada uno de ellas sólo se haga referencia a una clase específica de componentes (por ejemplo, una vista para los objetos de visualización y otra para los de interacción, u otro tipo de división para la representación de los intereses de los desarrolladores.).

4.2 Modelo de Diseño.

4.2.1 Diagramas de Presentación.

Este es un artefacto nuevo dentro de la extensión del lenguaje UML, es específico de OMMMA – L, y sirve para describir la parte estática del modelo a través de una descripción intuitiva de la distribución espacial de objetos visuales de la interfaz de usuario. Aunque UML especifica propuesta de interfaz de usuario en sus requisitos no funcionales, no es un aspecto de fuerte medición, ni consideración en el análisis de la arquitectura del software.

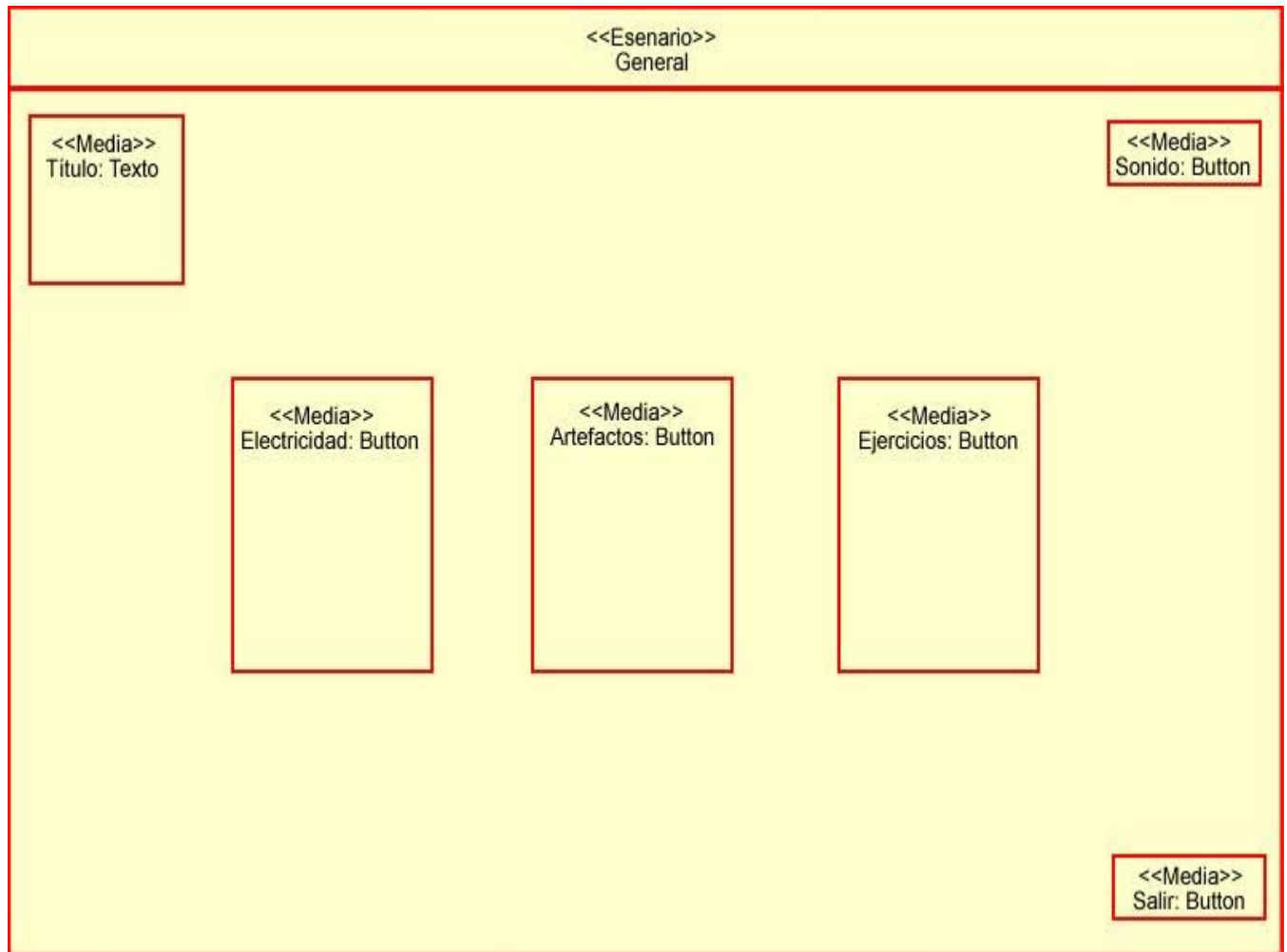


Figura 4. 1 Diagrama de Presentación para el Módulo Principal

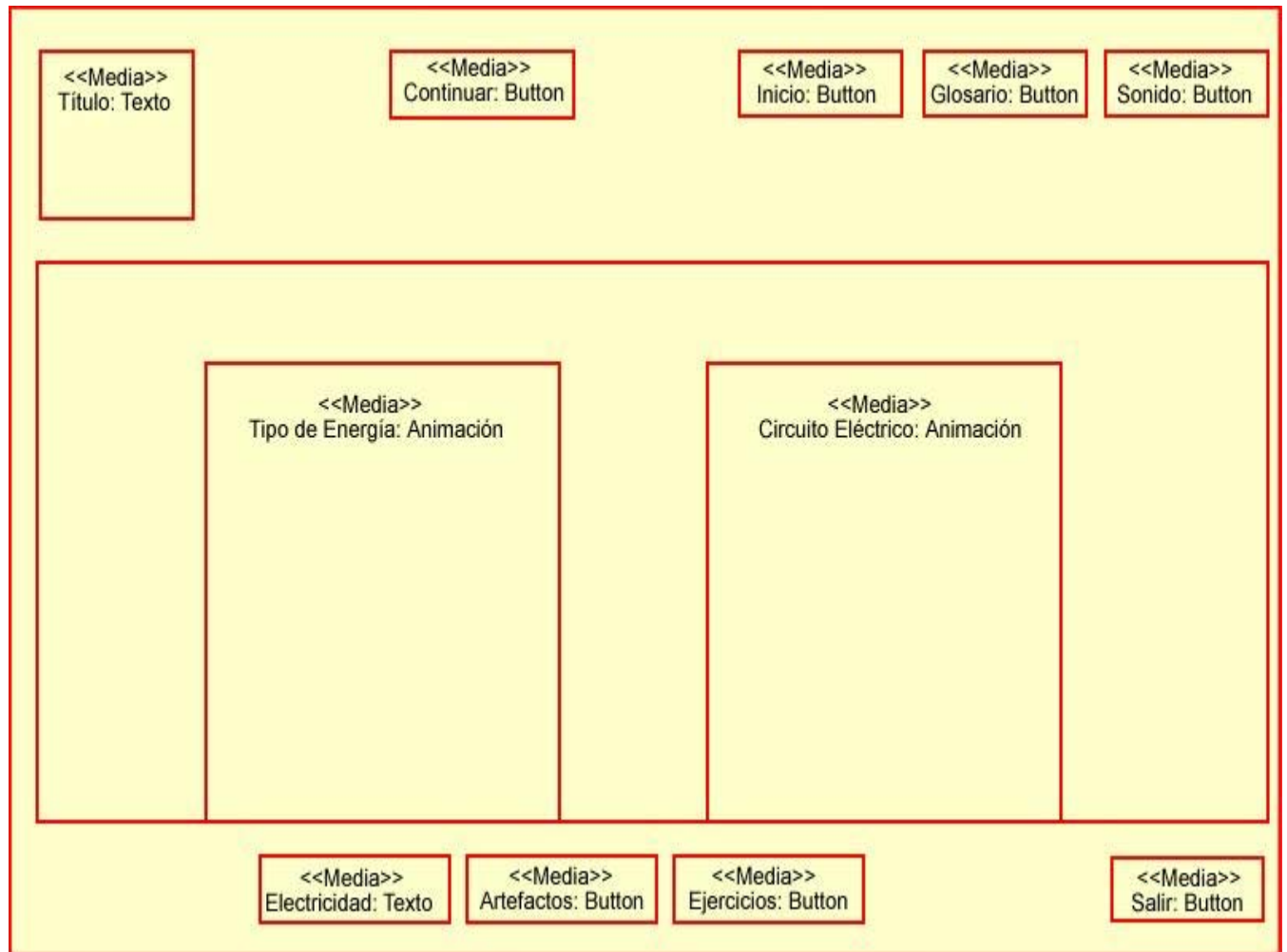


Figura 4. 2 Diagrama de Presentación para la primera del Módulo I

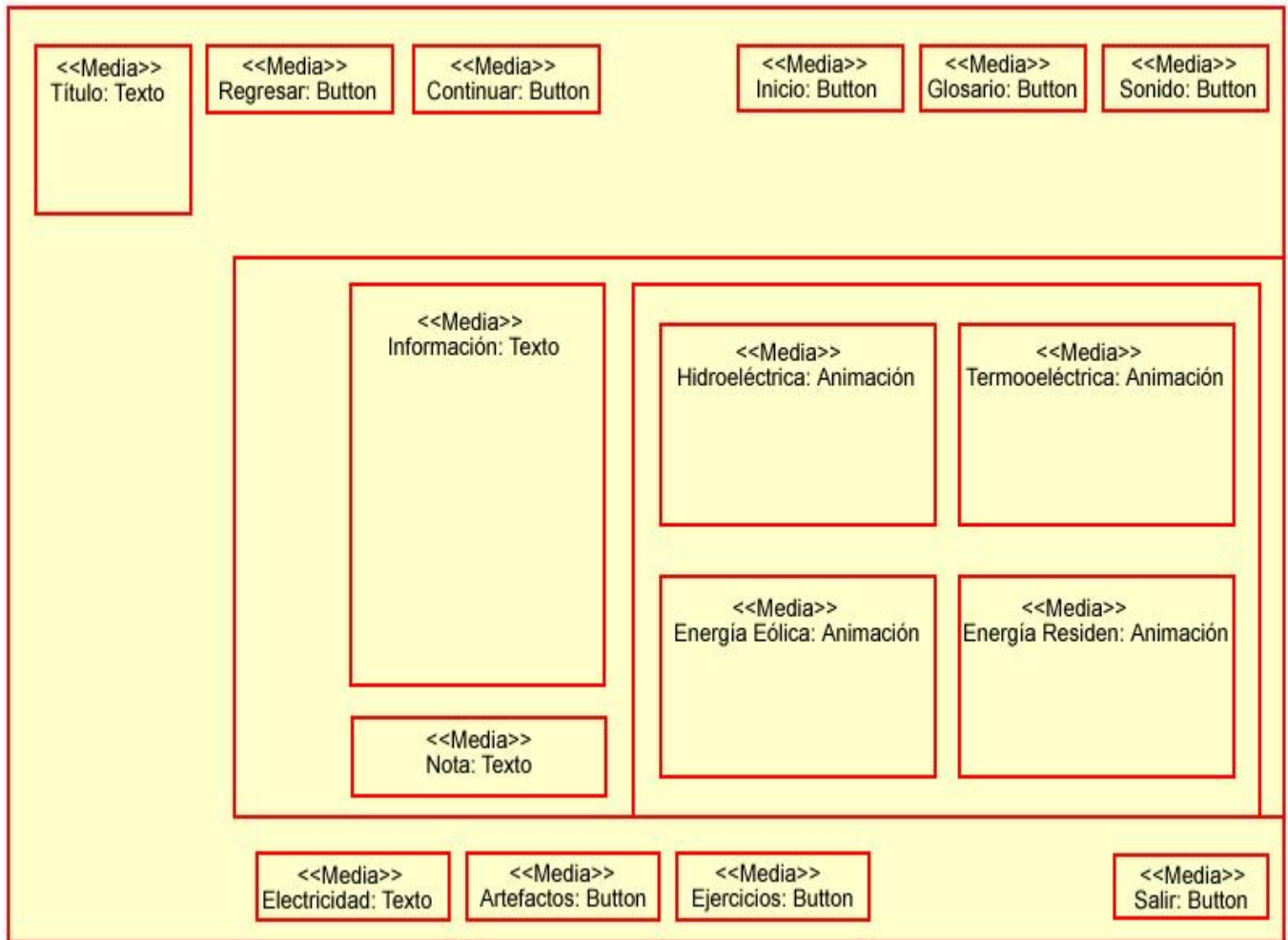


Figura 4. 3 Diagrama de Presentación para la segunda pantalla del Módulo I

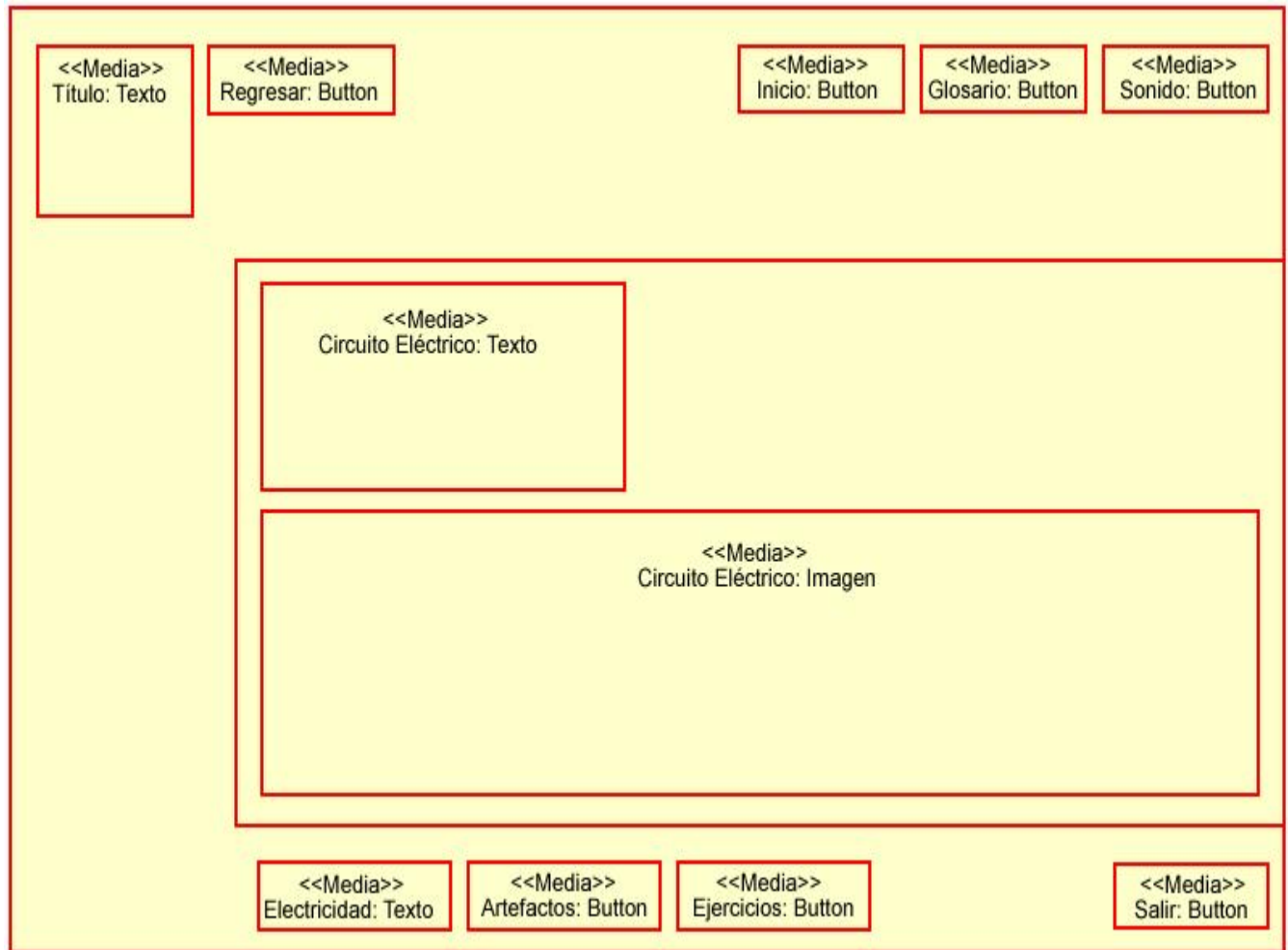


Figura 4. 4 Diagrama de Presentación para la tercera pantalla del Módulo I

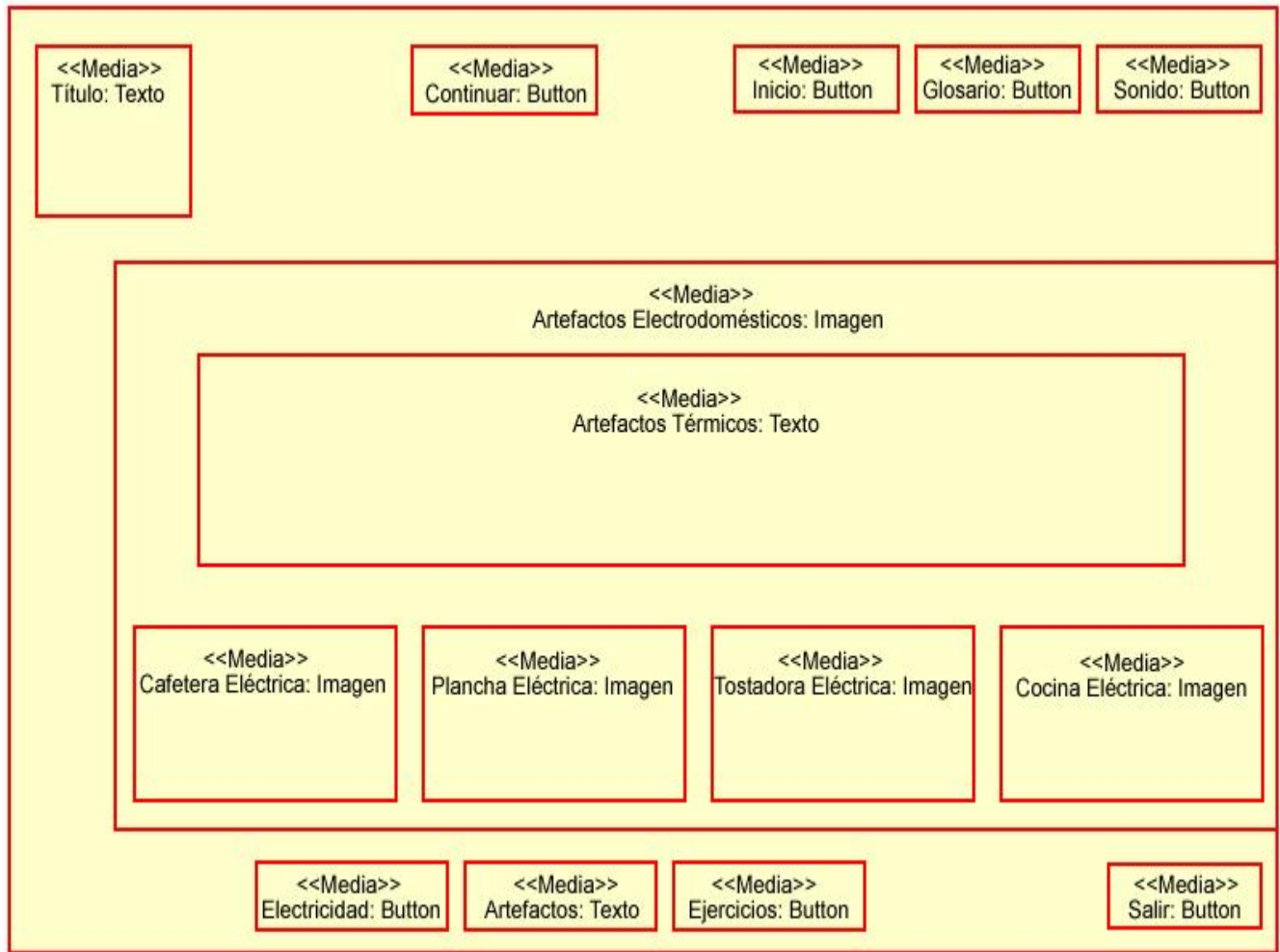


Figura 4. 5 Diagrama de Presentación para la primera pantalla del Módulo II

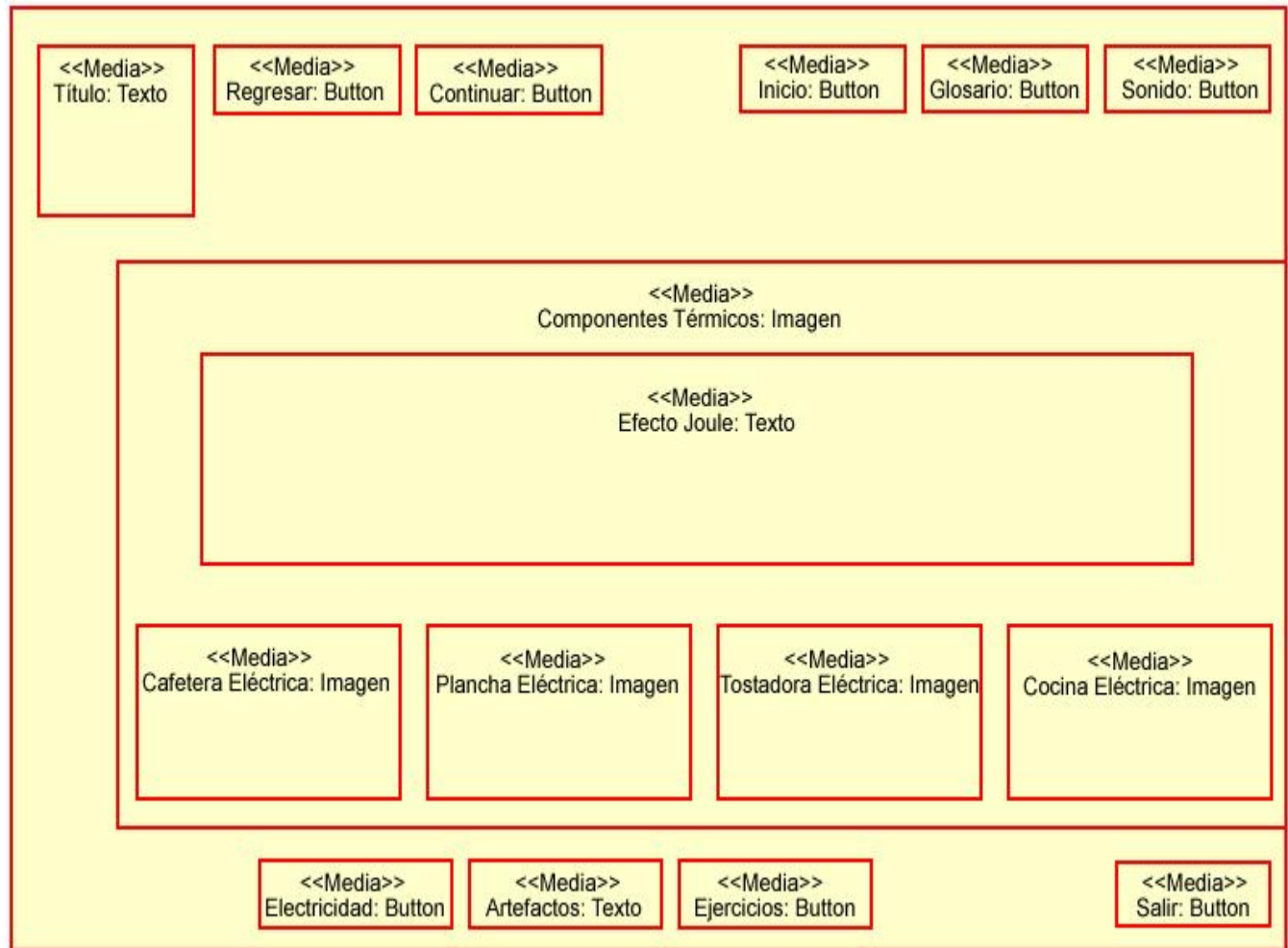


Figura 4. 6 Diagrama de Presentación para la segunda pantalla del Módulo II

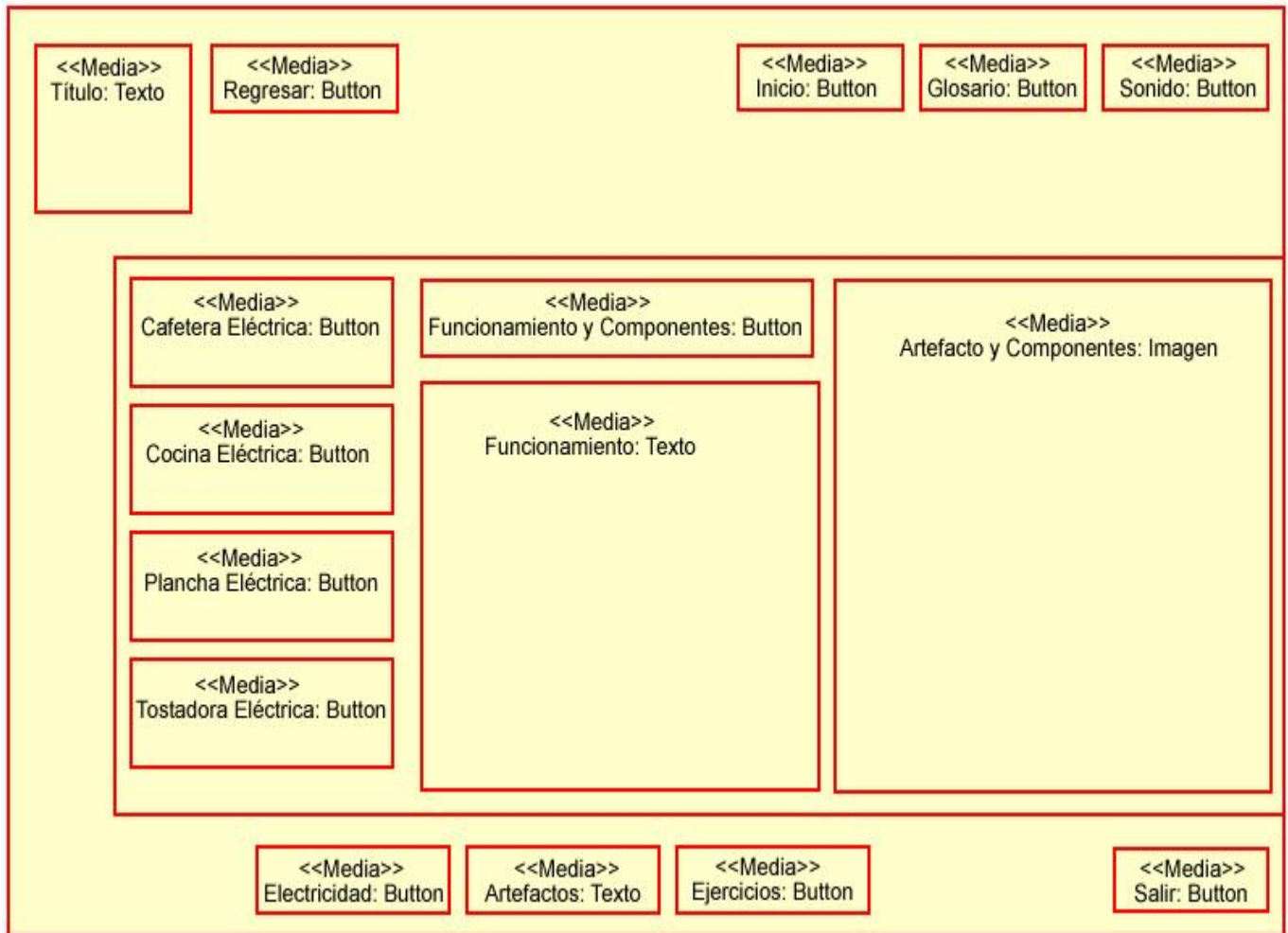


Figura 4. 7 Diagrama de Presentación para la tercera pantalla del Módulo II

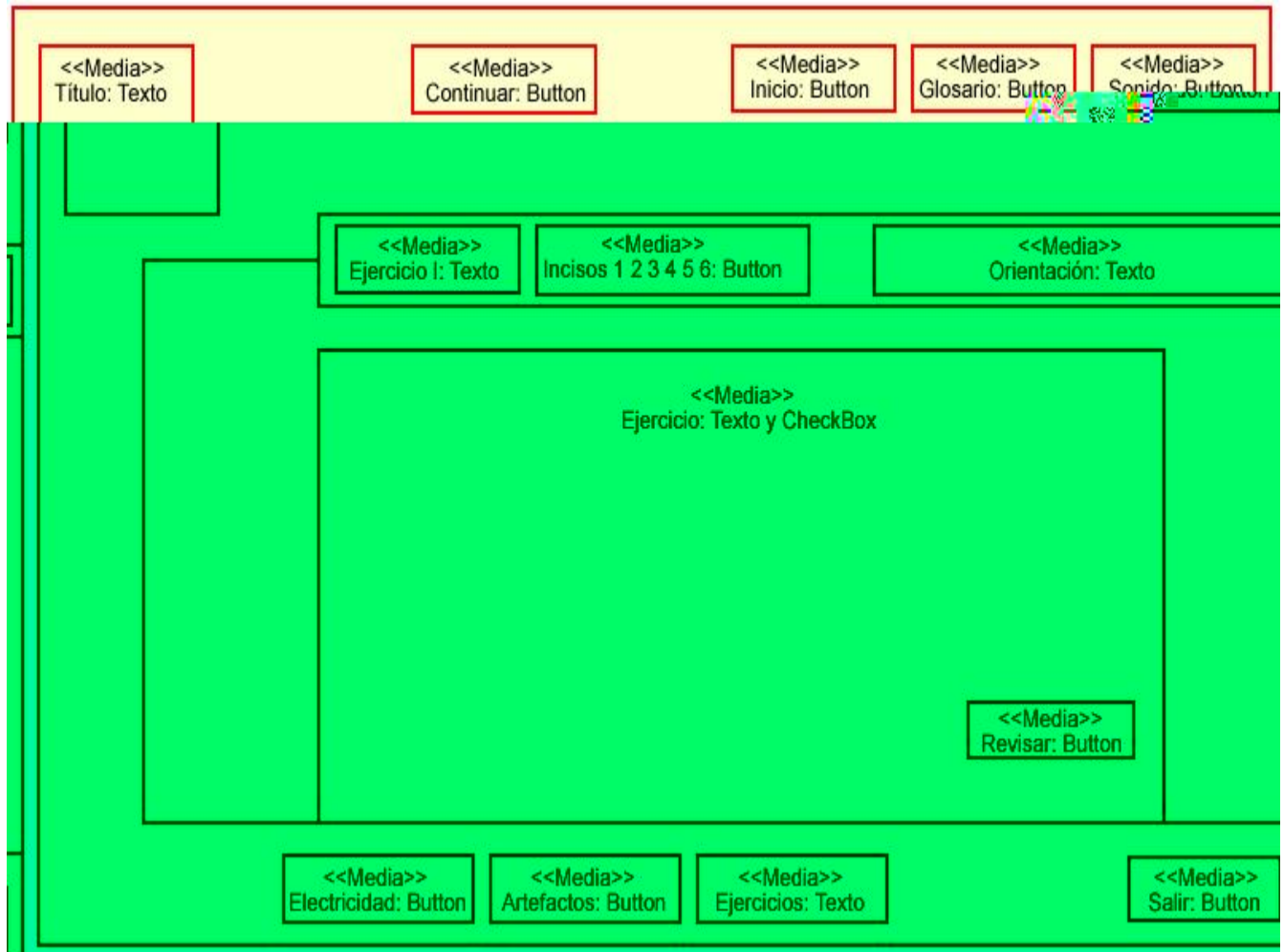


Figura 4. 8 Diagrama de Presentación para la primera pantalla del Módulo III

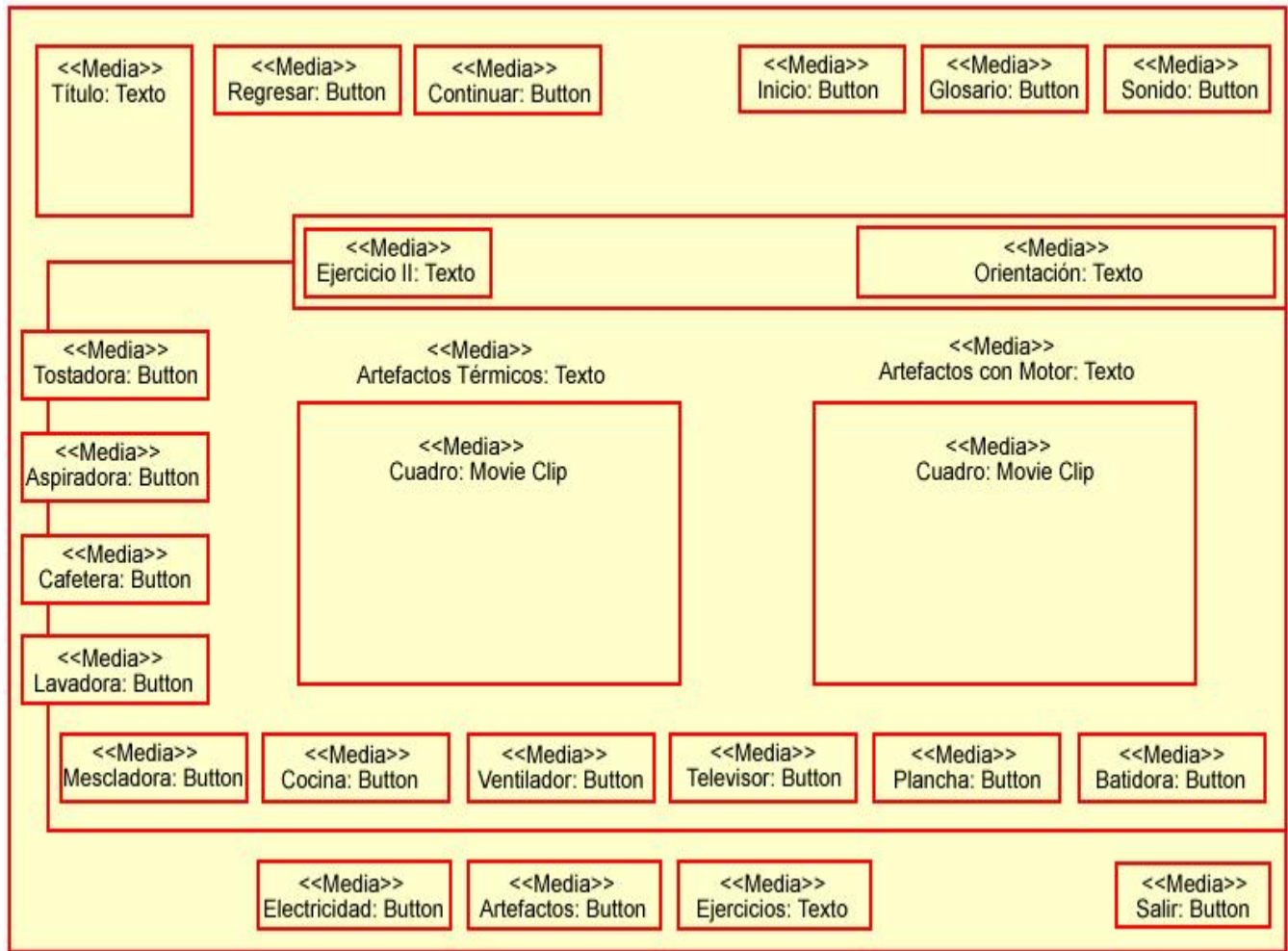


Figura 4. 9 Diagrama de Presentación para la segunda pantalla del Módulo III

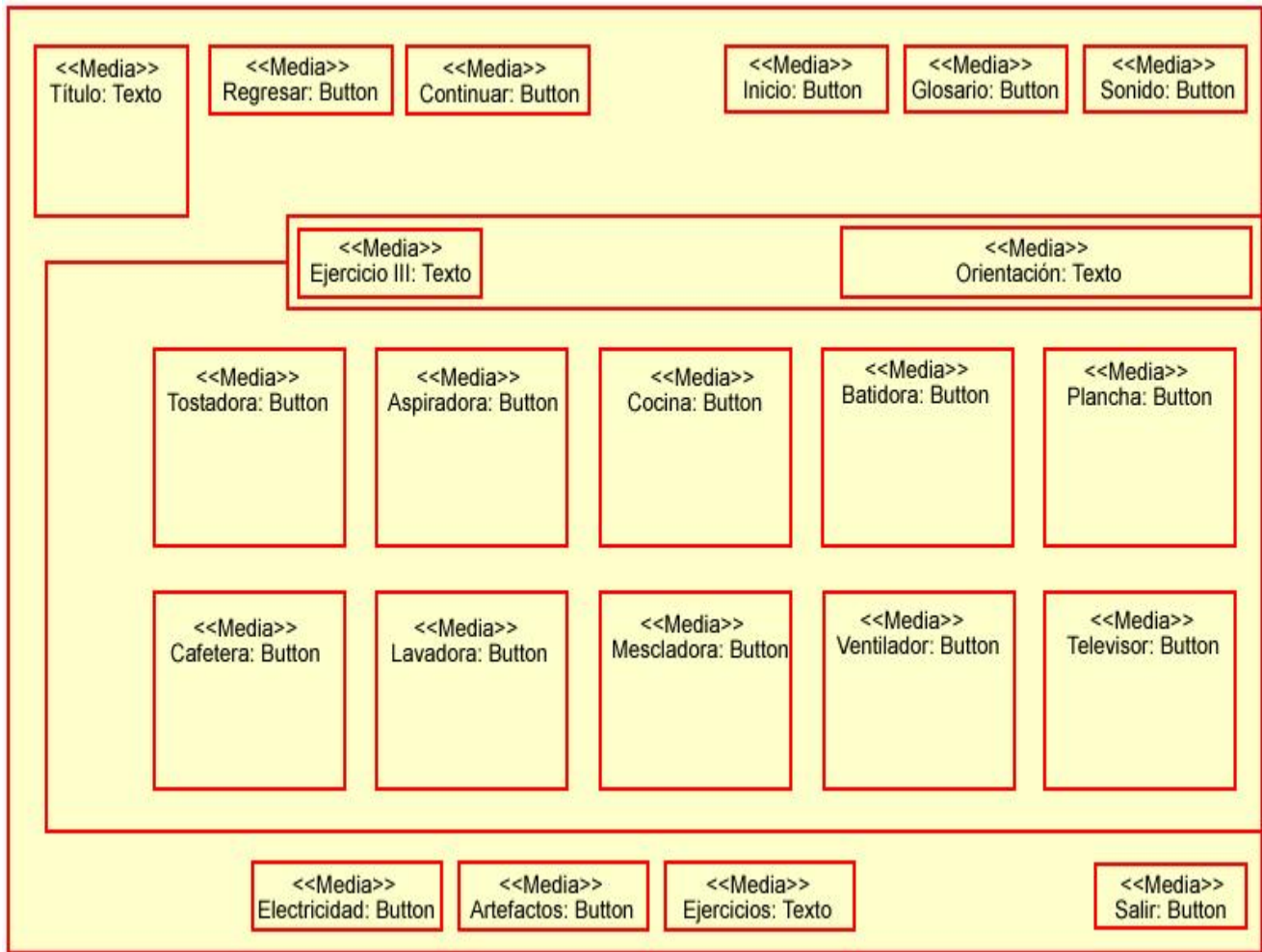


Figura 4. 10 Diagrama de Presentación para la tercera pantalla del Módulo III

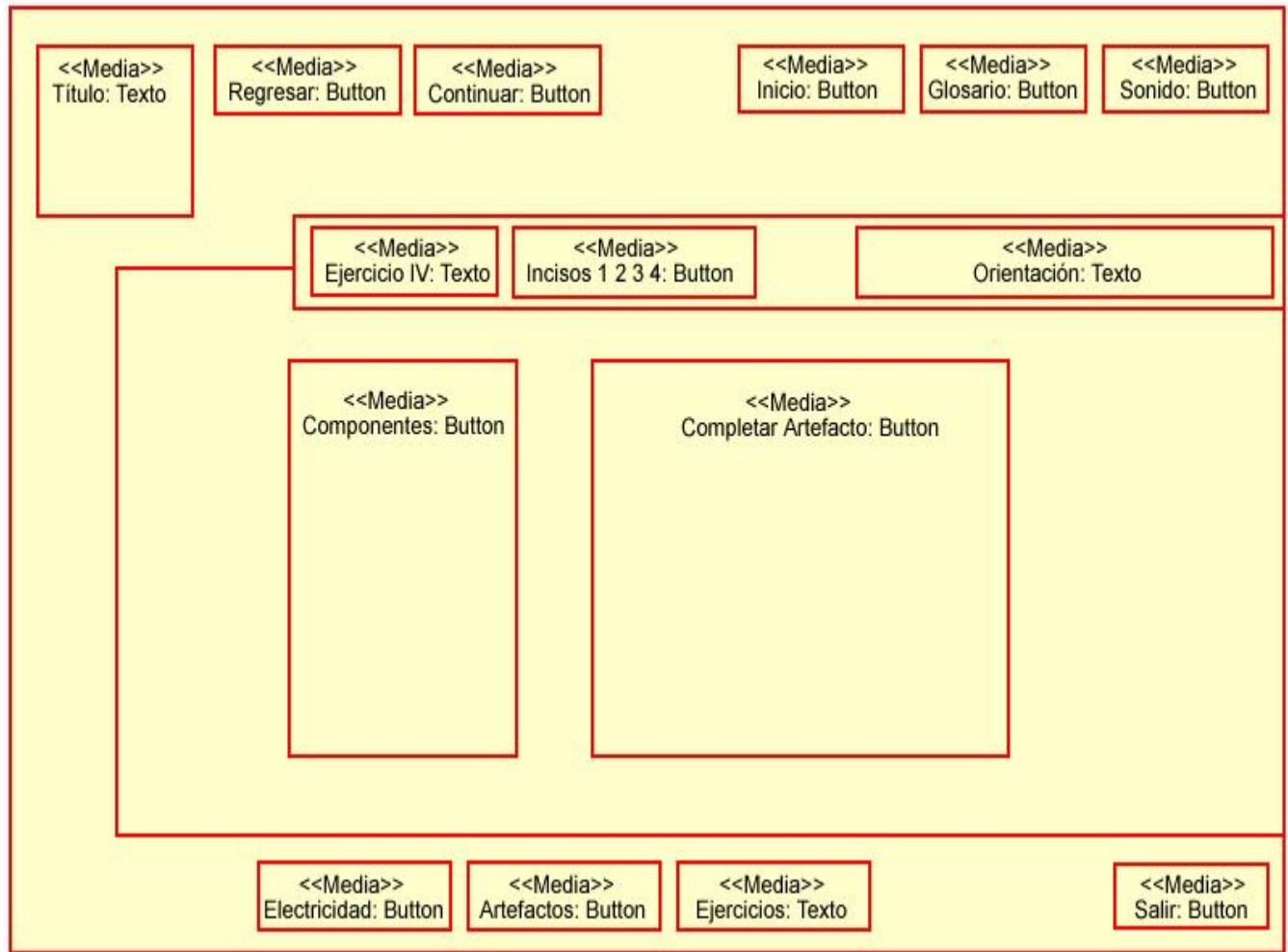


Figura 4. 11 Diagrama de Presentación para la cuarta pantalla del Módulo III

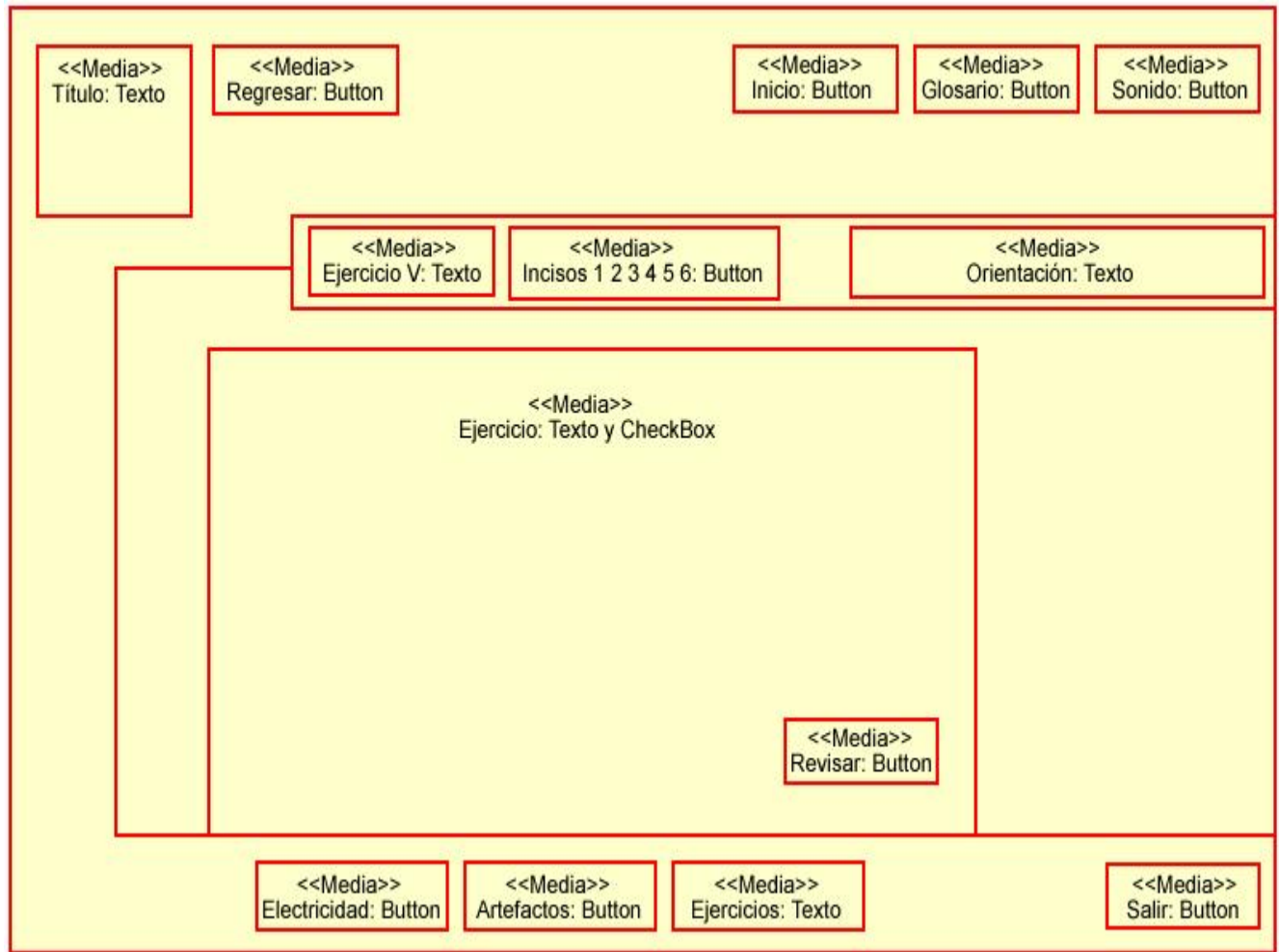


Figura 4. 12 Diagrama de Presentación para la quinta pantalla del Módulo III

4.2.2 Diagrama de componentes.

La declaración de cada uno de los módulos de ejecución se representó en este diagrama, en analogía con la estructura que se le fue moldeando desde el diagrama de navegación en la fase de análisis.

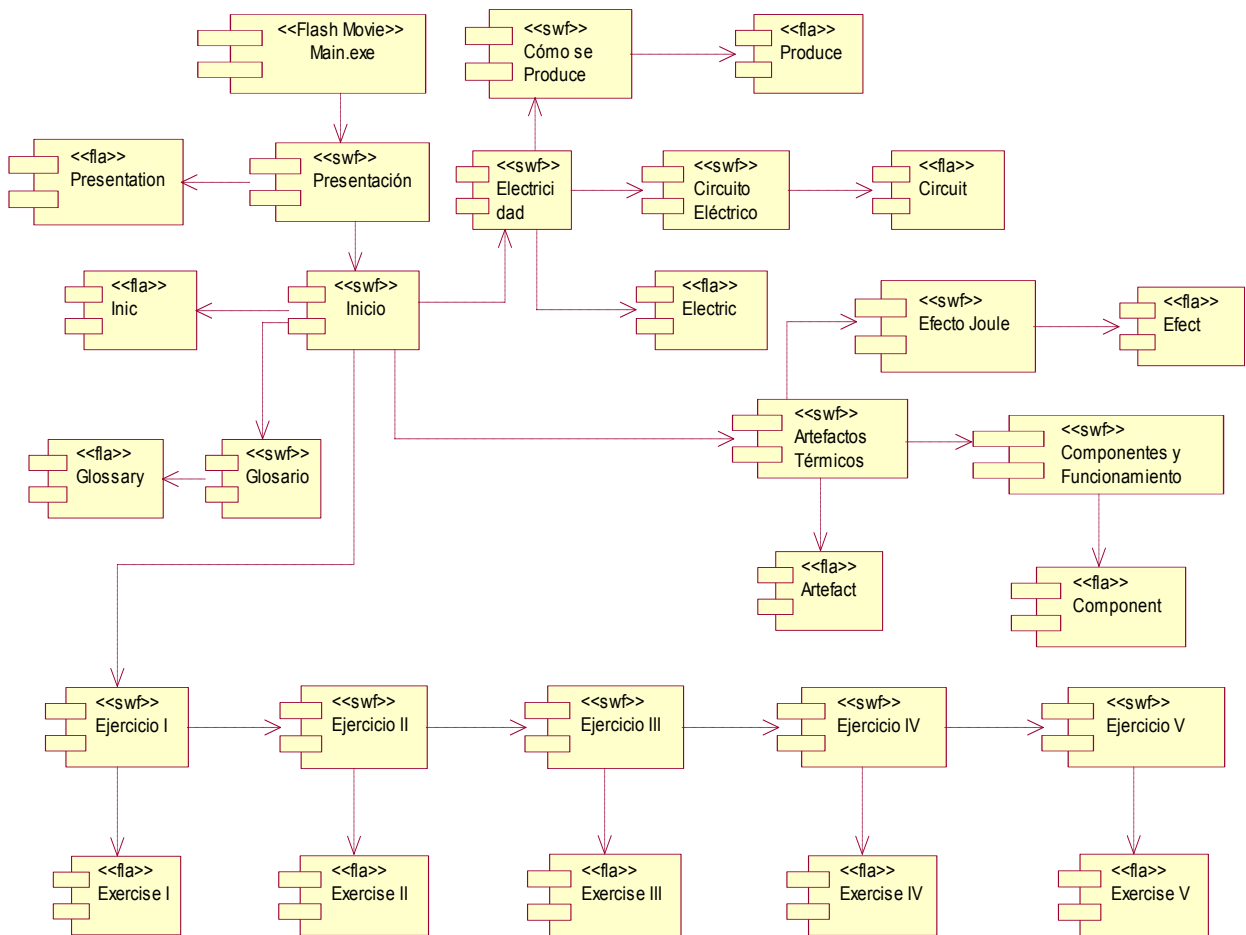


Figura 4. 13 Diagrama de Componentes.

4.3 Conclusiones.

En este capítulo se ha desarrollado el flujo de diseño con el Diagrama de Presentación y el Diagrama de Componentes utilizando OMMMA-L. Se analizaron los principios de diseño de la interfaz de la multimedia y los estándares de codificación a emplear durante la fase de construcción. Concluye así la modelación de la plantilla.

Capítulo 5 Estudio de factibilidad.

5.1 Introducción.

Para un buen diseño de un software no solo debe tenerse en cuenta el problema a resolver, la información de que se dispone, la preparación de los medios a utilizar y las potencialidades informáticas para su implementación, si no se tienen presente los gastos tanto en recursos humanos como en tiempo de desarrollo. Si no se realiza una adecuada planificación y evalúan los costos, la implementación del producto no culminará a tiempo y se utilizarán más recursos que los necesarios. Es por eso que se hace necesario el estudio de la factibilidad.

5.2 Planificación.

La estimación mediante el análisis de Puntos de Casos de Uso se trata de un método de estimación del tiempo de desarrollo de un proyecto mediante la asignación de "pesos" a un cierto número de factores que lo afectan, para finalmente, contabilizar el tiempo total estimado para el proyecto a partir de esos factores.

5.3 Costos.

5.3.1 Cálculo de puntos de Casos de Uso sin Ajustar.

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

$$UUCP = UAW + UUCW$$

Donde,

- **UUCP:** Puntos de Casos de Uso sin Ajustar.

- **UAW:** Factor de Peso de los Actores sin Ajustar.
- **UUCW:** Factor de Peso de los Casos de Uso sin ajustar.

5.3.2 Factor de Peso de los Actores sin ajustar.

Este valor se calcula mediante un análisis de la cantidad de Actores presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Actores se establece teniendo en cuenta en primer lugar si se trata de una persona o de otro sistema, y en segundo lugar, la forma en la que el actor interactúa con el sistema. Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Cantidad de Actores	Factor de Peso
Simple	Otro sistema que interactúa con el sistema a desarrollar mediante una interfaz de programación (API, Application Programming Interface).	0	1
Medio	Otro sistema que interactúa con el sistema a desarrollar mediante un protocolo o una interfaz basada en texto.	0	2
Complejo	Una persona que interactúa con el sistema mediante una interfaz gráfica.	1	3

Tabla 5. 1 Factor de Peso de los Actores sin ajustar.

Por tanto,

$$UAW = 0 * 1 + 0 * 2 + 1 * 3 = 3$$

$$UAW = 3$$

5.3.3 Factor de Peso de los Casos de Uso sin ajustar.

Este valor se calcula mediante un análisis de la cantidad de Casos de Uso presentes en el sistema y la complejidad de cada uno de ellos. La complejidad de los Casos de Uso se establece teniendo en cuenta la cantidad de transacciones efectuadas en el mismo, donde una transacción se entiende como una secuencia de actividades atómica, es decir, se efectúa la secuencia de actividades completa, o no se efectúa ninguna de las actividades de la secuencia. Los criterios se muestran en la siguiente tabla:

Tipo de Actor	Descripción	Cantidad de CU	Factor de Peso
Simple	El caso de uso contiene de 1 a 3 transacciones.	7	5
Medio	El caso de uso contiene de 4 a 7 transacciones.	1	10
Complejo	El caso de uso contiene más de 8 transacciones.	1	15

Tabla 5. 2 Factor de Peso de los Casos de Uso sin ajustar.

Por tanto,

$$UUCW = 7 * 5 + 1 * 10 + 1 * 15$$

$$UUCW = 60$$

Finalmente los Puntos de Casos de Uso sin ajustar resultan.

$$UUCP = UAW + UUCW$$

$$UUCP = 3 + 60$$

UUCP = 63

5.3.4 Cálculo de Puntos de Casos de Uso Ajustados.

Una vez que se tienen los Puntos de Caso de Uso sin ajustar, se debe tomar este valor mediante la siguiente ecuación:

$$UCP = UUCP * TCF * EF$$

Donde,

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

5.3.5 Factor de Complejidad Técnica.

Este coeficiente se calcula mediante la cuantificación de un conjunto de factores que determinan la complejidad técnica del sistema. Cada uno de los factores se cuantifica con un valor de 0 a 5, donde 0 significa un aporte irrelevante y 5 un aporte muy importante. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores:

Tipo de Actor	Descripción	Valor	Factor de Peso
T1	Sistema distribuido.	5	2
T2	Objetivos de performance o tiempo de respuesta.	4	1
T3	Eficiencia del usuario final.	4	1
T4	Procesamiento interno complejo.	3	1
T5	El código debe ser reutilizable.	3	1
T6	Facilidad de instalación.	5	0.5
T7	Facilidad de uso.	5	0.5
T8	Portabilidad.	4	2
T9	Facilidad de cambio.	4	1
T10	Concurrencia.	3	1
T11	Incluye objetivos especiales de seguridad.	0	1
T12	Provee acceso directo a terceras partes.	0	1
T13	Se requieren facilidades especiales de entrenamiento a usuarios.	0	1

Tabla 5. 3 Factor de Complejidad Técnica.

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

$$TCF = 0.6 + 0.01 * \Sigma (\text{Pesoi} * \text{Valori})$$

$$TCF = 0.6 + 0.01 * (10 + 4 + 4 + 3 + 3 + 2.5 + 2.5 + 8 + 4 + 3 + 0 + 0 + 0)$$

$$TCF = 0.6 + 0.01 * 44$$

$$TCF = 1.04$$

5.3.6 Factor Ambiente.

Las habilidades y el entrenamiento del grupo involucrado en el desarrollo tienen un gran impacto en las estimaciones de tiempo. Estos factores son los que se contemplan en el cálculo del Factor de ambiente. El cálculo del mismo es similar al cálculo del Factor de complejidad técnica, es decir, se trata de un conjunto de factores que se cuantifican con valores de 0 a 5. En la siguiente tabla se muestra el significado y el peso de cada uno de éstos factores.

Factor	Descripción	Valor	Peso
E1	Familiaridad con el modelo de proyecto utilizado.	4	1.5
E2	Experiencia en la aplicación.	5	0.5
E3	Eficiencia en la orientación a Objetos.	4	1
E4	Capacidad del Analista Líder.	3	0.5
E5	Motivación.	5	1
E6	Estabilidad de los requerimientos.	3	2
E7	Personal part.-time	1	-1
E8	Dificultad del lenguaje de programación.	3	-1

Tabla 5. 4 Factor Ambiente.

El Factor de ambiente se calcula mediante la siguiente ecuación:

$$EF = 1.4 - 0.03 * \sum (\text{Pesoi} \times \text{Valori})$$

$$EF = 1.4 - 0.03 * (6 + 2.5 + 4 + 1.5 + 5 + 6 - 1 - 3)$$

$$EF = 0.77$$

Por lo tanto con los valores correspondientes:

$$UCP = UUCP * TCF * EF$$

Donde,

- **UCP** = ?
- **UUCP** = 63
- **TCF** = 1.04
- **EF** = 0.77

$$UCP = 63 * 1.04 * 0.77$$

$$UCP = 50.45$$

5.3.7 De los puntos de casos de uso a la estimación del esfuerzo.

Karner originalmente sugirió que cada Punto de Casos de Uso requiere 20 horas-hombre. Posteriormente, surgieron otros refinamientos que proponen una granularidad algo más fina, según el siguiente criterio:

- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por debajo del valor medio (3), para los factores **E1** a **E6**.
- Se contabilizan cuántos factores de los que afectan al Factor de ambiente están por encima del valor medio (3), para los factores **E7** y **E8**.
- Si el total es 2 o menos, se utiliza el factor de conversión 20 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 20 horas-hombre.
- Si el total es 3 o 4, se utiliza el factor de conversión 28 horas-hombre/Punto de Casos de Uso, es decir, un Punto de Caso de Uso toma 28 horas-hombre.

- Si el total es mayor o igual que 5, se recomienda efectuar cambios en el proyecto, ya que se considera que el riesgo de fracaso del mismo es demasiado alto.

El esfuerzo en horas-hombre viene dado por:

$$E = UCP * CF$$

Donde,

E: esfuerzo estimado en horas-hombre

UCP: Puntos de Casos de Uso ajustados

CF: factor de conversión

$$UCP = 50.45$$

$$CF = 20$$

$$E = 50.45 * 20 = 1009 \text{ Horas / Hombre}$$

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

Finalmente, para una estimación más completa de la duración total del proyecto, hay que agregar a la estimación del esfuerzo obtenida por los Puntos de Casos de Uso, las estimaciones de esfuerzo de las demás actividades relacionadas con el desarrollo de software. Para ello se puede tener en cuenta el siguiente criterio, que estadísticamente se considera aceptable. El criterio plantea la distribución del esfuerzo entre las diferentes actividades de un proyecto, según la siguiente aproximación:

Actividad	Porcentaje	Horas/Hombre
Análisis	10 %	126,13
Diseño	20 %	252,25
Programación	40 %	504,5
Pruebas	15 %	189,19
Sobrecarga (otras actividades)	15 %	189,19
Total	100 %	1261,25

Tabla 5. 5 Distribución del Esfuerzo.

Finalmente se obtiene la cantidad de horas por hombre, como en este proyecto trabajan dos hombres la cantidad de horas trabajada por cada uno es de 1261,25 horas.

Como la jornada laboral de un día de trabajo es de 8 horas y se trabajan en un mes aproximadamente de 24 a 25 días de este se puede calcular que el tiempo de duración del proyecto es de 7,3 meses de trabajo.

Para determinar el salario se debe tener en cuenta que los desarrolladores pueden ser ingenieros recién graduados de la UCI (Universidad de las Ciencias Informáticas) y su salario básico es \$ 225.00.

Como trabajan dos hombres en este proyecto entonces el costo es:

$$\text{Costo} = 7.3 * \$ 225 * 2$$

$$\text{Costo} = \$ 3285$$

5.4 Beneficios Tangibles e Intangibles.

Artefactos Electrodomésticos Vol. 1 permite socializar los conocimientos. Con este software se puede llevar la información a gran escala a todas las personas que quieran aprender cuáles son los tipos de electrodomésticos de uso frecuente en el hogar, cómo funciona y cuáles son sus componentes, a fin de posteriormente en el Vol. 2, aprendan cómo realizar mantenimiento y reparaciones básicas a esos aparatos electrodomésticos.

Permitirá incentivar a las comunidades a participar en la conservación y mantenimiento de los artefactos electrodomésticos. También se logrará un aumento considerable en la comprensión del contenido al tener la posibilidad de contar con imágenes y animaciones.

5.5 Análisis de Costos y Beneficios.

El desarrollo de este sistema no supone grandes gastos de recursos. La fácil utilización y navegabilidad el Software no genera daño físico alguno a los manipuladores de este, ni a los dispositivos utilizados para su uso. La interfaz está diseñada cuidadosamente y resultará agradable al entorno del usuario.

En el caso de Artefactos Electrodomésticos Vol. 1, al ser una multimedia a desarrollar en el marco de la Universidad de las Ciencias Informáticas, no se hace necesario calcular el costo monetario, aunque se realizó dando como resultado un costo de \$3285 y un tiempo de desarrollo de 7,3 meses; pues la elaboración del sistema no supone gastos adicionales por concepto de salario para la institución, ya que cuenta con el capital humano necesario (estudiantes y profesores vinculados al proyecto) y no necesita contratar otro personal. Por otra parte la universidad cuenta con la infraestructura tecnológica necesaria para llevar a cabo el proyecto.

5.6 Conclusiones.

Con el desarrollo de este capítulo se expresa claramente la ventaja que implica la implementación de esta plantilla que permite ahorrar recursos humanos, tiempo de desarrollo, así como la necesidad de superación de las personas en las diferentes comunidades venezolanas. Se concluye que es factible el desarrollo de esta aplicación, por todos los aportes sociales que genera.

Conclusiones Generales.

En este trabajo se llegó a la conclusión de la necesidad de diseñar este sistema debido a la importancia de incrementar la cultura acerca de los principales Artefactos Electrodomésticos del hogar, sus componentes y su funcionamiento, necesidades que fueron descritas en el Capítulo I. Utilizando para ello un mecanismo de fácil manipulación y que pueda brindar al estudiante un método para facilitar el aprendizaje. Por lo cual se optó por la creación de una multimedia clasificada como tutorial informativa que muestre la información de manera amena y de fácil navegación.

Se llegó a la conclusión también de que las metodologías idóneas para llevar a cabo el proceso de desarrollo es RUP (Proceso Unificado de Desarrollo de Software) y dentro de ella OMMMA-L como una extensión, la cual permite la modelación del proceso de desarrollo de un software multimedia.

Se definieron los requerimientos del sistema, tanto funcionales como no funcionales, y posteriormente se estructuró el modelo de casos de uso del sistema, describiéndose cada caso de uso para una mejor comprensión de la funcionalidad que brindan.

Finalmente, se analizaron los beneficios y costos que genera el sistema, se compararon y se concluyó que el proyecto es factible. En este trabajo se demostró la importancia y facilidades que brindará este sistema para el aprendizaje sobre los equipos electrodomésticos más importantes, haciendo uso de las tecnologías que se han puesto en nuestras manos. Permitiendo a su vez interactuar de forma amena y sencilla con toda la información que pondrá a disposición en esta multimedia. Esperando que cumpla con su objetivo, en el mejoramiento de su cultura integral y formación.

Recomendaciones.

Se recomienda implementar la multimedia para valorar los resultados a obtener.

Se recomienda hacer un mejor estudio de OMMMA-L como alternativa para el modelado de multimedia y su posible aplicación a las clases de ingeniería de software en las facultades 8 y 9 ya que su segundo aborda esta materia.

Se recomienda hacer un estudio sobre los posibles cambios de Sistema Operativos en las escuelas, talleres, laboratorios e infocentros del país, en caso de que sea necesario hacer la plantilla con alguna alternativa libre.

Referencias Bibliográficas.

- [1] Cabrero Almeida, Julio. Nuevas Tecnologías, Comunicación y Educación.
[<http://tecnologiaedu.us.es/bibliovir/pdf/3.pdf>]
- [2] Díaz – Antón, María Gabriela. Propuesta de una metodología de desarrollo de software educativo bajo un enfoque de calidad sistémica.
[<http://www.infedu.coord.usb.ve/proyectos/proyecto3.html>]
- [3] Engels, Gregor. UML-based Behavior. Specification of Interactive Multimedia Applications.
[<http://wwwcs.upb.de/cs/ag-engels/Papers/2001/SauerHCC01.pdf>]
- [4] Sauer, Stefan-Engels Gregor. MVC-Based Modeling Support for Embedded Real-Time Systems.
- [5] [http://es.wikipedia.org/wiki/Modelo_Vista_Controlador]
- [6] Sauer, Stefan. Extending UML for Modeling of Multimedia Applications.
[<http://www.itec.uni-klu.ac.at/~harald/proseminar02/sauer1.pdf>.]
- [7] [<http://es.wikipedia.org/wiki/POO>]
- [8] [<http://es.wikipedia.org/wiki/RUP>]
- [9] [<http://es.wikipedia.org/wiki/UML>]
- [10] [<http://es.wikipedia.org/wiki/Hipermedia>]
- [11] [<http://es.wikipedia.org/wiki/Multimedia>]
- [12] [<http://es.wikipedia.org/wiki/MVC>]
- [13] M. Gómez, Leopoldo Sebastián. Diseño de Interfaces de Usuario Principios y Heurísticas para Evaluación.
[<http://www.ilustrados.com/publicaciones/EpZVVZkAVFDnjetyop.php>]

Bibliografía.

- [1]. O y Domínguez, Cabrita. Reparación de Artefactos Electrodomésticos. Caracas, Venezuela, Editorial Triángulo, P (1995).
- [2]. Gershon, Wheeler. Reparación de Electrodomésticos. Barcelona, España, Boixareu Editores Marcombo S. A., J (1976).
- [3]. Cenit. Curso Práctico de Electricidad Instalaciones- Reparaciones – Proyectos. Bogotá, Colombia, Compañía Editorial Electrónica, (2000).
- [4]. Pérez Aquiles. Guía de reparación de artefactos electrodomésticos. Caracas, Venezuela, (2000-2003).
- [5]. Edissa. Electricidad y Energía. Enciclopedia Práctica. Bogotá, Colombia, Editado por EDISSA LTDA. (1987).
- [6]. Isakowitz, Tomás, Stohr, Edward y A. Balasubramanian, P. RMM: Metodología para el Diseño Estructurado de Hipermédios.
- [7]. Isakowitz, T., Stohr, E. A. and Balasubramanian, P. RMM: a methodology for structured hypermedia design. Comm. of the ACM, 38(8), 34-44.
- [8]. Valdés Jiménez, Sasha. La Constitución Venezolana. Volumen 2. Universidad de las Ciencias Informáticas, Facultad 1. Ciudad de la Habana, 2006. 114 p.
- [9]. [<http://www.wikilearning.com>]
- [10]. Apoyo a la Educación, La Educación a Distancia.
[[http:// www.sld.cu/libros/distancia/cap1.html](http://www.sld.cu/libros/distancia/cap1.html)]

- [11]. Malberti, María Alejandra y Valenzuela, Francisca Adriana. La Hipermedia: Una Estrategia Didáctica para la incorporación de la geología en los niveles E. G. B. 3 y Polimodal. [http://colos.fcu.um.es/TICEC05/TICEC05/28_502.pdf]
- [12]. Pastor, Juan Antonio y Saorín, Tomás. La Escritura Hipermedia. [<http://www.ucm.es/info/multidoc/multidoc/revista/cuad6-7/saorin.htm>]
- [13]. [<http://www.macromedia.com/it/software/flash/flashpro/productinfo/features>]
- [14]. Carneros, Rocío. Macromedia Shockwave Player. [<http://www.globaliza.net/default.asp?idprod=1434&idcatpad=1303&lang=es>]
- [15]. [<http://www.globalechange.com/files/Aprendizaje%20a%20Distancia%20Articulo%20del%20Microfinance%20Experience%20Series.pdf>]
- [16]. Aprendizaje a Distancia: una nueva experiencia, 1998. [<http://www.unesco.org/iiep/spa/newsletter/1998/jans98.pdf>]
- [17]. [http://www.uhu.es/agora/digital/numeros/01/01-articulos/miscelanea/tirado_flores.PDF]
- [18]. Arroyo Valera, Silvia Regina y Rodríguez Rodríguez, José. El uso de las nuevas tecnologías educativas: la teleformación. [http://www.ieev.uma.es/edutec97/edu97_c3/2-3-15.htm]
- [19] Prevención Automatizada de Errores. [<http://www.synspace.com/ES/Services/tcc.html>]
- [20] Ing. Dávila, Abraham. Estándares para la construcción de software, Agosto del 2001. [http://gidis.inf.pucp.edu.pe/estandares/ManEst_GUI.v.1.0.pdf]
- [21] Modelo Vista Controlador. [http://es.wikipedia.org/wiki/Modelo_Vista_Controlador]

Glosario de Términos

Pantalla: Es la agrupación visual de elementos de medias contenidas en una vista determinada.

RUP: El Proceso Unificado Racional o RUP (Rational Unified Process), es un proceso de desarrollo de software y junto con el Lenguaje Unificado de Modelado UML, constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. Se caracteriza por ser iterativo e incremental, estar centrado en la arquitectura y guiado por los casos de uso. Incluye artefactos (que son los productos tangibles del proceso como por ejemplo, el modelo de casos de uso, el código fuente, etc.) y roles (papel que desempeña una persona en un determinado momento, una persona puede desempeñar distintos roles a lo largo del proceso). [8]

UML: Es un lenguaje gráfico para visualizar, especificar, construir y documentar un sistema de software. El UML ofrece un estándar para escribir un "plano" del sistema, incluyendo aspectos conceptuales tales como procesos de negocios y funciones del sistema, y aspectos concretos como expresiones de lenguajes de programación, esquemas de bases de datos y componentes de software reutilizables, es un "lenguaje" para especificar y no un método o un proceso. El UML se usa para definir un sistema de software; para detallar los artefactos en el sistema, para documentar y construir. El UML se puede usar en una gran variedad de formas para soportar una metodología de desarrollo de software (tal como el Proceso Unificado de Rational) -pero no especifica en sí mismo qué metodología o proceso usar. [9]

OMMMA-L: El Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia es una extensión de UML especializado en aplicaciones multimedia. (ver explicación en el capítulo 2 "Tendencias y Tecnología").

Multimedia: Es un sistema que utiliza más de un medio de comunicación al mismo tiempo en la presentación de la información, como texto, imagen, animación, vídeo y sonido.[11]

Hipermedia: Es un término usado como lógica extensión del término Hipertexto, en el cual audio, video, texto e hipervínculos generalmente no secuenciales, se entrelazan para formar un continuo

de información, que puede considerarse como virtualmente infinito desde la perspectiva de Internet. [10]

Hipertexto: Un hipertexto es un documento digital o no, que se puede leer de manera no secuencial. Un hipertexto tiene los siguientes elementos: secciones, enlaces o hipervínculos y anclajes. Las secciones o nodos son los componentes del hipertexto o hiperdocumento. Los enlaces son las uniones entre nodos que facilitan la lectura secuencial o no secuencial del documento. Los anclajes son los puntos de activación de los enlaces. [11]

MVC: es un patrón de diseño de software que separa los datos de una aplicación, la interfaz de usuario, y la lógica de control en tres componentes distintos de forma que las modificaciones al componente de la vista pueden ser hechas con un mínimo impacto en el componente del modelo de datos. [12]