

**Universidad de las Ciencias Informáticas
Facultad 9**



LA METODOLOGÍA RMM (RELATIONSHIP MANAGEMENT METHODOLOGY) APLICABLE AL DESARROLLO DEL SOFTWARE EDUCATIVO MULTIMEDIA TOPOGRAFÍA

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE INGENIERO EN CIENCIAS
INFORMÁTICAS**

AUTORES: Yurelisis Pérez Rodríguez

Jorquiel Sánchez Torres

TUTOR: Ing. Yaneisis Pérez Heredia

**Ciudad de la Habana, junio 28 | 2007
"Año 49 de la Revolución"**

“Hoy, el reto de la educación es aprovechar los nuevos medios - video, audio, computadores, inteligencia artificial, realidad virtual, etc. - para atraer al estudiante y permitirle la construcción de su conocimiento.”

Pedro Salcedo Lagos

DECLARACIÓN DE AUTORÍA

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

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

Autor(es): Yurelixis Pérez Rodríguez

Jorquiel Sánchez Torres

Firma del Autor

Firma del Autor

Tutor: Ing. Yaneisis Pérez Heredia

Firma del Tutor

AGRADECIMIENTOS

A todas las personas que nos han ayudado a lo largo del camino para poder llegar hasta aquí, a los profesores y los compañeros de grupo que hemos vivido juntos todos estos años de universidad.

A Yaneisis nuestra tutora por su ayuda en todo momento.

A Fidel y la revolución por darnos la oportunidad de ser parte de este hermoso proyecto que no solo es un sueño de nuestro Comandante sino que ha llegado a ser la vida de muchos de nosotros.

Yurelxis:

A toda mi familia que siempre me apoyó, en especial a mis padres y a mi hermana porque son la luz de mi vida, a mi novio por apoyarme.

Agradezco a todos mis compañeros del proyecto UCITeVe, a mis amistades que el transcurso de la vida compartimos momentos inolvidables.

Jorquiel:

A toda mi familia que siempre me apoya, a mis abuelos, mis padres, mi esposa e hijo pero muy especialmente a ese gran abuelo que tengo, que es mi apoyo, mi guía y mi ejemplo a seguir en la vida.

Agradezco a todos mis compañeros, los tantos que han compartido inolvidables momentos de mi vida de estudiante, los del proyecto UCITeVe y sobre todo a esos grandes amigos de universidad Campbell, Blanco, Camejo, Armando y de forma especial a Ramón que siempre me brindó su ayuda en todo momento.

DEDICATORIA

Yurelxis:

A toda mi familia y mis amistades que siempre estuvieron para apoyarme en los tiempos difíciles.
A mi madre por siempre acogerme en sus brazos y siempre estar a mi lado en los momentos que más la necesité, por ser mi mejor amiga y darme tanto amor.
A mi padre por quererme tanto y guiarme en vida, por estar siempre a mi lado apoyándome.
A mi hermana por todo su cariño.
A mi novio por hacerme tan feliz.

Jorquiel:

A toda mi familia y amigos por apoyarme en todo momento.
A mis padres que los quiero muchísimo y siempre están pendientes de mis resultados.
A mis abuelos que son muy especiales en mi vida.
A mi hermano que espero poder servirle de ejemplo.
A mi esposa por ser tan especial.
Y a mi pequeño hijo que es mi vida.

OPINIÓN DEL TUTOR

Como tutora del Trabajo de Diploma “La Metodología RMM aplicable al desarrollo del Software Educativo Multimedia Topografía”, luego de haber culminado la realización del mismo, considero que los autores Jorquiel Sánchez Torres y Yurelixis Pérez Rodríguez, han desarrollado un conjunto de habilidades que les permitirán darle solución adecuadamente a cualquier tipo de necesidad de informatización que se presente en su vida profesional.

Durante la realización del presente trabajo las estudiantes han demostrado un alto grado de responsabilidad ante el cumplimiento en tiempo de las tareas que se les programaron. Han trabajado organizadamente dando muestras de poseer responsabilidad y compromiso en la realización de su tesis. Su desempeño ha manifestado que han desarrollado un valioso nivel de asimilación de nuevas metodologías, llegando a alcanzar un profundo conocimiento y una gran capacidad para la toma de decisiones correctas.

La originalidad, la elegancia en el trabajo y la independencia, han sido cualidades dignas de destacar a lo largo de la realización del trabajo realizado. Los estudiantes manifestaron laboriosidad a lo largo del cumplimiento de las tareas programadas, logrando resultados satisfactorios, realizando una publicación del mismo en el Instituto Superior Politécnico José Antonia Echevarría.

Por otra parte, el elemento investigativo del documento, estuvo desde el inicio muy bien orientado y estructurado, basado en una gran cantidad de bibliografía actualizada. Cada contenido se ha expuesto con claridad y aporta grandes conocimientos al lector.

Por todo lo anteriormente planteado, considero que los diplomantes están aptos para ejercer como Ingenieros Informáticos; y propongo al tribunal que se le otorgue al Trabajo de Diploma la máxima calificación.

RESUMEN

Con el aumento en los últimos años del desarrollo de software educativo (SWE) y la utilización de estos como apoyo al proceso de enseñanza-aprendizaje, se investiga sobre una de las metodologías que ayuden en el proceso de desarrollo de los mismos y se aplica en un caso de estudio.

Actualmente en nuestro país existen varios proyectos donde se desarrollan SWE y aunque en las escuelas cubanas se utilizan en el proceso de enseñanza aprendizaje estos productos, se presentan dificultades en su proceso de desarrollo y en muchas ocasiones no se obtiene el producto con la calidad requerida una de las causas es debido fundamentalmente a que no se utiliza una metodología que guíe este proceso.

Teniendo en cuenta la situación anterior, los autores de este trabajo de diploma buscan una solución al problema realizando una profunda investigación de la Metodología RMM (Relationship Management Methodology) y aplicando los conocimientos obtenidos de la metodología en el análisis y diseño particularmente en el caso de estudio Multimedia Topografía, logrando obtener resultados favorables.

El trabajo se estructura de la manera siguiente: una introducción, tres capítulos, conclusiones, recomendaciones, bibliografía, glosario y anexos.

La introducción da al lector una idea general del contenido del trabajo, la situación problemática, el problema que se pretende resolver, el objetivo general del trabajo y las tareas que se realizaron para lograrlo.

El primer capítulo es la fundamentación teórica donde se exponen los conceptos y aspectos fundamentales para la comprensión del posterior desarrollo del trabajo.

El segundo capítulo es el estudio de la metodología RMM, su ciclo de vida, que se hace en cada una de las siete etapas, limitaciones y propuestas de mejoras.

El tercer capítulo es la aplicación de la metodología RMM al caso de estudio Multimedia Topografía para su mejor comprensión y se centra principalmente en las etapas de diseño. Al final se plasman los resultados obtenidos de aplicar la metodología.

Se dan unas conclusiones generales sobre el trabajo realizado y se realizan algunas recomendaciones.

PALABRAS CLAVES:

Metodología RMM, software educativo.

ÍNDICE

AGRADECIMIENTOS	III
DEDICATORIA	IV
RESUMEN	VI
INTRODUCCIÓN	1
CAPÍTULO 1	4
1.1 Introducción	4
1.2 Principales Conceptos	4
1.2.1 Software	4
1.2.2 Ingeniería de Software.	5
1.2.3 Proceso de Ingeniería de Software.	5
1.2.4 Proceso de Ingeniería de Software Educativo.	6
1.3 Metodologías de desarrollo de software.	7
1.3.1 Características de las metodologías de desarrollo.	9
1.3.2 ¿Qué hay que saber para construir o elegir una metodología?	10
1.4 El Software Educativo.	13
1.4.1 Tipologías del software	13
1.4.2 Concepto de Software Educativo.	14
1.4.3 Clasificaciones de los software educativos.	15
1.4.4 Funciones del software educativos.	22
1.4.5 Ventajas de los software educativos.	22
1.4.6 Desventajas de los software educativos.	23
1.5 El uso del software educativo en Cuba. Su inserción en el proceso pedagógico.	23
1.6 Conclusiones	24
CAPÍTULO 2	25
2.1 Introducción	25
2.2 Modelo de referencia.	25
2.2.1 Modelo de Hipertexto de Dexter.....	25
2.3 Modelo con semántica de navegación formalmente definida.	27
2.3.1 Modelo de hipermedia de Ámsterdam.	27
2.4 Modelo de representación relacional.	28
2.4.1 El modelo HDM	28
2.4.2 Esquema HDM	31
2.5 RMM (Relationship Management Methodology).	32
2.5.1 Las Primitivas de Datos y de Acceso.	35
2.5.2 Obtención del diagrama RMDM.	38
2.5.3 Limitaciones que presenta originalmente RMM.	39
2.5.4 Propuestas de mejoras a RMM Original.	39
2.5.5 Otras propuestas de mejoras para la Metodología RMM.....	40
2.6 Etapas de la Metodología RMM.	41
2.6.1 Etapa 1: Diseño del Modelo Entidad-Relación.....	41
2.6.2 Etapa 2: Diseño de slices.....	42
2.6.3 Etapa 3: Diseño de la Navegación.	43
2.6.4 Etapa 4: Definir el Protocolo de Conversión.	44
2.6.5 Etapa 5: Diseño de Interfaz de Usuario	44
2.6.6 Etapa 6: Diseño del Comportamiento en Ejecución.....	45
2.6.7 Etapa 7: Construcción y Prueba.....	45
2.7 Análisis Crítico del modelo RMDM.	45
2.8 Herramienta RMCase soporte de la Metodología RMM.	45
2.8.1 Diseño E-R	45
2.8.2 Modelo E-R+ (Diseño de slices).....	48
2.8.3 Modelo de navegación	50
2.8.4 Prototipado	53

2.9	Evaluación de RMCCase como herramienta de software.	54
2.9.1	Utilidad:	54
2.9.2	Coste	56
2.9.3	Compatibilidad	56
2.9.4	Fiabilidad	57
2.10	Conclusiones	57
CAPÍTULO 3		58
3.1	Introducción	58
3.2	Caso de estudio.	58
3.3	Aplicación de la metodología RMM al caso de estudio Multimedia Topografía.	58
3.3.1	Etapa 1: Diseño del Modelo Entidad-Relación (E-R).	59
3.3.2	Etapa 2: Diseño de los Slice.	62
3.3.3	Etapa 3: Diseñar la Navegación.	64
3.3.4	Etapa 4: Definir el Protocolo de Conversión.	68
3.3.5	Etapa 5: Diseño de Interfaz de Usuario.	68
3.3.6	Etapa 6: Diseño del Comportamiento en Ejecución.	72
3.3.7	Etapa 7: Construcción y Prueba.	72
3.4	Resultados de aplicar la metodología RMM.	73
3.5	Conclusiones	74
	Conclusiones	75
	Recomendaciones	76
	Bibliografía	77
	Anexos	78
	Glosario	82

ÍNDICE DE FIGURAS

Figura 1. La ingeniería de software es una tecnología multicapas.	5
Figura 2. La tres capas del modelo de Dexter.	27
Figura 3. La metodología RMM en el contexto del ciclo de vida del software.	33
Figura 4. Representación gráfica de una entidad	35
Figura 5. Representación gráfica de un atributo.	35
Figura 6. Representación gráfica de un slice.	35
Figura 7. Representación gráfica de un índice condicional.	36
Figura 8. Representación gráfica de una visita guiada condicional.	36
Figura 9. Representación gráfica de una visita guiada indexada condicional.	36
Figura 10. Representación gráfica de una primitiva de grupo o menú.	37
Figura 11. Representación gráfica de un enlace.....	37
Figura 12. Representación de las relaciones de asociación muchos a muchos en el diagrama E-R.	41
Figura 13. Representación de los enlaces estructurales entre slice en la metodología RMM.	43
Figura 14. Edición de la estructura de navegación "Escribe".....	51
Figura 15. Edición de la condición en la estructura de navegación "Escribe".	52
Figura 16. Diagrama E-R del SWE Multimedia Topografía.....	61
Figura 17. La estructura de menú del SWE Multimedia Topografía.	65
Figura 18. Información General es una parte de diagrama RMDM del SWE Multimedia Topografía.	65
Figura 19. Docencia es una parte del diagrama RMDM del SWE Multimedia Topografía.	66
Figura 20. Evaluación es parte del diagrama RMDM del SWE Multimedia Topografía.	67
Figura 21. Galería y Otras Multimedia son partes del diagrama RMDM del SWE Multimedia Topografía.	67
Figura 22. Interfaz de la herramienta RMCCase para el diseño del diagrama E-R.	79
Figura 23. Interfaz de la herramienta RMCCase para el diseño de slice.	79
Figura 24. Interfaz de la herramienta RMCCase para diseñar el diagrama RMDM.....	80
Figura 25. Diseño de una interfaz de usuario con RMCCase.	80
Figura 26. Prototipo de aplicación con RMCCase.	81

ÍNDICE DE TABLAS

Tabla 1. Capas del modelo de Dexter.	25
Tabla 2. Las primitivas de datos y de acceso de la metodología RMM.	38
Tabla 3. División en slice de las entidades del SWE Multimedia Topografía.	62

INTRODUCCIÓN

Utilizar la información como apoyo a los procesos de enseñanza/aprendizaje ha sido una inquietud que durante mucho tiempo ha sido investigada y probada por muchas instituciones y docentes. Su asimilación dentro de instituciones educativas ha aumentado en los últimos años, por lo que la demanda por SWE de calidad es cada vez mayor y para obtener eficacia en el proceso de desarrollo de estos software es muy importante la utilización de alguna de las metodologías.

Existen varias metodologías para el desarrollo de aplicaciones hipermedia que se pueden utilizar para desarrollar SWE, no todas ellas están orientadas al desarrollo del mismo tipo de aplicaciones, en principio existen dos grandes grupos: las que están basadas en un modelo de datos relacional como HDM (Hypermedia Design Model), RMM (Relationship Management Methodology) y RMM extendido; y otras basadas en un modelo de datos orientado a objetos como OOHDM (Object Oriented Hypermedia Design Methodology) y EORM (Enhanced Object Relationship Model); entre otras.

En el presente trabajo se realizó un estudio de la metodología RMM y se aplicó en el desarrollo de SWE Multimedia Topografía, obteniendo un conjunto de resultados favorables.

Nuestro país en los últimos años se ha visto enfrascado en una revolución tecnológica donde la esfera de educación no ha estado exenta en toda una serie de inversiones que ha realizado el estado cubano en equipos y tecnología. Se han instalado en todas las escuelas computadoras y otros equipos para que se haga uso de las Tecnologías de la Informática y las Comunicaciones (TIC) en los procesos de enseñanza-aprendizaje.

A nivel internacional existe un progresivo aumento en el desarrollo de SWE para su utilización en el proceso de enseñanza-aprendizaje donde tanto alumnos como docentes son beneficiados al poder apoyarse en las maravillosas prestaciones que brindan las TIC para su preparación y desarrollo del conocimiento, se utilizan una gran variedad de SWE que hacen que el proceso de enseñanza/aprendizaje sea más ameno y divertido.

En nuestro país el Departamento Nacional de Software Educativo del Ministerio de Educación que forma parte de la Dirección de Computación Educativa tiene dentro de sus funciones dirigir, coordinar, organizar y controlar la producción e introducción en la práctica de SWE para la escuela cubana desde el ámbito de la red de Centros de Estudio de Software Educativo de los Institutos Superiores Pedagógicos, bajo un esquema de investigación-producción.

Se han creado proyectos de desarrollo de SWE en los Institutos Superiores Pedagógicos donde trabajan fundamentalmente especialistas en pedagogía junto a programadores y otros especialistas, con el objetivo de satisfacer las necesidades de SWE para apoyar el proceso de enseñanza-aprendizaje en la escuela cubana.

Como resultado de estos proyectos en los últimos años se han desarrollado más de 78 productos, otros que están desarrollándose y algunos que están en proyecto para un futuro no muy lejano. Nuestras escuelas cuentan con una diversidad de SWE que los alumnos y docentes lo utilizan de forma satisfactoria en su preparación.

Para citar algunos ejemplos, actualmente se trabaja con "Multisaber", colección que tiene un enfoque curricular y multidisciplinario por su relación con los contenidos de los programas de cada asignatura del currículo de estudio de la educación primaria. "El Navegante", colección de 10 software para la educación secundaria. "Futuro", colección de 19 software para la educación preuniversitaria. "Jugar y Aprender", este nuevo software es el resultado de un proyecto de investigación que se lleva a cabo en nuestro país por un equipo de especialistas relacionado con la inclusión de la computación en las edades preescolares.

Además se han desarrollado varios software monotemáticos de diferentes educaciones que son utilizados según el tema específico para el cual fueron diseñados e implementados.

Aunque en nuestras escuelas cubanas se utilizan los productos para el proceso de enseñanza aprendizaje se presentan dificultades en el proceso de desarrollo ya que una de las causas radica en que en estos proyectos no se utiliza una metodología que guíe el análisis y diseño de los SWE, por lo que la principal tarea estaría encaminada a resolver la siguiente problemática:

¿Cómo lograr mayor eficiencia en el desarrollo del software educativo cubano Multimedia Topografía en la CUJAE a través del uso de la metodología RMM?

En este trabajo fue objeto de estudio el proceso de desarrollo de software educativo en Cuba.

Para esto se hizo necesario profundizar en el estudio del campo de acción:

Metodología de desarrollo de software educativo en Cuba.

El objetivo general que se pretende alcanzar con el desarrollo de este trabajo es:

Aplicar en el proceso de desarrollo del software educativo Multimedia Topografía, la Metodología RMM.

Se trabajó en base a la siguiente idea a defender:

La utilización de la metodología RMM en el proceso de desarrollo del software educativo

Multimedia Topografía producirá resultados favorables.

Para dar cumplimiento al objetivo planteado se ha propuesto cumplir las siguientes tareas.

1. Estudio de las metodologías nacionales e internacionales para el desarrollo de software educativos.
2. Estudio de los conceptos fundamentales vinculados a software educativos
3. Estudio de las condiciones actuales en la que se desarrolla el software educativo en Cuba.
4. Estudio de la Metodología RMM.
5. Aplicar la metodología RMM al caso de estudio software educativo Multimedia Topografía.

Se utilizaron en la realización del trabajo de diploma dos métodos de investigación:

- El Teórico.
 - Análisis y Síntesis
 - Histórico-Lógico
 - Hipotético-Deductivo

- El Empírico.
 - Entrevistas
 - Encuestas

CAPÍTULO 1

Fundamentación Teórica

1.1 Introducción

En este primer capítulo del presente trabajo de diploma se hace referencia a los conceptos fundamentales y se describen de forma general los aspectos relacionados con el objeto de estudio y el campo de acción en que se trabaja. Este capítulo es la base teórica para la comprensión del trabajo que se desarrolla y los aspectos fundamentales son el proceso de desarrollo de software educativo, y las metodologías de desarrollo.

1.2 Principales Conceptos

1.2.1 Software

El término «software» fue usado por primera vez en este sentido por John W. Tukey en 1957. En las ciencias de la computación y la ingeniería de software, el software es toda la información procesada por los sistemas informáticos: programas y datos. El concepto de leer diferentes secuencias de instrucciones de la memoria de un dispositivo para controlar cálculos fue inventado por Charles Babbage como parte de su máquina diferencial. La teoría que forma la base de la mayor parte del software moderno fue propuesta por vez primera por Alan Turing en su ensayo de 1936, los números computables, con una aplicación al problema de decisión. [19]

Se denomina software, programática, equipamiento lógico o soporte lógico a todos los componentes intangibles de un ordenador o computadora, es decir, al conjunto de programas y procedimientos necesarios para hacer posible la realización de una tarea específica, en contraposición a los componentes físicos del sistema (hardware). Esto incluye aplicaciones informáticas tales como un procesador de textos, que permite al usuario realizar una tarea, y software de sistema como un sistema operativo, que permite al resto de programas funcionar adecuadamente, facilitando la interacción con los componentes físicos y el resto de aplicaciones.[8] Se puede definir software como «la suma total de los programas de cómputo, procedimientos, reglas documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo».

Bajo esta definición, el concepto de software va más allá de los programas de cómputo en sus distintas formas: código fuente, binario o ejecutable, además de su documentación: es decir, todo lo intangible. [19]

1.2.2 Ingeniería de Software.

La Ingeniería de Software es una tecnología multicapa en la que, según Pressman, se pueden identificar: los métodos (indican cómo construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos) (ver Figura 1). [18]

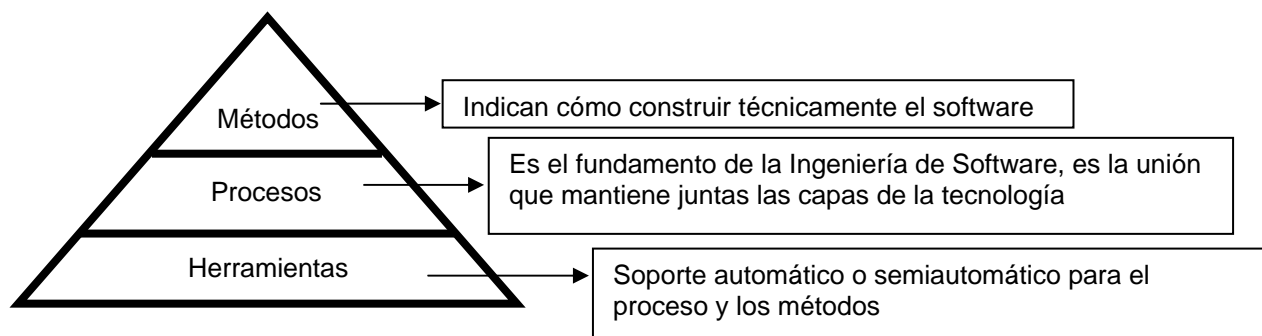


Figura 1. La ingeniería de software es una tecnología multicapas.

Ingeniería de Software es la rama de la ingeniería que aplica los principios de la ciencia de la computación y las matemáticas para lograr soluciones costo-efectivas (eficaces en costo o económicas) a los problemas de desarrollo de software, es decir, permite elaborar consistentemente productos correctos, utilizables y costo-efectivos.

La ingeniería de software está compuesta por una serie de modelos que abarcan los métodos, las herramientas y los procedimientos. Estos modelos se denominan frecuentemente paradigmas de la ingeniería del software y la elección de un paradigma se realiza básicamente de acuerdo al tipo de proyecto y de aplicación, los controles y las entregas a realizar. [3]

1.2.3 Proceso de Ingeniería de Software.

El proceso de ingeniería de software se define como "un conjunto de etapas parcialmente ordenadas con la intención de lograr un objetivo, en este caso, la obtención de un producto de software de calidad". [2]

Un proceso de desarrollo de software es el conjunto de actividades necesarias para transformar los requisitos de un usuario en un sistema de software.

El proceso de desarrollo de software "es aquel en que las necesidades del usuario son traducidas en requerimientos de software, estos requerimientos transformados en diseño y el diseño implementado en código, el código es probado, documentado y certificado para su uso operativo".

Concretamente "define quién está haciendo qué, cuándo hacerlo y cómo alcanzar un cierto objetivo". [2]

El proceso de desarrollo de software requiere de un conjunto de conceptos, una metodología y un lenguaje propio. A este proceso también se le llama el ciclo de vida del software que comprende cuatro grandes fases: concepción, elaboración, construcción y transición. La concepción define el alcance del proyecto y desarrolla un caso de negocio. La elaboración define un plan del proyecto, especifica las características y fundamenta la arquitectura. La construcción crea el producto y la transición transfiere el producto a los usuarios. [18]

En otros documentos se dice que el proceso de construcción de un sistema de software está formado por las siguientes etapas: la obtención de los requisitos, el diseño del sistema, la codificación y las pruebas del sistema. Desde la perspectiva del producto, se parte de una necesidad, se especifican los requisitos, se obtiene el diseño del mismo, el código respectivo y por último el sistema de software.

Algunos autores sostienen que el nombre ciclo de vida ha sido relegado en los últimos años, utilizando en su lugar proceso de software, cambiando la perspectiva de producto a proceso. En todas las etapas del ciclo de vida del software o proceso de software es necesario definir los procesos, las actividades y las tareas a desarrollar.

El proceso de desarrollo de software puede describirse sintéticamente como: la obtención de los requisitos del software, el diseño del sistema de software (diseño preliminar y diseño detallado), la implementación, las pruebas, la instalación, el mantenimiento y la ampliación o actualización del sistema.

A veces se habla de ciclo de desarrollo para denominar al subconjunto del ciclo de vida que empieza en el análisis y finaliza en la entrega del producto. [3]

1.2.4 Proceso de Ingeniería de Software Educativo.

Según Alvaro Galvis de la Universidad de los Andes, es muy coherente y pertinente considerar la producción de SWE dentro de la Ingeniería de software.

¿Qué tan válida es la propuesta de Alvaro Galvis, donde habla de la ingeniería del software, se puede considerar una ingeniería de SWE aparte de la ingeniería del software o es un caso particular donde el análisis del dominio del problema es educativo?

Se considera que la ingeniería del SWE es una adecuación al área de las ciencias informáticas enfocadas a la creación de software que facilite el proceso de aprendizaje-enseñanza como herramienta pedagógica.

En el proceso de ingeniería del SWE se tiene como una de las fundamentales tareas la inclusión de los aspectos pedagógicos, siguiendo las pautas de las teorías educativas y de la comunicación subyacente.

1.3 Metodologías de desarrollo de software.

Las metodologías son el camino a seguir para desarrollar software de una manera sistemática, las metodologías persiguen tres necesidades principales:

- Mejores aplicaciones, conducentes a una mejor calidad.
- Un proceso de desarrollo controlado.
- Un proceso normalizado en una organización, no dependiente del personal.

En cuanto al concepto de metodologías de desarrollo varios autores han tratado el tema:

Se define metodología como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información. [5] Se define una metodología de desarrollo como: "Un conjunto de procedimientos, técnicas, herramientas, y un soporte documental que ayuda a los desarrolladores a realizar nuevo software". [7]

En el proceso de desarrollo de software la metodología define " Quién debe hacer, Qué, Cuándo y Cómo debe hacerlo". La metodología constituye la columna vertebral del proceso de desarrollo de software. No existe una metodología universal. Las características de cada proyecto (equipo de desarrollo, recursos, etc.) exigen que el proceso sea configurable.

Hay que destacar que el proceso de elaboración de software no es un proceso lineal, sino iterativo: en determinados momentos de la realización se comprueba el funcionamiento, el resultado, se evalúa el producto... y frecuentemente se detecta la conveniencia de introducir cambios.

Como dice N. Wirth, creador del lenguaje PASCAL, " la construcción de programas consiste en una secuencia de pasos de perfeccionamiento". Desde otra perspectiva, Jean Michel Lefèvre afirma: "escribir un programa didáctico es como tener una aventura: generalmente se conoce el punto de partida, más o menos se sabe donde se quiere ir, pero se desconoce con exactitud lo que pasará por el camino". [6]

Existen diferentes metodologías para la creación de software, así como específicamente para la arquitectura de información. Entre ellas existen similitudes entre las etapas y los pasos que realizan:

En el mundo:

- *HDM* (Método de Diseño Hipermedia) [Hypermedia Design Method]
- *RMM* (Metodología de Administración de Relaciones) [Relationship Management Methodology]
- *OOHDM* (Metodología de Diseño Hipermedia Orientada a Objetos) [Object-oriented Hypermedia Design Methodology]
- *EORM* (Metodología de Relaciones de Objetos Mejorada) [Enhanced Object Relationship Methodology]
- *SOHDM* (Metodología de Diseño Hipermedia Orientada a Objetos y basada en escenarios) [Scenario-based Object-oriented Hypermedia Design Methodology]
- *WSDM* (Método de Diseño de Sitios Web) [Web Site Design Method]
- *WAE-Proceso Conallen* (Extensión de Aplicación Web para UML) [Web Application Extension for UML – Process Conallen].
- *OOTD* (Object Oriented Text Descomposition) de Talbert y Umphress (1989) para la realización de hipertextos
- *OOTDD* (Object Oriented Top Down Design) actualizada para la realización de hipermedia.
- *OO-Method*, con su adaptación a interfaces hipertextuales OO-H Method como un modelo de estructuración semántica de interfaces Web orientado a objeto.
- *Web ML* (Web Modelling Language) de Ceri, Fraternali y Bongio
- *OOWS* (Object Oriented Web Solutions) basado en el OO-Method
- *W3DT* (World Wide Web Design Technique)
- *OxO* (Prof. Jonás M. y Prof. Beatriz Sandia)
- *Chicago Information Architecture Plan*
- *IBM OVID* (Object, View and Interaction Desing) (Objeto, Vista y Diseño de Interacción)

En Cuba:

- *Metvisual, ADOOSI* (Instituto Superior Politécnico "José Antonio Echeverría" ISPJAE)
- *Multimet* (Centro de Estudio de Ingeniería de Sistema, Instituto Superior Politécnico "José Antonio Echeverría" ISPJAE).
- Metodología de la Empresas CITMATEL por el MS Rafael Barrera

1.3.1 Características de las metodologías de desarrollo.

Cada metodología tiene características particulares pero de forma general se puede enumerar una serie de características [7] que debe tener la metodología y que influirán en el entorno de desarrollo:

- Reglas predefinidas
- Determinación de los pasos del ciclo de vida
- Verificaciones en cada etapa
- Planificación y control
- Comunicación efectiva entre desarrolladores y usuarios
- Flexibilidad: aplicación en un amplio espectro de casos
- De fácil comprensión
- Soporte de herramientas automatizadas
- Que permita definir mediciones que indiquen mejoras
- Que permita modificaciones
- Que soporte reusabilidad del software

1.3.2 ¿Qué hay que saber para construir o elegir una metodología?

En el momento de adoptar un estándar o construir una metodología, se han de considerar unos requisitos deseables, por lo que seguidamente se proponen una serie de criterios de evaluación de dichos requisitos.

1- La metodología debe ajustarse a los objetivos.

Cada aproximación al desarrollo de software está basada en unos objetivos. Por ello la metodología que se elija debe recoger el aspecto filosófico de la aproximación deseada, es decir que los objetivos generales del desarrollo deben estar implementados en la metodología de desarrollo.

2- La metodología debe cubrir el ciclo entero de desarrollo de software.

Para ello la metodología ha de realizar unas etapas:

- Investigación
- Análisis de requisitos
- Diseño

3- La metodología debe integrar las distintas fases del ciclo de desarrollo.

Rastreabilidad: Es importante poder referirse a otras fases de un proyecto y fusionarlo con las fases previas. Es importante poder moverse no sólo hacia adelante en el ciclo de vida, sino hacia atrás de forma que se pueda comprobar el trabajo realizado y se puedan efectuar correcciones.

Fácil interacción entre etapas del ciclo de desarrollo. Es necesaria una validación formal de cada fase antes de pasar a la siguiente. La información que se pierde en una fase determinada queda perdida para siempre, con un impacto en el sistema resultante.

4- La metodología debe incluir la realización de validaciones.

La metodología debe detectar y corregir los errores cuanto antes. Uno de los problemas más frecuentes y costosos es el aplazamiento de la detección y corrección de problemas en las etapas finales del proyecto.

Cuanto más tarde sea detectado el error más caro será corregirlo. Por lo tanto cada fase del proceso de desarrollo de software deberá incluir una actividad de validación explícita.

5- La metodología debe soportar la determinación de la exactitud del sistema a través del ciclo de desarrollo.

La exactitud del sistema implica muchos asuntos, incluyendo la correspondencia entre el sistema y sus especificaciones, así como que el sistema cumple con las necesidades del usuario. Por ejemplo, los métodos usados para análisis y especificación del sistema deberían colaborar a terminar con el problema del entendimiento entre los informáticos, los usuarios, y otras partes implicadas. Esto implica una comunicación entre usuario y técnico amigable y sencillo, exento de consideraciones técnicas.

6- La metodología debe ser la base de una comunicación efectiva.

Debe ser posible gestionar a los informáticos, y éstos deben ser capaces de trabajar conjuntamente. Ha de haber una comunicación efectiva entre analistas, programadores, usuarios y gestores, con pasos bien definidos para realizar progresos visibles durante la actividad del desarrollo.

7- La metodología debe funcionar en un entorno dinámico orientado al usuario.

A lo largo de todo el ciclo de vida del desarrollo se debe producir una transferencia de conocimientos hacia el usuario. La clave del éxito es que todas las partes implicadas han de intercambiar información libremente. La participación del usuario es de importancia vital debido a que sus necesidades evolucionan constantemente. Por otra parte la adquisición de conocimientos del usuario le permitirá la toma de decisiones correctas.

Para involucrar al usuario en el análisis, diseño y administración de datos, es aconsejable el empleo de técnicas estructuradas lo más sencillas posible. Para esto, es esencial contar con una buena técnica de diagramación.

8- La metodología debe especificar claramente los responsables de resultados.

Debe especificar claramente quienes son los participantes de cada tarea a desarrollar, debe detallar de una manera clara los resultados de los que serán responsables.

9- La metodología debe poder emplearse en un entorno amplio de proyectos software.

Variedad: Una empresa deberá adoptar una metodología que sea útil para un gran número de sistemas que vaya a construir. Por esta razón no es práctico adoptar varias metodologías en una misma empresa.

Tamaño de vida: Las metodologías deberán ser capaces de abordar sistemas de distintos tamaños y rangos de vida.

Complejidad: La metodología debe servir para sistemas de distinta complejidad, es decir puede abarcar un departamento, varios departamentos o varias empresas.

Entorno: La metodología debe servir con independencia de la tecnología disponible en la empresa.

10- La metodología se debe de poder enseñar.

Incluso en una organización sencilla, serán muchas las personas que la van a utilizar, incluso los que se incorporen posteriormente a la empresa. Cada persona debe entender las técnicas específicas de la metodología, los procedimientos organizativos y de gestión que la hacen efectiva, las herramientas automatizadas que soportan la metodología y las motivaciones que subyacen en ella.

11- La metodología debe estar soportada por herramientas CASE.

La metodología debe estar soportada por herramientas automatizadas que mejoren la productividad, tanto del ingeniero de software en particular, como la del desarrollo en general. El uso de estas herramientas reduce el número de personas requeridas y la sobrecarga de comunicación, además de ayudar a producir especificaciones y diseños con menos errores, más fáciles de probar, modificar y usar.

12- La metodología debe soportar la eventual evolución del sistema.

Normalmente durante su tiempo de vida los sistemas tienen muchas versiones, pudiendo durar incluso más de 10 años. Existen herramientas CASE para la gestión de la configuración y otras denominadas "Ingeniería inversa" para ayudar en el mantenimiento de los sistemas no estructurados, permitiendo estructurar los componentes de éstos facilitando así su mantenimiento.

13- La metodología debe contener actividades que conduzcan a mejorar el proceso de desarrollo de software.

Para mejorar el proceso es básico disponer de datos numéricos que evidencian la efectividad de la aplicación del proceso con respecto a cualquier producto software resultante del proceso. Para disponer de estos datos, la metodología debe contener un conjunto de mediciones de proceso para identificar la calidad y coste asociado a cada etapa del proceso. Sería ideal el uso de herramientas CASE.

1.4 El Software Educativo.

1.4.1 Tipologías del software

- **Software de sistema**, que permite funcionar al hardware. Su objetivo es aislar tanto como sea posible al programador de aplicaciones de los detalles del computador particular que se use, especialmente de las características físicas de la memoria, dispositivos de comunicaciones, impresoras, pantallas, teclados, etcétera. Incluye entre otros:
 - Sistemas operativos
 - Controladores de dispositivo
 - Herramientas de diagnóstico
 - Servidores
 - Sistemas de ventanas
 - Utilidades
- **Software de programación**, que proporciona herramientas para ayudar al programador a escribir programas informáticos y a usar diferentes lenguajes de programación de forma práctica. Incluye entre otros:
 - Editores de texto
 - Compiladores
 - Intérpretes
 - Enlazadores
 - Depuradores
- **Software de aplicación**, que permite a los usuarios llevar a cabo una o varias tareas más específicas, en cualquier campo de actividad susceptible de ser automatizado o asistido, con especial énfasis en los negocios. Incluye entre otros:
 - Aplicaciones de automatización industrial.
 - Aplicaciones ofimáticas.
 - **Software educativo.**
 - Software médico.
 - Bases de datos.
 - Videojuegos

1.4.2 Concepto de Software Educativo.

El SWE se puede definir como un programa automatizado diseñado con una intencionalidad educativa para ser utilizado en el proceso de aprendizaje, utiliza procedimientos para que el estudiante aprenda, se fomenta el análisis de problemas, facilita el trabajo en grupo, provee soporte en actividades docentes y en el sentido más amplio, mejora las habilidades del pensamiento y la resolución de problemas. [12]

Otros autores definen como software educativo como cualquier programa computacional cuyas características estructurales y funcionales sirvan de apoyo al proceso de enseñar, aprender y administrar. [9]

Un concepto más restringido de SWE lo define como aquel material de aprendizaje especialmente diseñado para ser utilizado con una computadora en los procesos de enseñar y aprender. [9]

Finalmente, los SWE se pueden considerar como el conjunto de recursos informáticos diseñados con la intención de ser utilizados en el contexto del proceso de enseñanza-aprendizaje.

El software educativo se ha enfocado principalmente en dos polos:

El SWE de tipo algorítmico que es aquel en donde predomina el aprendizaje por transmisión de conocimientos, dentro de estos se pueden considerar los denominados tutoriales, entrenadores y libros electrónicos.

El SWE de tipo heurísticos que es donde el estudiante descubre el conocimiento interactuando con el ambiente de aprendizaje que le permite llegar a él, aquí se encuentran los simuladores, juegos educativos, sistemas expertos y sistemas tutoriales inteligentes.

Cada uno de estos software tienen sus cualidades y limitaciones que se deben tener en cuenta a la hora de seleccionar uno de ellos, dada una necesidad educativa. El SWE es uno de los pilares del sistema de educación a distancia y se perfila como la herramienta base de las próximas generaciones de educandos.

1.4.3 Clasificaciones de los software educativos.

El SWE es la base fundamental del proceso de enseñanza-aprendizaje en la actualidad, estos se presentan de diferentes formas, unos se presentan como juegos, otros como laboratorios o libros, etc.

Los materiales educativos, que integran diversos elementos textuales (secuenciales e hipertextuales) y audiovisuales (gráficos, sonido, vídeo, animaciones)

Se clasifican atendiendo a su estructura y a su concepción sobre el aprendizaje.

➤ **Atendiendo a su estructura:** los materiales didácticos multimedia se pueden clasificar en programas tutoriales, de ejercitación, simuladores, bases de datos, constructores, programas herramientas, 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 detalles, la clasificación es la siguiente:

- **Materiales formativos directivos:** Proporcionan información, proponen preguntas y ejercicios a los alumnos y corrigen sus respuestas.

- Programas de ejercitación. Se limitan a proponer ejercicios auto correctivos de refuerzo sin proporcionar explicaciones conceptuales previas.

Su estructura puede ser:

- Lineal (la secuencia en la que se presentan las actividades es única o totalmente aleatoria).
- Ramificada (la secuencia depende de los aciertos de los usuarios).
- 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 auto correctivos al respecto. Si utilizan técnicas de Inteligencia Artificial para personalizar la tutorización según las características de cada estudiante, se denominan tutoriales expertos.

- Bases de datos: Presentan datos organizados en un entorno estático mediante unos criterios que facilitan su exploración y consulta selectiva para resolver problemas, analizar y relacionar datos, comprobar hipótesis, extraer conclusiones. Al utilizarlos se pueden formular preguntas del tipo: ¿Qué características tiene este dato? ¿Qué datos hay con la característica X? ¿Y con las características X e Y?

- Programas tipo libro o cuento: Presenta una narración o una información en un entorno estático como un libro o cuento.

- Bases de datos convencionales: Almacenan la información en ficheros, mapas o gráficos, que el usuario puede recorrer según su criterio para recopilar información.
- Bases de datos expertas: Son bases de datos muy especializadas que recopilan toda la información existente de un tema concreto y además asesoran al usuario cuando accede buscando determinadas respuestas.
- **Simuladores:** Presentan modelos dinámicos interactivos (generalmente con animaciones) y los alumnos realizan aprendizajes significativos por descubrimiento al explorarlos, modificarlos y tomar decisiones ante situaciones de difícil acceso en la vida real (pilotar un avión, viajar por la historia a través del tiempo). Al utilizarlos se pueden formular preguntas del tipo: ¿Qué le pasa al modelo si modifico el valor de la variable X? ¿Y si modifico el parámetro Y?
- Modelos físico-matemáticos. Presentan de manera numérica o gráfica una realidad que tiene unas leyes representadas por un sistema de ecuaciones deterministas. Incluyen los programas-laboratorio, trazadores de funciones y los programas que con un convertidor analógico-digital captan datos de un fenómeno externo y presentan en pantalla informaciones y gráficos del mismo.
- Entornos sociales. Presentan una realidad regida por unas leyes no del todo deterministas. Se incluyen aquí los juegos de estrategia y de aventura.
- **Constructores o talleres creativos:** Facilitan aprendizajes heurísticos, de acuerdo con los planteamientos constructivistas. Son entornos programables (con los interfaces convenientes se pueden controlar pequeños robots), que facilitan unos elementos simples con los cuales pueden construir entornos complejos. Los alumnos se convierten en profesores del ordenador. Al utilizarlos se pueden formular preguntas del tipo: ¿Qué sucede si añado o elimino el elemento X?
- Constructores específicos. Ponen a disposición de los estudiantes unos mecanismos de actuación (generalmente en forma de órdenes específicas) que permiten la construcción de determinados entornos, modelos o estructuras.
- Lenguajes de programación. Ofrecen unos "laboratorios simbólicos" en los que se pueden construir un número ilimitado de entornos.
- **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.
- Programas de uso general. Los más utilizados son programas de uso general (procesadores de textos, editores gráficos, hojas de cálculo) que provienen del mundo laboral. No obstante, se han elaborado versiones "para niños" que limitan sus posibilidades a cambio de una, no siempre clara, mayor facilidad de uso.
- Lenguajes y sistemas de autor. Facilitan la elaboración de programas tutoriales a los profesores que no disponen de grandes conocimientos informáticos.

➤ **Atendiendo a su concepción sobre el aprendizaje:** en los materiales didácticos multimedia se identifican diversos planteamientos: la perspectiva conductista, la teoría del procesamiento de la información, el aprendizaje por descubrimiento, el aprendizaje significativo, el enfoque cognitivo, el constructivismo, el socio-constructivismo.

- La perspectiva conductista. Desde la perspectiva conductista, formulada por B.F. Skinner hacia mediados del siglo XX y que arranca de Wundt y Watson, pasando por los estudios psicológicos de Pavlov sobre condicionamiento y de los trabajos de Thorndike sobre el refuerzo, intenta explicar el aprendizaje a partir de unas leyes y mecanismos comunes para todos los individuos.

- Condicionamiento operante. Formación de reflejos condicionados mediante mecanismos de estímulo-respuesta-refuerzo. Aprendizaje = conexiones entre estímulos y respuestas.

- Ensayo y error con refuerzos y repetición: las acciones que obtienen un refuerzo positivo tienden a ser repetidas.

- Asociacionismo: los conocimientos se elaboran estableciendo asociaciones entre los estímulos que se captan. Memorización mecánica.

- Enseñanza programada: Resulta especialmente eficaz cuando los contenidos están muy estructurados y secuenciados y se precisa un aprendizaje memorístico. Su eficacia es menor para la comprensión de procesos complejos y la resolución de problemas no convencionales. Los primeros ejemplos están en las máquinas de enseñar de Skinner y los sistemas ramificados de Crowder.

- En muchos materiales didácticos multimedia directivos (ejercitación, tutoriales) subyace esta perspectiva.

• **Teoría del procesamiento de la información:** La teoría del procesamiento de la información, influida por los estudios cibernéticos de los años cincuenta y sesenta, presenta una explicación sobre los procesos internos que se producen durante el aprendizaje. Sus planteamientos básicos, en líneas generales, son ampliamente aceptados. Considera las siguientes fases principales:

- Captación y filtro de la información a partir de las sensaciones y percepciones obtenidas al interactuar con el medio.

- Almacenamiento momentáneo en los registros sensoriales y entrada en la memoria a corto plazo, donde, si se mantiene la actividad mental centrada en esta información, se realiza un reconocimiento y codificación conceptual.

- Organización y almacenamiento definitivo en la memoria a largo plazo, donde el conocimiento se organiza en forma de redes. Desde aquí la información podrá ser recuperada cuando sea necesario.

- En muchos materiales didácticos multimedia directivos (ejercitación, tutoriales) subyace esta perspectiva.

- **Aprendizaje por descubrimiento:** La perspectiva del aprendizaje por descubrimiento, desarrollada por J. Bruner, atribuye una gran importancia a la actividad directa de los estudiantes sobre la realidad.

- Experimentación directa sobre la realidad, aplicación práctica de los conocimientos y su transferencia a diversas situaciones.
- Aprendizaje por penetración comprensiva. El alumno experimentando descubre y comprende lo que es relevante, las estructuras.
- Práctica de la inducción: de lo concreto a lo abstracto, de los hechos a las teorías.
- Utilización de estrategias heurísticas, pensamiento divergente.
- Currículum en espiral: revisión y ampliación periódica de los conocimientos adquiridos.
- Esta perspectiva está presente en la mayoría de los materiales didácticos multimedia no directivos (simuladores, constructores).

- **Aprendizaje significativo:** (D. Ausubel, J. Novak) postula que el aprendizaje debe ser significativo, no memorístico, y para ello los nuevos conocimientos deben relacionarse con los saberes previos que posea el aprendiz. Frente al aprendizaje por descubrimiento de Bruner, defiende el aprendizaje por recepción donde el profesor estructura los contenidos y las actividades a realizar para que los conocimientos sean significativos para los estudiantes.

Condiciones para el aprendizaje:

- Significación lógica (se puede relacionar con conocimientos previos).
- Significación psicológica (adecuación al desarrollo del alumno).
- Actitud activa y motivación.
- Relación de los nuevos conocimientos con los saberes previos. La mente es como una red proposicional donde aprender es establecer relaciones semánticas.
- Utilización de organizadores previos que faciliten la activación de los conocimientos previos relacionados con los aprendizajes que se quieren realizar.
- Diferenciación-reconciliación integradora que genera una memorización comprensiva.
- Funcionalidad de los aprendizajes, que tengan interés, se vean útiles.
- Esta perspectiva está presente en la mayoría de los materiales didácticos multimedia.
- **Enfoque cognitivo:** Psicología cognitivista. El cognitivismo (Merrill, Gagné, Solomon), basado en las teorías del procesamiento de la información y recogiendo también algunas ideas conductistas (refuerzo, análisis de tareas) y del aprendizaje significativo, aparece en la década de los sesenta y pretende dar una explicación más detallada de los procesos de aprendizaje, distingue:
 - El aprendizaje es un proceso activo. El cerebro es un procesador paralelo, capaz de tratar con múltiples estímulos. El aprendizaje tiene lugar con una combinación de fisiología y emociones. El desafío estimula el aprendizaje, mientras que el miedo lo retrae.

El estudiante representará en su mente simbólicamente el conocimiento, que se considera (igual que los conductistas) como una realidad que existe externamente al estudiante y que éste debe adquirir. El aprendizaje consiste en la adquisición y representación exacta del conocimiento externo. La enseñanza debe facilitar la transmisión y recepción por el alumno de este conocimiento estructurado.

Posteriormente cuando se haga una pregunta al estudiante se activarán las fases: recuerdo, generalización o aplicación (si es el caso) y ejecución (al dar la respuesta, que si es acertada dará lugar a un refuerzo)

Condiciones internas que intervienen en el proceso: motivación, captación y comprensión, adquisición, retención.

- Condiciones externas: son las circunstancias que rodean los actos didácticos y que el profesor procurará que favorezcan al máximo los aprendizajes.

- En muchos materiales didácticos multimedia directivos (ejercitación, tutoriales) subyace esta perspectiva.

• **Constructivismo:** J. Piaget, en sus estudios sobre epistemología genética, en los que determina las principales fases en el desarrollo cognitivo de los niños, elaboró un modelo explicativo del desarrollo de la inteligencia y del aprendizaje en general a partir de la consideración de la adaptación de los individuos al medio.

- Considera tres estadios de desarrollo cognitivo universales: sensorio motor, estadio de las operaciones concretas y estadio de las operaciones formales. En todos ellos la actividad es un factor importante para el desarrollo de la inteligencia.

- Construcción del propio conocimiento mediante la interacción constante con el medio. Lo que se puede aprender en cada momento depende de la propia capacidad cognitiva, de los conocimientos previos y de las interacciones que se pueden establecer con el medio. En cualquier caso, los estudiantes comprenden mejor cuando están envueltos en tareas y temas que cautivan su atención. El profesor es un mediador y su metodología debe promover el cuestionamiento de las cosas, la investigación.

- Reconstrucción de los esquemas de conocimiento. El desarrollo y el aprendizaje se produce a partir de la secuencia: equilibrio - desequilibrio – reequilibrio (que supone una adaptación y la construcción de nuevos esquemas de conocimiento).

Aprender no significa ni reemplazar un punto de vista (el incorrecto) por otro (el correcto), ni simplemente acumular nuevo conocimiento sobre el viejo, sino más bien transformar el conocimiento. Esta transformación, a su vez, ocurre a través del pensamiento activo y original del aprendiz. Así pues, la educación constructivista implica la experimentación y la resolución de problemas y considera que los errores no son antitéticos del aprendizaje sino más bien la base del mismo.

El constructivismo considera que el aprendizaje es una interpretación personal del mundo (el conocimiento no es independiente del alumno), de manera que da sentido a las experiencias que construye cada estudiante. Este conocimiento se consensua con otros, con la sociedad.

- Esta perspectiva actualmente está presente en muchos materiales didácticos multimedia de todo tipo, especialmente en los no tutoriales.

• **Socio-constructivismo:** Basado en muchas de las ideas de Vigotski, considera también los aprendizajes como un proceso personal de construcción de nuevos conocimientos a partir de los conocimientos previos (actividad instrumental), pero inseparable de la situación en la que se produce. Tiene lugar conectando con la experiencia personal y el conocimiento base del estudiante y se sitúa en un contexto social donde él construye su propio conocimiento a través de la interacción con otras personas (a menudo con la orientación del docente). Enfatiza en los siguientes aspectos:

- Importancia de la interacción social y de compartir y debatir con otros los aprendizajes. Aprender es una experiencia social donde el contexto es muy importantes y el lenguaje juega un papel básico como herramienta mediadora, no solo entre profesores y alumnos, sino también entre estudiantes, que así aprenden a explicar, argumentar... Aprender significa "aprender con otros", recoger también sus puntos de vista. La socialización se va realizando con "otros" (iguales o expertos).

- Incidencia en la zona de desarrollo próximo, en la que la interacción con los especialistas y con los iguales puede ofrecer un "andamiaje" donde el aprendiz puede apoyarse.

Actualmente el aprendizaje colaborativo y el aprendizaje situado, que destaca que todo aprendizaje tiene lugar en un contexto en el que los participantes negocian los significados, recogen estos planteamientos. El aula debe ser un campo de interacción de ideas, representaciones y valores. La interpretación es personal, de manera que no hay una realidad compartida de conocimientos. Por ello, los alumnos individualmente obtienen diferentes interpretaciones de los mismos materiales, cada uno construye (reconstruye) su conocimiento según sus esquemas, sus saberes y experiencias previas su contexto.

- Esta perspectiva actualmente está presente en algunos materiales didácticos multimedia no tutoriales.

➤ **Otras clasificaciones:** Además de considerar la "estructura", los materiales didácticos multimedia se pueden clasificar según múltiples criterios:

- Según los contenidos (temas, áreas curriculares).
- Según los destinatarios (criterios basados en niveles educativos, edad, conocimientos previos).
- Según sus bases de datos: cerrado, abierto (= bases de datos modificables).
- Según los medios que integra: convencional, hipertexto, multimedia, hipermedia, realidad virtual.
- Según su "inteligencia": convencional, experto (o con inteligencia artificial).
- Según los objetivos educativos que pretende facilitar: conceptuales, procedimentales, actitudinales (o considerando otras taxonomías de objetivos).
- Según las actividades cognitivas que activa: control psicomotriz, observación, memorización, evocación, comprensión, interpretación, comparación, relación (clasificación, ordenación), análisis, síntesis, cálculo, razonamiento (deductivo, inductivo, crítico), pensamiento divergente, imaginación, resolución de problemas, expresión (verbal, escrita, gráfica), creación, exploración, experimentación, reflexión meta cognitiva, valoración.
- Según el tipo de interacción que propicia: reconocitiva, reconstructiva, intuitiva/global, constructiva.
- Según su función en el aprendizaje: instructivo, revelador, conjetural, emancipador.
- Según su comportamiento tutor, herramienta, aprendiz.
- Según el tratamiento de errores: tutorial (controla el trabajo del estudiante y le corrige), no tutorial.
- Según sus bases psicopedagógicas sobre el aprendizaje: conductista, cognitivista, constructivista.
- Según su función en la estrategia didáctica: entrenar, instruir, informar, motivar, explorar, experimentar, expresarse, comunicarse, entretener, evaluar, proveer recursos (calculadora, comunicación telemática).
- Según su diseño: centrado en el aprendizaje, centrado en la enseñanza, proveedor de recursos.
- Según el soporte: disco, Web.

1.4.4 Funciones del software educativos.

Los materiales 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 son las siguientes:

Informativa: Estos materiales, a través de sus actividades, presentan unos contenidos que proporcionan información estructurada de la realidad, a los estudiantes.

Instructiva Entrenadora: Todo el SWE orienta y regula el aprendizaje de los estudiantes ya que, explícita o implícitamente, promueven determinadas actuaciones de los mismos encaminadas a este fin. Además, mediante sus códigos simbólicos, estructuración de la información e interactividad condicionan los procesos de aprendizaje.

Motivadora: La interacción con el ordenador suele resultar por sí misma motivadora.

Estos programas incluyen además elementos para captar la atención de los alumnos, mantener su interés y focalizarlo hacia los aspectos más importantes.

Evaluadora: La posibilidad de "feed back" inmediato a las respuestas y acciones de los alumnos, hace adecuados a los programas para evaluarles. Esta evaluación puede ser:

- Implícita: el estudiante detecta sus errores, se evalúa a partir de las respuestas que le da el ordenador.
- Explícita: el programa presenta informes valorando la actuación del alumno.

Explorar Experimentar: Les presentan a los estudiantes interesantes entornos donde explorar, experimentar, investigar, buscar determinadas informaciones, cambiar los valores de las variables de un sistema, etc.

Innovadora: Los programas educativos pueden desempeñar esta función ya que utilizan una tecnología actual y, en general, suelen permitir muy diversas formas de uso. Esta versatilidad abre amplias posibilidades de experimentación didáctica e innovación educativa en el aula.

1.4.5 Ventajas de los software educativos.

- Enriquece el campo de la pedagogía al incorporar la tecnología de punta que revoluciona los métodos de enseñanza - aprendizaje.
- Constituyen una nueva, atractiva, dinámica y rica fuente de conocimientos.
- Pueden adaptar el software a las características y necesidades de su grupo teniendo en cuenta el diagnóstico en el proceso de enseñanza - aprendizaje.
- Permiten elevar la calidad del proceso docente - educativo.
- Permiten controlar las tareas docentes de forma individual o colectiva.
- Muestran la interdisciplinariedad de las asignaturas.
- Marca las posibilidades para una nueva clase más desarrolladora.

1.4.6 Desventajas de los software educativos.

- Adicción: El software educativo interactivo resulta motivador, pero un exceso de motivación puede provocar adicción.
- Distracción: Los alumnos a veces se dedican a jugar en vez de trabajar.
- Ansiedad: La continua interacción ante el ordenador puede provocar ansiedad en los estudiantes.
- Aprendizajes incompletos y superficiales: La libre interacción de los alumnos con estos materiales (no siempre de calidad) a menudo proporciona aprendizajes incompletos con visiones de la realidad simplista y poco profunda.
- Cansancio visual y otros problemas físicos: Un exceso de tiempo trabajando ante el ordenador o malas posturas pueden provocar diversas dolencias.
- Problemas con los ordenadores: A veces los alumnos cambian la configuración o contaminan con virus los ordenadores.

1.5 El uso del software educativo en Cuba. Su inserción en el proceso pedagógico.

El Programa de Informática Educativa del MINED contempla, tanto la formación informática de nuestros niños y jóvenes, como la introducción progresiva del software educativo como medio de enseñanza a todos los niveles de educación. Si bien en lo primero es donde se ha acumulado la mayor experiencia durante una década, no es así en el segundo.

A partir del cambio de la tecnología y la introducción de las computadoras en todos los niveles de enseñanza, en el curso 2001-2002, se implementan acciones concretas para transitar progresivamente hacia un uso masivo de estos recursos como medio de enseñanza.

La presencia de las computadoras en las instituciones escolares cubanas es un hecho palpable y su empleo como medio de enseñanza se encuentra en un período de tránsito. Este tránsito se caracteriza por un uso progresivo de SWE y sistemas autorizados de recuperación informativa (SARI) combinado con las habilidades informáticas adquiridas por los alumnos en las clases de computación.

A nivel curricular debe lograrse que la dosificación de la asignatura (computación) en los diferentes grados, garantice el dominio de procedimientos bien definidos de las diferentes tipologías de software que pueden ser usados por el resto de las asignaturas del grado en los tiempos de máquina planificados.

Hoy se cuenta con computadoras, un conjunto de SWE instalados en los centros educacionales, profesores con una preparación informática adecuada en todas las escuelas en los diferentes niveles, para poder avanzar en la utilización del SWE como medio de enseñanza.

Una aplicación multimedia educativa en la actualidad, resulta un excelente medio de aprendizaje en tanto que puede presentarle a un estudiante, material proveniente de diferentes fuentes: textos, gráficos, audio, video, animación, simulaciones, fotografías, esquemas, mapas contextuales, etc. Cuando estos recursos se combinan a través de la interactividad se crean las posibilidades para el desarrollo de un entorno educativo realmente efectivo y tan centrado en el estudiante que más que llamarlo medio de enseñanza, resultaría más correcto denominarlo medio de aprendizaje.

En las escuelas de oficios el objetivo del uso de las computadoras es contribuir al trabajo correctivo-compensatorio, es por ello que los beneficios de la utilización en los estudiantes se multiplican y constituyen un recurso con grandes posibilidades educativas: enriquecen su aprendizaje, acentúan sus fortalezas, eliminan el sentido del fracaso y ayudan a identificar áreas de talentos e intereses vocacionales.

1.6 Conclusiones

Al término de este capítulo se han tenido en cuenta los conceptos que están presentes o asociados de alguna forma con el dominio del problema, se ha identificado el objeto de estudio, se investigó referente a las metodologías de desarrollo de software, las clasificaciones, funciones, ventajas y desventajas del SWE y la situación actual uso en Cuba. Este capítulo tiene gran importancia para poder realizar una buena comprensión del posterior desarrollo del trabajo de diploma.

CAPÍTULO 2

Metodología RMM (Relationship Management Methodology).

2.1 Introducción

En este segundo capítulo del presente trabajo de diploma se realiza un estudio detallado de la Metodología RMM (Relationship Management Methodology) se presentan modelos de datos que antecedieron a RMM y posteriormente se abordan todos los aspectos de la metodología, las fases de su ciclo de vida, limitaciones que presenta originalmente y propuestas de mejoras; se explica cada una de sus fases y además la herramienta RMCASE que da soporte a algunas fases de esta metodología.

2.2 Modelo de referencia.

2.2.1 Modelo de Hipertexto de Dexter

El modelo de referencia de hipertexto de Dexter surge en 1988 durante unas reuniones, la primera de las cuales tuvo lugar en la Dexter Inn (New Hampshire), de donde toma su nombre, como un intento de respuesta a la pregunta: ¿qué tienen en común los distintos sistemas de hipertexto en sus nociones de nodos y enlaces entre dichos nodos?

Así, se pretendía que el modelo sirviera como un estándar para comparar las características y funcionalidades de varios sistemas de hipertexto. También como una base sobre la cual se pudiera desarrollar estándares para la interoperabilidad e intercambio entre sistemas de hipertexto.

Otro punto importante fue el de proveer de una terminología común al campo del hipertexto.

Breve descripción del sistema.

El modelo de Dexter divide el sistema hipertexto en tres capas diferentes. La primera, la más próxima al usuario es la denominada capa de tiempo de ejecución (*run-time layer*); la intermedia es la llamada capa de almacenamiento (*storage layer*), mientras que la última recibe el nombre de capa del componente (*within-component layer*).

Tabla 1. Capas del modelo de Dexter.

Capa de tiempo de ejecución (<i>run-time layer</i>).
Especificaciones de presentación.
Capa de almacenamiento (<i>storage layer</i>).
Anclaje (Anchoring).
Capa del componente (<i>within-component layer</i>).

El modelo se centra en la capa de almacenamiento, que modela la estructura básica de nodos y enlaces. Es como una base de datos donde se almacena una jerarquía de componentes (contenedores de la información) interconectados por enlaces.

Los componentes es lo que se conoce típicamente como nodos (por ejemplo una tarjeta en Hypercard, o un frame en Director). Por tanto, un componente puede contener fragmentos de texto, gráficos, animaciones. En la capa de almacenamiento se observan los llamados componentes atómicos, que al unirse forman componentes compuestos, que es el equivalente a la noción de nodo. La capa de almacenamiento estudia los mecanismos por los cuales se unen dichos componentes y enlaces.

Por el contrario, la capa del componente refleja los contenidos y estructura dentro de los componentes. Debido a que hay un sin fin de posibles contenidos/estructuras que pueden ser incluidos en un componente, el modelo de Dexter no profundiza el estudio de esta capa, dando libertad en este nivel.

Una parte fundamental del modelo de Dexter es la interfaz entre las capa de almacenamiento y la capa del componente. Se basa en el mecanismo de anclaje (anchoring).

En cuanto a la capa de tiempo de ejecución, ésta provee al usuario de unas herramientas para acceder, ver y manipular la estructura de la red de hipertexto. Al igual que en la capa del componente, también existen un gran número de posibles herramientas para acceder, ver y manipular un hipertexto. Por consiguiente, el modelo Dexter sólo ofrece unas ideas básicas de cómo son esos mecanismos, pero no puede hacer referencia a los detalles de cómo el usuario interactúa con el hipertexto.

Un punto importante es la interfaz entre esta capa y la capa de almacenamiento, a la que llaman especificaciones de presentación. Ésta es un mecanismo que permite que la forma en que un componente se presenta al usuario pueda estar en función no sólo de una herramienta de hipertexto específica (es decir, de una capa de tiempo de ejecución específica), sino que puede ser una propiedad del componente por sí mismo, del enlace tomado para llegar a tal componente o también de las preferencias del usuario. Por ejemplo, un componente que contiene una animación puede presentarse de dos maneras diferentes según el usuario. Si el usuario es un profesor, se le permitirá editar el componente (la animación), mientras que si el usuario es un alumno, sólo se le permite la visualización.

En la figura 2, se observa una representación más gráfica de las tres capas del modelo de Dexter. En la capa de tiempo de ejecución, se tiene la presentación del hipertexto al usuario a través de la pantalla. En la capa de almacenamiento, se muestra una base de datos donde se almacenan todos los componentes y enlaces entre ellos (los enlaces también son componentes). Y por último en la capa del componente, se guarda la estructura interna del componente.

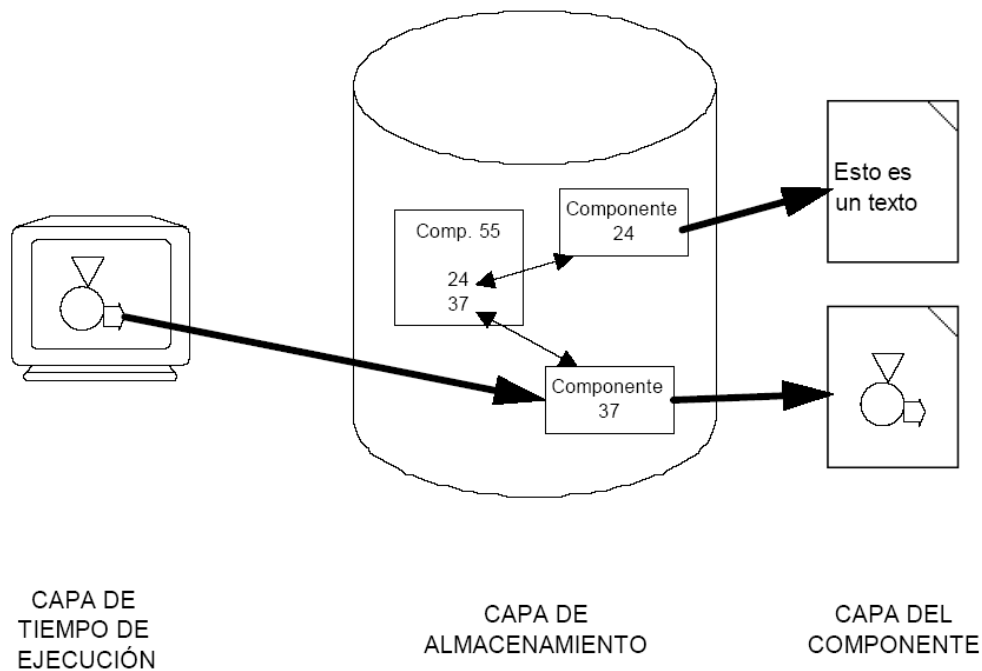


Figura 2. La tres capas del modelo de Dexter.

Se puede decir que el modelo está fundamentalmente destinado sólo a hipertexto, y la aplicación a la hipermedia es muy difícil.

En concreto, no tiene en cuenta los aspectos relacionados con el tiempo, algo fundamental en el audio y el vídeo. Es por ello que posteriormente apareció el modelo de Ámsterdam, que basado en Dexter, añade el estudio del tiempo.

2.3 Modelo con semántica de navegación formalmente definida.

2.3.1 Modelo de hipermedia de Ámsterdam.

Este modelo se definió a partir de la combinación del modelo de Dexter con los canales, añadiendo extensiones para soportar ciertos requerimientos de la hipermedia que no habían sido contemplados.

En este modelo, un componente compuesto no contiene datos, sino que los datos sólo pueden ser referenciados a través de los componentes atómicos.

Este enfoque tiene tres ventajas sobre Dexter:

- Localiza la información temporal y de presentación en los componentes atómicos, mientras que la estructura de la presentación se deja para los componentes compuestos. En teoría, esto facilita las tareas de mantenimiento.
- Promueve la reusabilidad de datos al forzar que todos los elementos se mantengan por separado.

- Modela de forma más aproximada la forma en que la información multimedia se almacena, esto es, en sistemas de ficheros o bases de datos.

Así, los componentes atómicos contienen información sobre la presentación, atributos, información de anclas y campo de contenidos (datos). Lo nuevo respecto al modelo de Dexter es el campo de presentación, en el cual este se dedica a aspectos relacionados con el tiempo.

En cuanto a los componentes compuestos, se añaden varios elementos nuevos respecto a Dexter. La diferencia principal es que ahora es usado para especificar la estructura de una presentación y no sólo para unir componentes para la navegación. Los atributos temporales de la presentación se obtienen de los componentes atómicos, mientras que en la especificación de la presentación (componentes compuestos) aparecen los arcos de sincronización, unas nuevas estructuras que dan una información sobre el orden relativo de los componentes atómicos en la presentación.

Como ya se ha explicado el principal interés del modelo de Ámsterdam consiste en el intento de extender el modelo de Dexter para que contemple la complejidad de los tipos multimedia, y muy en especial del problema del tiempo que éstos conllevan.

2.4 Modelo de representación relacional.

2.4.1 El modelo HDM

HDM (*Hypertext Design Model*) fue creado por los profesores Franca Garzotto y Paolo Paolini del Politecnico di Milano y por Daniel Schwabe de la Pontificia Universidad Católica do Río de Janeiro en 1991, y en parte, en el marco del proyecto de la Comunidad Europea HYTEA. El objetivo era de crear un modelo que fuera de utilidad para realizar el diseño de una aplicación de hipertexto (o hipermedia).

Al desarrollar una aplicación hipermedia se pueden identificar dos tipos principales de trabajos: los globales, tales como definir las clases de elementos de información y estructuras navegacionales de la aplicación; y por otro lado, los locales, tales como llenar los contenidos de los nodos.

2.4.1.1 Primitivas

Este método tiene una base de terminología ya utilizada en otros métodos, pero a su vez añade otro nuevo conjunto de términos y conceptos.

La unidad básica del modelo es la entidad. Una entidad es la más pequeña parte autónoma de la información, es decir, que no necesita ninguna otra información para tener un sentido total. Las entidades se agrupan en tipos de entidades. Una entidad es una jerarquía de componentes, los cuales están formados a su vez por unidades. El concepto de unidad es similar al de nodo, mientras que un componente es un conjunto de nodos que crean una unidad lógica.

2.4.1.2 Entidades y tipos de entidades

Una entidad es una estructura (relativamente grande) de información que representa algún objeto del mundo real en el dominio de la aplicación. Las entidades se agrupan lógicamente en lo que se llaman tipos de entidades, que corresponden a las clases de objetos del mundo real. Como es evidente, el concepto de entidad es diferente en HDM que en la terminología relacional. Por ejemplo, en HDM las entidades tienen complejas estructuras internas.

2.4.1.3 Componentes

Una entidad es una colección de componentes estructurados en forma de árbol (jerárquicamente). Un componente es una abstracción de un conjunto de unidades, que son los contenedores de la información. Un componente dentro de la jerarquía de la entidad en general tiene un padre (excepto en el caso del componente raíz), varios hermanos, y un número de hijos (excepto en los componentes hoja -los últimos de la jerarquía-). Pero un componente sólo puede existir como parte de una entidad, y no tiene sentido por sí mismo, es decir, no es autónomo. Por tal razón en el modelo HDM no se permite que unas entidades sean componentes de otras entidades.

En cuanto a la jerarquía que estructura los componentes de una entidad, ésta puede ser según diversos criterios semánticos.

2.4.1.4 Perspectivas

En HDM la noción de tener diferentes presentaciones del mismo contenido se representa mediante el concepto de perspectiva. Por ejemplo, una aplicación multilingüe, la misma información se presenta de varias maneras diferentes (una en cada idioma). Pero esto, lógicamente, no altera la estructura jerárquica de una entidad. Las perspectivas son sólo una construcción para organizar este tipo de información. HDM deja a la libertad del usuario cuándo y cómo debe utilizar perspectivas.

2.4.1.5 Unidades

Una unidad es un conjunto de partes atómicas de información que se muestran conjuntamente, como una unidad. Una unidad corresponde a un componente asociado con una perspectiva específica. Una unidad se caracteriza por un nombre (su identificador) y un cuerpo. Los cuerpos de las unidades son el lugar donde queda almacenada la información. Por tanto, el concepto de unidad es muy similar al concepto en hipertexto tradicional de "nodo".

2.4.1.6 Enlaces en general

Los enlaces en hipertexto tienen una doble función: por un lado representar las relaciones entre elementos del dominio (entidades, componentes o unidades), y por otro, una función de navegación (representar los patrones navegacionales). A menudo, estas dos funciones son consistentes entre sí, aunque no siempre: puede ocurrir que las relaciones de dominio no sean relevantes para la navegación, o que en un enlace navegacional interesante no se tenga ningún contenido semántico.

Por tanto, la definición de enlace debe reflejar ambas funciones. Para ello se identifican tres tipos diferentes:

- **Enlaces de perspectiva o de componentes:** Interconectan unidades dentro del mismo componente. Nótese que al cruzar un enlace de este tipo, el contexto sigue siendo el mismo. Los enlaces de presentación o de componente pueden representarse de dos formas: enlaces de índice o enlaces de visita guiada.

- **Enlaces estructurales:** Interconectan componentes dentro de la misma entidad. Navegacionalmente son algo más complejos que los de perspectiva, pero bastante simples. Hay varios tipos, como por ejemplo: de un componente a su hermano, de un componente a su hijo o a su padre.

- **Enlaces de aplicación:** Representan relaciones de dominio entre entidades o sus componentes. Estas relaciones son las elegidas por el autor como navegacionales y semánticamente relevantes. Los enlaces de aplicación se estructuran en tipos, llamados tipos de enlaces de aplicación, o más brevemente, tipos de enlaces. El tipo viene definido por un nombre, los tipos de entidades origen y destino, y por si es simétrico o no.
 - Los enlaces de aplicación pueden ser de dos tipos diferentes:
 - De esquema: pertenecen a un determinado tipo de enlace.
 - Genéricos: no pertenecen a ningún tipo de enlace. Son independientes de la estructura, y por tanto menos estandarizados.

2.4.2 Esquema HDM

Una especificación de un hipertexto consta de una definición de esquema y un conjunto de definiciones de instancias. Un esquema especifica un conjunto de tipos de entidad y tipos de enlace. Las instancias se introducen en el sistema sólo si cumplen las restricciones establecidas en el esquema.

El concepto de esquema en hipertexto es novedoso, pero ya ha sido ampliamente utilizado en el campo de las bases de datos.

Por último, también se pueden distinguir dos conceptos en una aplicación:

- La estructura, que es la organización de los contenidos de la aplicación: entidades, componentes, enlaces.
- La dinámica, que representa el comportamiento de la aplicación.

HDM es más que un intento de modelar la estructura del hipertexto-hipermedia, es un modelado de las estructuras de navegación. En HDM la estructura de navegación viene marcada por la estructura de datos. Además HDM puede resultar útil para estudiar aplicaciones ya desarrolladas, con el fin de detectar errores en la estructura navegacional.

Realizar un modelo siguiendo las normas de HDM es extremadamente complicado cuando el número de entidades involucradas crece.

2.5 RMM (Relationship Management Methodology).

RMM a diferencia de su antecesora HDM, se puede considerar una metodología pues asume un ciclo de vida completo compuesto de siete fases o etapas en las que el diseñador va modelando la estructura de la aplicación y las posibilidades de navegación de la misma.

La metodología fue creada por Isakowitz, Stohr y Balasubramanian y según sus autores, es apropiada para dominios con estructuras regulares (es decir, con clases de objetos bien definidas, y con claras relaciones entre esas clases). [11] Por ejemplo, catálogos o "frentes" de bases de datos tradicionales. Está orientada a problemas con datos dinámicos que cambian con mucha frecuencia, más que a entornos estáticos.

La Metodología RMM se define como un proceso de análisis, diseño y desarrollo de aplicaciones hipermedia cuya estructura es estable y cuyo contenido sufre modificaciones frecuentes.

La propuesta está basada en el modelo entidad-relación E-R y en HDM. Partiendo de ellos define un nuevo modelo, el modelo de datos RMDM, que propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipermedia de la aplicación.

El modelo de datos RMDM (Relationship Management Data Model), se encuentra incluido como parte de la metodología RMM [11] la cual se muestra en la figura 3 en el contexto del ciclo de vida del software. Al igual que con Labyrinth y OOHDM estas metodologías rozan su caracterización como modelo de proceso, pero en última instancia son metodologías ya que utilizan con profusión técnicas concretas de desarrollo e implementación. Aunque esta caracterización pudiera ser discutible, lo indiscutible es que los modelos de proceso surgen como generalizaciones de uso de metodologías concretas.

RMM se centra en las fases de diseño, desarrollo y construcción, aunque prestando especial atención al diseño de los mecanismos de acceso [13] (dentro de la zona gris de la figura 3). Las etiquetas en las flechas muestran los productos generados durante el uso de la metodología. Los bucles de realimentación de las fases de diseño aparecen en línea discontinua. Aunque también hay bucles de realimentación entre el resto de las fases, no están explícitas en la figura 3. El modelo de datos RMDM proporciona un importante vehículo para el diseño de la aplicación hipermedia, pero también existen otros elementos no representables mediante este modelo de datos. En particular, el modelo es incapaz de representar algo tan sencillo como un ancla dentro de un contenido.

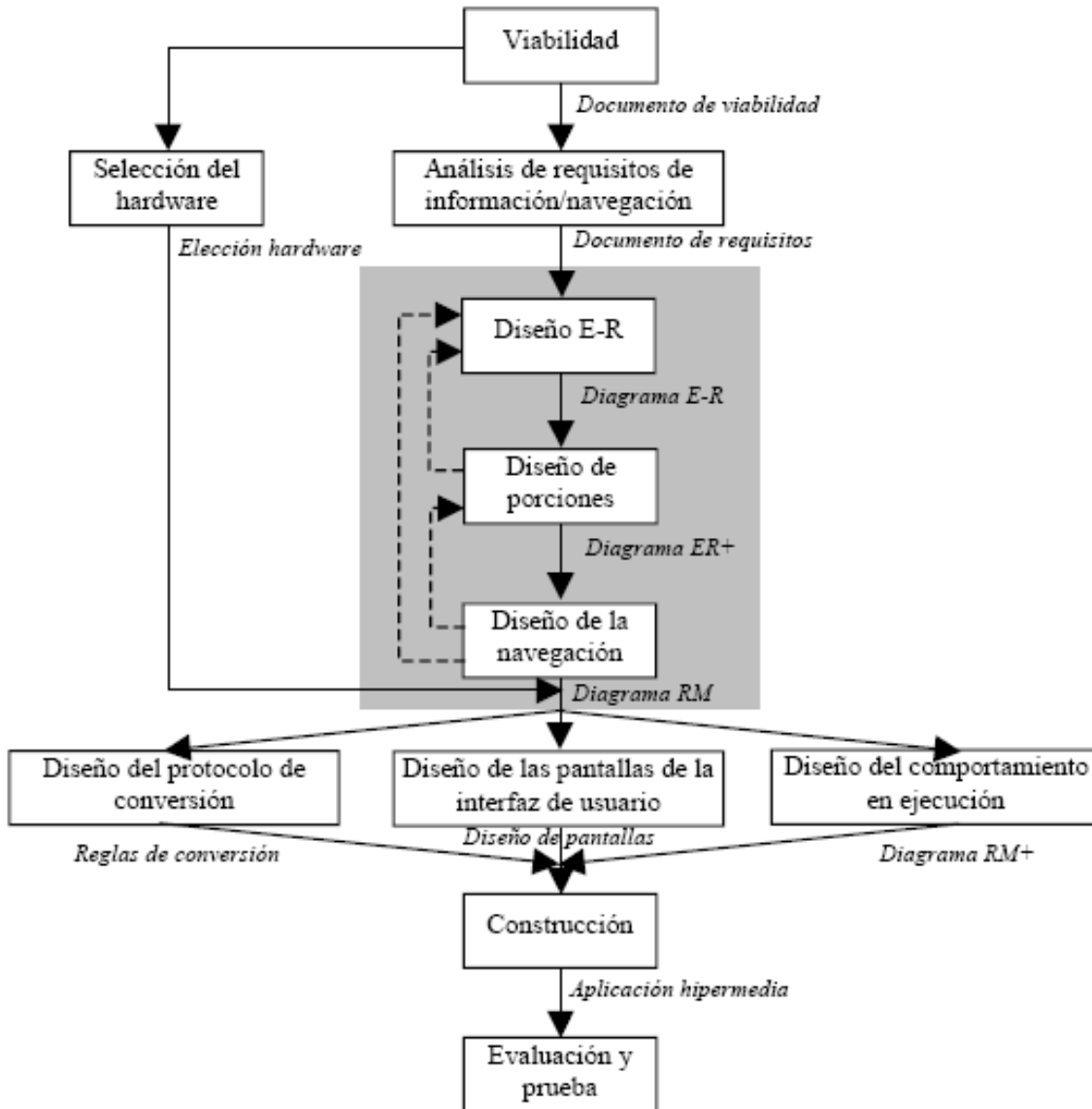


Figura 3. La metodología RMM en el contexto del ciclo de vida del software.

La Metodología RMM considera en sus etapas de diseño como primer modelo de datos al modelo entidad-relación (E-R), que permite caracterizar el dominio de información y sus relaciones. Tal representación se realiza mediante un esquema entidad-relación.

Los tres elementos básicos del modelo a utilizar son: entidades, atributos y relaciones.

- Una entidad es un elemento conceptual del dominio de aplicación, caracterizado por un conjunto de atributos.
- Un atributo representa una unidad de información. Los atributos poseen un nombre, un tipo (texto, imagen u otro medio) y están siempre asociados a una única entidad.
- Una relación es una unión conceptual entre dos o más entidades. Su cardinalidad puede ser uno-uno o uno-muchos. Las interrelaciones con cardinalidad muchos-muchos son divididas en dos uno-muchos.

Así los objetos del dominio se definen con la ayuda de entidades, atributos y relaciones asociativas. La base de la metodología es el modelo de datos RMDM (Relationship Management Data Model), que se genera a partir del modelo entidad-relación. Con él se describirá no sólo la información referente a las clases de objetos, sino también a la navegación entre ellos.

Así, hay definidas unas primitivas para modelar los dominios (clases de objetos) y otras para el acceso a tales objetos. De entre las primeras, la más típica es la entidad. Como en la teoría relacional una entidad está compuesta por varios atributos. Además, en RMDM se incorpora una nueva primitiva muy importante denominada slice, que define conjuntos de atributos de una entidad que se agrupan de forma lógica. Un slice agrupará uno o más atributos de una entidad de tipos muy diferentes, ya que cuando existen diferentes medios, la información de una entidad puede ser muy variada.

Una entidad se representa mediante un recuadro y su nombre, mientras que un slice se representa mediante una figura similar a una gota de agua.

Como ya se expresó anteriormente el modelo introduce el concepto de slice (trozo) que corresponde a un subconjunto de atributos de una misma entidad destinados a ser presentados de forma agrupada en una pantalla. La navegación se representa con la ayuda de primitivas de acceso, enlaces estructurales (unidireccional y bidireccional) que permiten especificar la navegación entre slices e índice condicional, visita guiada condicional, visita guiada indexada condicional y agrupación, que permiten especificar la navegación entre entidades. Otro término utilizado en este modelo es la primitiva de grupo, que permite hacer explícita la jerarquía de menús.

2.5.1 Las Primitivas de Datos y de Acceso.

2.5.1.1 Primitivas de Datos.

- Entidad: elemento conceptual del dominio de aplicación, caracterizado por un conjunto de atributos, se representa mediante un recuadro y su nombre.



Figura 4. Representación gráfica de una entidad

- Atributo: representa una unidad de información. Los atributos poseen un nombre, un tipo (texto, imagen u otro medio) y están siempre asociados a una única entidad. Se representan mediante un círculo y su nombre.



Figura 5. Representación gráfica de un atributo.

- Slice: define conjuntos de atributos de una entidad que se agrupan de forma lógica. Se representa mediante una figura similar a una gota de agua.



Figura 6. Representación gráfica de un slice.

2.5.1.2 Primitivas de Acceso.

En cuanto a las primitivas de acceso, se definen tres tipos de acceso diferentes. Las instancias de una entidad se pueden visitar según esos tres tipos de acceso:

Índice condicional (*Conditional Index*): en una pantalla aparecerá un índice del contenido en cuestión, y pulsando sobre uno, se puede ir a ver su información. Para ver los demás se debe volver al índice. En realidad el índice no es necesario que sea siempre una lista de las instancias como tal. Su representación gráfica:

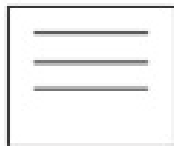


Figura 7. Representación gráfica de un índice condicional.

Visita guiada condicional (*Conditional Guided Tour*): de la pantalla inicial se pasa al primer contenido, y de él se puede pasar al segundo, y así sucesivamente, de forma que para llegar al último hay que pasar por todos los anteriores. Se permite volver a visitar al anterior. En definitiva, es una estructura de acceso secuencial.

Su representación gráfica:

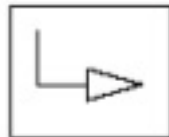


Figura 8. Representación gráfica de una visita guiada condicional.

Visita guiada indexada condicional (*Conditional Indexed Guided Tour*): es una solución híbrida, donde se tiene un índice para acceder puntualmente a los elementos de la entidad, pero también se permite la navegación secuencial una vez seleccionado uno.

Su representación gráfica es:

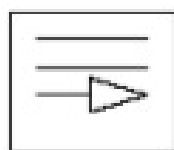


Figura 9. Representación gráfica de una visita guiada indexada condicional.

Además, se define una **Primitiva de Grupo** (*Grouping*) mediante la cual se representa el menú. Se puede colgar tantas opciones como se quiera que haya en el menú; de una primitiva de grupo. La primitiva de grupo, o a la que a veces también denominaremos de menú, se representa mediante un triángulo invertido.

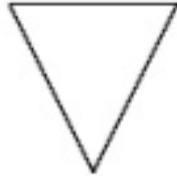


Figura 10. Representación gráfica de una primitiva de grupo o menú.

De este colgarán todas las opciones del menú.

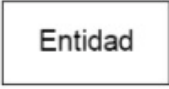







También, se define **Enlace**: esta primitiva se asocia con las relaciones asociativas (entre entidades) y las relaciones estructurales (entre trozos).



Figura 11. Representación gráfica de un enlace.

A continuación se muestra una tabla resumen con las primitivas de datos y de acceso.

Tabla 2. Las primitivas de datos y de acceso de la metodología RMM.

Primitivas de datos	Entidad	
	Atributo	
	Slice	
Primitivas de acceso	Índice condicional	
	Visita guiada condicional	
	Visita guiada indexada condicional	
	Grupo	
	Enlace	

2.5.2 Obtención del diagrama RMDM.

El diagrama RMDM es el esquema completo del dominio y de la navegación de la aplicación y se obtiene como resultado de las tres primeras etapas de diseño en la metodología RMM.

Etapas 1: Representar los objetos del dominio con la ayuda del modelo entidad-relación ampliado con relaciones asociativas (aquellas que permiten representar caminos navegacionales entre entidades puestos en evidencia en la fase de análisis).

Etapas 2: Determinar la presentación del contenido de las entidades de la aplicación así como su modo de acceso. El esquema obtenido como resultado de esta etapa se denomina esquema (E-R+). Se trata de un esquema entidad-relación en el que cada entidad ha sido reemplazada por su esquema de entidad. Un esquema de entidad está constituido por nodos (los trozos o *slices*) unidos por relaciones estructurales.

Etapas 3: Definir los caminos de navegación inducidos por las relaciones asociativas del esquema (E-R+). A continuación, es posible definir estructuras de acceso de alto nivel (agrupaciones), lo que permite dotar a la aplicación de accesos jerárquicos a niveles diferentes de los trozos de información. El esquema RMDM resultante se obtiene añadiendo al esquema (E-R+) las agrupaciones y caminos navegacionales definidos en esta etapa.

2.5.3 Limitaciones que presenta originalmente RMM.

La metodología RMM original cuando surgió tenía algunas limitantes, a continuación se muestran las tres fundamentales:

- La imposibilidad de especificar qué información se debe mostrar como ancla, es decir qué texto o imagen aparece como hiperenlaces.
- RMM sólo permite agrupar atributos de una única entidad, es decir un slice no puede contener atributos de varias entidades.
- Por último, no se permite que un slice contenga además de atributos, estructuras de acceso. Es decir, para llegar a una estructura de acceso (índice o visita guiada) uno debe de atravesar un enlace.

2.5.4 Propuestas de mejoras a RMM Original.

Sus autores Balasubramanian, Bieber e Isakowitz, analizaron la estructura de navegación en RMM y propusieron añadir 3 nuevos tipos de *slices*:

- **slices mínimos** (la mínima parte de una entidad que es necesaria para identificar uno de sus elementos y que se utilizará como ancla).
- **slices híbridos** (que permiten combinar atributos de diferentes entidades y estructuras de acceso).
- **m-slices** (que permiten combinar primitivas de acceso con otros slices de otras entidades para crear un m-slice).

Se propone crear la estructura de la navegación no en base a las entidades sino a los slices, es decir a conjuntos de atributos.

2.5.4.1 Slice mínimo, slice híbrido y m-Slice.

Debido a la necesidad de extender el modelo RMDM se añaden estos nuevos slice para hacerlo realmente utilizable:

- El **slice mínimo** (minimal slice), que representa el conjunto de atributos de una entidad que permite que el usuario pueda identificar claramente las diferentes instancias de la misma. Este slice mínimo se utilizará como ancla a instancias concretas de una entidad. Por ejemplo si se accede a una entidad mediante una primitiva índice, lo que se desea es que el usuario vea la lista de los identificadores de los diferentes registros, ya que muy probablemente esa información le sea del todo inútil. Se utilizará el slice mínimo para tal fin.
- Los **slices híbridos** (hybrid slices) que permiten combinar atributos de diferentes entidades y estructuras de acceso. En la definición original de RMM, los slices se formulaban únicamente por atributos de una misma entidad. Con este añadido se permite que en una pantalla

aparezcan informaciones de más de una entidad. Con estos tipos de slices se consiguen unas estructuras más dinámicas tanto de navegación como de organización de la pantalla.

- Lo que se propuso es un nuevo tipo de slice, el que se denomina **m-slice**. La *m* viene de las muñecas matrjeska, las típicas muñecas rusas que se meten una dentro de la otra. Por tanto, un m-slice es un nuevo tipo de slice que permite solventar los tres aspectos. Así un m-slice podrá contener atributos de la entidad a la que pertenece, además de otros slices (m-slices posiblemente) formadas con atributos de otras entidades, unido todo ello por primitivas de acceso.

2.5.5 Otras propuestas de mejoras para la Metodología RMM.

A partir de las carencias de RMM e incorporando la novedad de los slices mínimos e híbridos, Lopistéguy, Losada y Dagorret analizan el modelo RMM y encuentran algunas carencias, por lo que proponen la creación de unas nuevas estructuras de navegación que doten de una mayor flexibilidad al modelo RMM, a la vez que crean una serie de primitivas de acceso nuevas.

- **Navegación jerárquica:** al tener varias relaciones 1: N encadenadas, se permite navegar desde cualquier entidad a otra que esté por debajo de ella en la jerarquía. Estos enlaces inferidos, no extraídos directamente de una relación 1: N, se representarán con trazo discontinuo.
- **Navegación en relaciones N:M:** se permite navegar de un extremo al otro de la relación, pero teniendo en cuenta la entidad intermedia, cuyos atributos deberán incluirse en un slice híbrido. Para representar un enlace de este tipo, se unirá la primitiva de acceso (índice, visita guiada,...) con la entidad intermedia.
- **Navegación múltiple:** se crean unas nuevas primitivas que permiten el acceso múltiple de una entidad a otra, seleccionando un elemento de una tercera entidad de la que la entidad destino es parte. En el enlace quedará especificado qué entidad es la origen, cuál la destino y cuál la tercera. Esta navegación es especialmente apropiada en estructuras todo-parte.
- **Acceso aleatorio:** permite acceder a un elemento de forma aleatoria, sin saber exactamente a cuál.
- **Acceso simultáneo:** permite representar el acceso a todos los elementos de una entidad a la vez. Por ejemplo, si se está en un Seminario, con esta primitiva se accede a todos sus ponentes simultáneamente, todos a la vez, y no mediante un índice o uno tras otro en una visita guiada.

2.6 Etapas de la Metodología RMM.

Como toda metodología se debe comenzar con un estudio de viabilidad del cual se obtiene un (Documento de Viabilidad) y un análisis de los requerimientos tanto de la información, como de la navegación, obteniendo un (Documento de Requerimientos). Además debe hacerse una selección del hardware y software que se necesitará.

Los autores de la metodología RMM no hacen más referencias respecto a esta fase inicial, por lo que es aplicable cualquier metodología típica de la ingeniería de software tradicional.

2.6.1 Etapa 1: Diseño del Modelo Entidad-Relación

En la primera etapa comienza el diseño y se representa el dominio de información de la aplicación mediante un diagrama entidad-relación. Este estado del proceso de diseño representa un estudio de las entidades y relaciones relevantes en el dominio de información. Dichas entidades y relaciones forman la base de la aplicación, y muchas de ellas aparecerán en la aplicación como una red de nodos y enlaces. El objetivo en el diseño es hacer explícitos los enlaces entre objetos (que evidentemente se concretarán en entidades), ya que estos son el vehículo principal para que el usuario pueda acceder a la información. Un análisis del dominio utilizando la aproximación E-R ayuda a identificar las relaciones importantes sobre las cuales puede basarse la navegación. Del mismo modo, si se necesita un enlace entre entidades, es necesaria la existencia de la relación en el diagrama E-R. Esta aproximación presenta el inconveniente de su dificultad de aplicación a dominios complejos, puede llevar a diseños “aberrantes” desde el punto de vista E-R.

Una vez identificadas todas las entidades se procede a identificar las relaciones existentes entre ellas. Relaciones que pueden ser de uno a uno (1:1), uno a muchos (1: N) y muchos a muchos (N: M) las cuales se desglosarán en dos relaciones 1: N, como muestra el ejemplo:

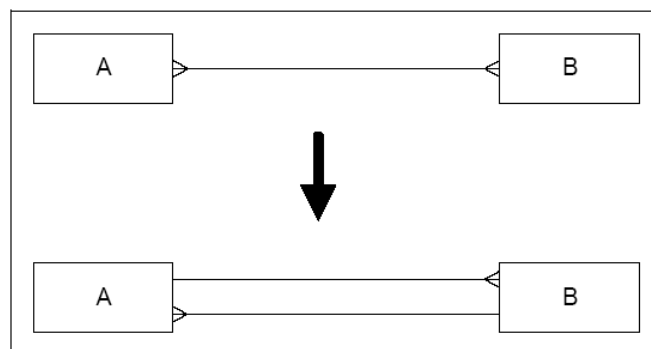
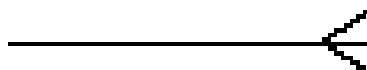


Figura 12. Representación de las relaciones de asociación muchos a muchos en el diagrama E-R.

Como se puede ver en la figura anterior las relaciones se representan con el símbolo:



El objetivo de esta etapa es hacer explícitos todos los enlaces entre objetos. Más tarde, las relaciones darán lugar a la navegación. Así, una relación especificará un camino en la navegación. El resultado final de esta etapa es la elaboración del diagrama entidad-relación.

2.6.2 Etapa 2: Diseño de slices

Esta etapa es exclusiva para aplicaciones hipertexto y determina que información de las entidades elegidas se presentara al usuario, y como podrá acceder a la misma (es decir, es algo así como una proyección del álgebra relacional). [14] Implica dividir la entidad en slice o trozos significativos, y organizar estos en la red de hipertexto. En su vertiente más sencilla, toda la información de una entidad puede mostrarse en una pantalla. De manera alternativa, la información puede dividirse en unidades significativas que pueden presentarse de manera interrelacionadas. La organización de entidades en slice se denomina etapa de diseño de slices, cuyo resultado es un diagrama de slices. Todos los diagramas de slices contienen un slice cabeza, que es el que se muestra al acceder a dicha entidad (algo así como la vista por defecto en HDM). El diagrama de entidades también modela la navegación entre slice a través de enlaces unidireccionales y bidireccionales. Estos enlaces que representan conexiones entre slices se denominan enlaces estructurales, de manera consistente con la notación HDM. Los enlaces estructurales se diferencian de los enlaces de asociación (relaciones) en que los primeros conectan piezas dentro de la misma instancia de entidad, mientras que los segundos conectan diferentes instancias de entidad, pertenecientes por lo general a distintas clases de entidades.

Desde un punto de vista navegacional, hay una importante razón para distinguir entre estas dos conexiones. Cuando un usuario atraviesa un enlace de asociación, el contexto de información cambia, por ejemplo de un profesor a un curso. Sin embargo, cuando se navega un enlace estructural el contexto de información se mantiene dentro de la misma entidad. Para diferenciar ambos enlaces los enlaces estructurales se dibujan con líneas sólidas, mientras que los de asociación se dibujan con líneas discontinuas.

El resultado final de esta etapa es un diagrama E-R enriquecido, denotado por (E-R+). Se trata de un esquema entidad-relación en el que cada entidad ha sido reemplazada por su esquema de entidad. Un esquema de entidad está constituido por nodos (los trozos o *slices*) unidos por relaciones estructurales.

A continuación se muestra un ejemplo de esquema de entidad en la figura 5.

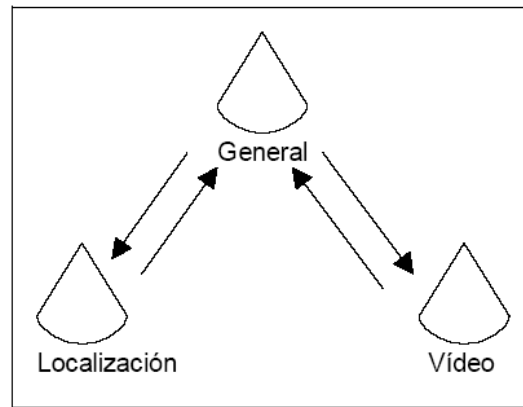


Figura 13. Representación de los enlaces estructurales entre slice en la metodología RMM.

2.6.3 Etapa 3: Diseño de la Navegación.

En esta etapa se diseñan los caminos que permitirán la navegación por la aplicación. Cada relación de asociación del diagrama (E-R+) se analiza, y si se desea hacerla accesible para la navegación, se reemplaza por una primitiva de acceso RMDM. Como la metodología RMM se utiliza en dominios donde la información se actualiza con relativa frecuencia, todos los accesos navegacionales se especifican en términos genéricos. Esto quiere decir que los enlaces se especifican mediante propiedades de entidades y relaciones. Esto es posible gracias a los índices condicionales, visitas guiadas condicionales, y visitas guiadas indexadas condicionales.

Por último solo falta incluir las estructuras de acceso a las slice por defecto (o a otros) de las entidades. Al final de esta fase el diagrama (E-R+) se ha convertido en un diagrama RMDM, que describe todas las estructuras de acceso que serán implementadas en el sistema.

En general, se preferirá una visita guiada condicional a un índice condicional cuando el número de instancias sea pequeño (menor que 10) y no exista un campo índice que pueda ayudar a los usuarios. Por el contrario, si el número es grande, se usará índices.

Las visitas guiadas indexadas son un híbrido, usado frecuentemente cuando hay un índice y se desea una navegación entre las instancias.

Por defecto se accede a través de una primitiva de acceso al slice principal (head). En caso contrario, debe especificarse, etiquetando el nombre de la estructura de acceso.

Aunque siguiendo al pie de la letra se debería especificar un enlace “de vuelta” si fuera necesario, entre las entidades que tienen una primitiva de acceso; se considera que siempre este enlace debe estar presente y por tanto no se representa en el diagrama, por considerarse redundante.

Además en esta fase se establece una jerarquía de menús, utilizando la primitiva de grupo o menú. Como regla general se deben evitar grandes profundidades en la jerarquía, ya que desorientan al usuario.

El resultado final de esta etapa es el diagrama RMDM el cual describe no sólo la información referente a las clases de objetos, sino también a la navegación entre ellos.

2.6.4 Etapa 4: Definir el Protocolo de Conversión.

En esta etapa hay que traducir las primitivas de acceso RMDM a primitivas de acceso del lenguaje o sistema que vaya a implementar el diseño. En el paso de diseño del protocolo de conversión, se crean un conjunto de reglas de conversión para transformar cada elemento del diagrama RMDM en un objeto de la plataforma destino. Se propone un paso manual para realizar dicha conversión.

2.6.5 Etapa 5: Diseño de Interfaz de Usuario

Esta etapa hace referencia al diseño de cada una de las pantallas de la aplicación.

Diseño gráfico de todas las pantallas correspondientes a cada uno de los slices que se han obtenido en la etapa 2.

Siguiendo la metodología, se tendrá una pantalla para cada uno de los objetos del modelo RMDM.

En concreto hay cuatro aspectos importantes sobre los que se centra:

- La información de los slices encontrados en la segunda fase de la metodología.
- Las estructuras de acceso que identificamos en la tercera fase.
- Herramientas de navegación, como backtracking y similares.
- Apariencia de las anclas, cómo han de aparecer los índices y dónde deben situarse las citadas herramientas de navegación.

Como ya se cita en los cuatro puntos a tener en cuenta, un aspecto muy importante que también hay que decidir es la representación de los anclas de los enlaces. Puede ser una buena solución el identificar cada una de las entidades mediante un icono.

Otro aspecto a resaltar es la conveniencia de diferenciar los enlaces estructurales (internos a una entidad: enlaces entre slices) y los enlaces entre diferentes entidades.

2.6.6 Etapa 6: Diseño del Comportamiento en Ejecución.

En la etapa de diseño del comportamiento en ejecución se determina si va a haber historial de selección de enlaces, vuelta atrás, y mecanismos navegacionales. Además se decide si el contenido de los nodos y de los enlaces se generará antes de la ejecución o dinámicamente en tiempo de ejecución.

2.6.7 Etapa 7: Construcción y Prueba.

Finalmente en la etapa de construcción y prueba se llevan a cabo las actividades tradicionales de ingeniería del software, prestando especial atención a la prueba de los caminos navegacionales.

2.7 Análisis Crítico del modelo RMDM.

El modelo RMDM supone una organización relacional bien estructurada que va a determinar a posteriori el esquema navegacional de la aplicación. La razón estriba a que la estructuración de los contenidos va a ser tan fuerte que difícilmente se va a poder obviar a la hora de especificar los enlaces entre estos. Es muy adecuado para proporcionar una visión hipertextual de bases de datos, este modelo adolece de capacidades para representar ideas tan fundamentales en hipermedia como la de contexto o sincronización. Este tipo de modelo tampoco se ajusta bien a cualquier aplicación cuyos contenidos no puedan organizarse a través de un esquema entidad-relación [16]. De hecho, autores como Balasubramanian en diversas ocasiones han optado por no utilizar RMDM, y ni siquiera OOHDM, en el diseño de un sistema a gran escala debido a la organización relacional que estos imponen.

2.8 Herramienta RMCASE soporte de la Metodología RMM.

La herramienta sólo soporta algunas fases de la metodología: Diseño E-R, diseño de slices, diseño de navegación y comportamiento en tiempo de ejecución. Esta pensada también para soportar el desarrollo de interfaz de usuario ya que aparece un botón para ello pero no está implementado aún.

2.8.1 Diseño E-R

El resultado de esta fase de diseño es la elaboración de un diagrama entidad-relación. La herramienta soporta la creación de este diagrama, es decir la creación de entidades, atributos y relaciones.

La creación de entidades es sencilla. Cada nueva entidad creada contiene en su interior dos iconos que permiten la creación de atributos para esa entidad y relaciones entre distintas. Todos los elementos creados (entidades, relaciones y atributos) se pueden seleccionar, editar y borrar (suprimir cuando están seleccionados). Al borrar una entidad se eliminan también las relaciones que mantiene con el resto de las entidades así como los atributos que contiene. Todos los elementos se pueden mover cuando están seleccionados. Los elementos donde aparece la etiqueta correspondiente a la relación sólo tienen un grado de libertad ya que deben permanecer a la misma distancia de las dos entidades a las que relacionan.

2.8.1.1 Creación de atributos

La creación de atributos se hace seleccionando la elipse en el interior de la entidad. No hay límite en el número de atributos que se pueden crear para una entidad dada pero debe existir al menos un atributo clave.

En la creación de un atributo se puede especificar si es o no el atributo clave y de qué tipo es. El tipo de atributo que se especifica en tiempo de creación no se puede modificar al editar el atributo, la clave sí.

Existe un conjunto de valores por defecto (texto, bitmap, media) para el tipo de atributo. Esta lista es bastante limitada ya que no contienen todos los tipos de medios en una aplicación (ejemplo audio, animación, etc.) y no es consistente ya que algunos tipos vienen especificados por el tipo de medio (texto), otros por el formato (bitmap) en lugar del tipo correspondiente (imagen) y una categoría más para el resto de los medios.

Sólo puede existir un atributo clave dentro de cada entidad. La herramienta se encarga de que no haya más de uno. Si se selecciona un atributo como atributo-clave y más adelante se crea un nuevo atributo-clave la clave permanece únicamente en el último atributo al que se le ha asignado y desaparece de todos los demás. Sin embargo si no se le ha asignado la clave a ningún atributo el error no se muestra hasta la etapa de simulación cuando se intenta acceder al contenido de la entidad correspondiente.

2.8.1.2 Creación de relaciones

Las relaciones se crean seleccionando el rombo que hay dentro de una entidad y arrastrando hasta otra. En la creación de una relación se le puede asignar un nombre y la cardinalidad (1:1 o 1:N) ambos valores son modificables en tiempo de edición.

Las relaciones N:M se implementan mediante dos relaciones 1:N y 1:M. Para el resto de las relaciones entre entidades en principio debería bastar una sola línea (1:1 o 1:N) pero como los enlaces de navegación entre entidades se crean a partir de las relaciones de este diagrama es necesario establecer la relación en ambos sentidos.

Las relaciones 1 a 1 en el diagrama E-R se convierten en enlaces unidireccionales en el diagrama de navegación y las relaciones 1:N se convierten en estructuras de acceso (índices por defecto) que pueden modificarse en el diagrama de navegación para convertirse en circuitos guiados o circuitos guiados e indexados.

2.8.1.3 Modificaciones del diagrama E-R

Una vez creado el diagrama de navegación las modificaciones sobre las relaciones en el diagrama E-R no se actualizan adecuadamente en el de navegación, es decir si se convierte una relación 1:N en una relación 1:1 en el diagrama E-R, el diagrama de navegación seguirá apareciendo una estructura de acceso (índice, circuito guiado o circuito guiado indexado) en lugar de un enlace unidireccional que sería el equivalente adecuado para la relación 1:1.

2.8.1.4 Otras consideraciones del diagrama E-R

Para facilitar la navegación dentro del contexto de la herramienta es necesario a veces que los atributos dentro de una entidad tengan el mismo nombre que otras entidades. Esto facilita la creación de condiciones en las estructuras de navegación.

2.8.1.5 Resumen Diagrama E-R

El diagrama entidad-relación es uno de los diagramas mejor implementados en la herramienta. Se echa de menos la posibilidad de copiar y pegar atributos, entidades o relaciones de iguales características así como la posibilidad de deshacer acciones o en su defecto que pida confirmación para algunas acciones "peligrosas" como borrar una entidad.

La forma de editar los distintos elementos me parece adecuada salvo las limitaciones en los tipos de entidades y el hecho de que sólo se puedan especificar en tiempo de creación, así como la limitación en la movilidad en el diagrama de las etiquetas de las relaciones (se obliga a que sea equidistante de las dos entidades que relaciona).

En cuanto a la integración con el resto de los diagramas se puede mejorar, en particular la integración con el diagrama de navegación es bastante deficiente ya que no siempre se reflejan las modificaciones realizadas en la fase de diseño E-R. Las características de la navegación obligan también a incorporar algunas cosas que no son necesarias en un modelo puro E-R como por ejemplo el hecho de crear relaciones de "vuelta" para las relaciones 1:1 y 1:N o la creación de atributos en una entidad con el mismo nombre que las entidades con las que se relaciona para poder crear condiciones en los índices. La falta de integración es una de las principales deficiencias ya que dificulta la obtención de feed back y obliga a que el diagrama E-R esté perfectamente diseñado antes de pasar a la fase de navegación.

2.8.2 Modelo E-R+ (Diseño de slices)

El modelo E-R extendido consiste en diseñar el contenido de las entidades mediante slices que agrupen distintos atributos.

2.8.2.1 Creación de slices

Para crear slices se selecciona una entidad y se activa el botón para diseño de slices. Al seleccionarlo se crea un slice por defecto con el mismo nombre que la entidad, pero sin ningún atributo.

Es necesario seleccionar alguno de los atributos de la entidad para que formen parte del slice y se muestren cuando esa entidad sea enlazada desde otra. El hecho de que el slice esté vacío no da ningún error en tiempo de ejecución, cuando se seleccionan aparece una ventana vacía con el nombre de la entidad y del slice con los atributos que contiene.

El slice principal de una entidad es el que actúa como ancla cuando se enlaza a esa entidad y se muestra su contenido. En teoría el slice principal se indica con un asterisco en el nombre.

El slice que la herramienta crea por defecto contiene el asterisco, es decir es el slice principal de la entidad. Se pueden crear otros slices simplemente con doble click sobre un espacio libre en el diagrama.

La forma en que se asigna el slice principal no funciona en la herramienta ya que si se crea un nuevo slice y se le pone un asterisco no borra el asterisco del que se creó primero. El mecanismo debería ser similar a la forma en que se gestiona el atributo clave de una entidad, pero a diferencia de lo que ocurriría entonces en este caso la herramienta no lo soporta.

Además si se borra el asterisco de un slice y lo se pone en otro, el slice que actúa como principal sigue siendo el que la herramienta creó por defecto. Por tanto debemos utilizar como slice principal el que crea la herramienta aunque podemos modificar su nombre y los atributos que contiene. Si se permite la creación de otros slices (no principales) que agrupen diferentes atributos de una entidad.

2.8.2.2 Relaciones entre slices

Cuando se crea un nuevo slice se crean por defecto relaciones entre todos ellos que pueden ser editadas (doble click) para cambiar su nombre, seleccionadas (click) y eliminadas (click + supr).

La creación de enlaces entre slices es indirecto, sólo está permitido cuando se crea un nuevo slice. Al hacerlo se crean automáticamente relaciones con el resto de slices de la misma entidad. Si luego borramos alguna de estas relaciones no pueden volver a ser creadas a menos que se borre y se vuelva a crear el slice completo.

En tiempo de ejecución cuando se llega a una entidad se muestra el contenido del slice principal (marcado con asterisco) y a la derecha de la ventana aparece una lista con los slices que mantienen relaciones con el principal. El resto de los slices son inaccesibles.

En teoría se puede mostrar como ancla de una entidad un slice que no sea el principal, para ello habría que llamar al enlace asociativo que lleva a esa entidad con el mismo nombre que el slice (no principal) que se quiere utilizar como ancla. Esta característica del modelo tampoco está soportada en la herramienta. Por lo tanto en la práctica sólo son accesibles aquellos slices que estén relacionados con el principal (siendo siempre el slice principal el que la herramienta crea por defecto). En el desarrollo de la aplicación se ha decidido dejar todos los enlaces que se crean por defecto entre los slices de manera que cualquier vista de los atributos de una entidad concreta sea accesible desde todos los demás. O dicho de otro modo todos los atributos de una entidad son accesibles en todo momento.

2.8.2.3 Resumen E-R+

El diagrama de slices es bastante deficiente, ya que no funcionan los mecanismos para convertir un slice en el slice principal de una entidad. Ni tampoco los mecanismos que permiten que el ancla de una entidad sea un slice distinto al principal.

Además aunque el mecanismo de creación de slices y atributos es similar al que sigue en el diagrama E-R para la creación de entidades y atributos, el mecanismo de creación de relaciones estructurales (entre slices) no es consistente con la forma en la que se crean relaciones asociativas (entre entidades). A mi entender el mecanismo de creación indirecta de slices no tiene sentido. Sí puede resultar adecuado que se creen por defecto relaciones entre todos los slices pero no tiene sentido que una vez borrados no exista un mecanismo de creación análogo al del diagrama E-R y es poco práctico que la única forma de crear una relación estructural sea creando un nuevo slice.

En cuanto a la integración con el diagrama E-R es buena en término de actualización de los atributos disponibles dentro de una entidad. El principal problema es el que se mencionó con las ventanas modales, no recuperan el foco cuando se minimiza y se vuelve a ampliar la aplicación.

Para acceder desde el modelo E-R al diagrama de slices debe haber una entidad seleccionada pero aquí también hay un pequeño defecto. Si se selecciona la entidad se activa el botón del

diagrama de slices pero si se edita una entidad y luego se cierra la ventana de la edición de la entidad aparece seleccionada pero el botón del diagrama de slices está desactivado.

2.8.3 Modelo de navegación

En el diagrama de navegación se crean automáticamente algunas primitivas de acceso a partir de las relaciones del modelo E-R y se permite la creación de agrupaciones así como enlazar las agrupaciones creadas con distintas entidades. Se pueden editar las nuevas primitivas de acceso creadas (es decir los grupos) y todas las relaciones 1:N entre entidades.

El principal problema la primera vez que te enfrentas al diagrama es que a diferencia del resto de los diagramas no todos los objetos son activos. Pienso que esto debería hacerse explícito de alguna forma. La forma más habitual sería colorear los elementos no seleccionables en un tono más claro o dibujar las líneas o todo el elemento no seleccionable en gris, etc.

2.8.3.1 Relación entre entidades

Las relaciones entre las distintas entidades se crean directamente a partir del diagrama E-R pero no se actualizan directamente. Además el proceso de actualización se produce de manera distinta cuando la ventana de navegación está abierta o cerrada en el momento en que se produce la modificación.

Los enlaces 1:1 en el diagrama E-R se convierten en enlaces unidireccionales con el mismo nombre que el que tenía en el diagrama E-R y no son editables. Por el contrario los enlaces 1:N se convierten por defecto en índices condicionales pero pueden modificarse y convertirse también en circuitos guiados o circuitos guiados e indexados. Para modificar los enlaces creados a partir de las relaciones 1:N simplemente hay que editar los enlaces haciendo doble click sobre la primitiva de acceso correspondiente (índice, circuito guiado o circuito guiado e indexado), en la ventana de edición también se puede modificar el nombre del enlace.

2.8.3.2 Creación de grupos

Los grupos se crean haciendo click con el ratón en un área vacía del diagrama. Es necesario que uno de los grupos creados se especifique como grupo de inicio para empezar la navegación pero en caso de no hacerse el error no se hace explícito hasta el momento de la simulación. No se puede empezar la navegación desde una de las entidades, sólo desde un grupo. Aunque pueden existir varios grupos sólo uno de ellos puede ser marcado como grupo de inicio, si se marca más de uno aparece como grupo de inicio el último que se marcó.

2.8.3.3 Relaciones entre grupos y entidades

Se crean pinchando sobre el grupo y arrastrando hasta la entidad que se quiera enlazar. En tiempo de ejecución cuando se pinche sobre el enlace se mostrarán los atributos de la entidad que aparezcan en el slice principal (si el enlace es uno a uno) o un nuevo índice (si el enlace es 1:N).

Aunque en teoría se puede conseguir que el ancla sea un slice distinto del principal en la práctica la herramienta no lo soporta. Para hacerlo bastaría con poner el nombre del enlace como el nombre del slice que se quiere que actúe como ancla. Los nombres de los enlaces unidireccionales hay que ponerlos en las relaciones 1:1 del diagrama E-R ya que se "mapean" directamente y no se puede modificar su nombre, sin embargo los nombres de los enlaces 1:N se pueden modificar directamente en el diagrama de navegación y se le puede dar un nombre diferente a la relación (diagrama E-R) y a la primitiva de acceso (diagrama de navegación).

2.8.3.4 Condiciones en las estructuras de navegación

La posibilidad de asociar condiciones a los índices da gran flexibilidad a la herramienta estas condiciones permiten comparar los valores de los campos de dos entidades distintas. Esto hace que sea necesario crear atributos en una entidad que coincidan por ejemplo con el nombre de otra. En el ejemplo que se ha creado la entidad receta contiene la etiqueta autor y también existe una entidad llamada autor que contiene un atributo llamado nombre. Esto nos permitiría crear un índice condicional entre el autor y las recetas que escribe que comparase el valor del atributo nombre de la entidad autor con el valor del atributo autor de la entidad receta y mostrara sólo las recetas cuyo autor coincida con el nombre de la instancia desde la que se ha activado el enlace. Se pueden crear condiciones más complejas agregando condiciones simples (comparaciones) mediante los operadores lógicos (AND, OR y NOT)

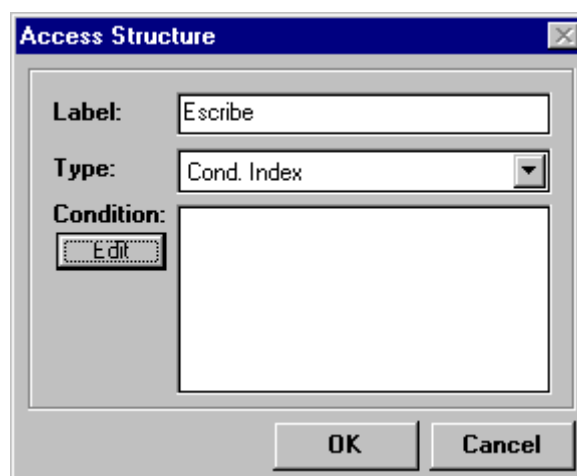


Figura 14. Edición de la estructura de navegación "Escribe".

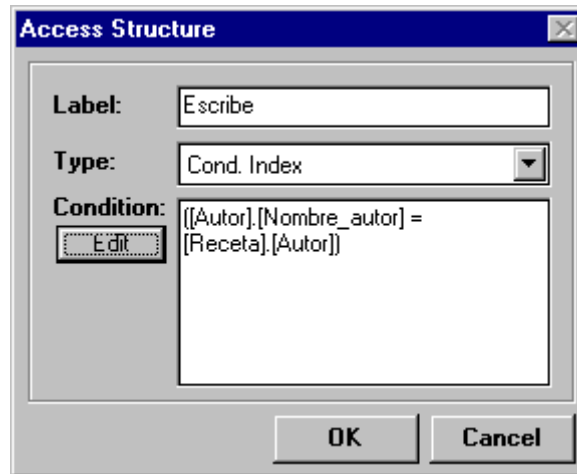


Figura 15. Edición de la condición en la estructura de navegación "Escribe".

2.8.3.5 Resumen diagrama de navegación

La principal deficiencia de este diagrama es la falta de integración con el diagrama E-R. Cuando se produce una modificación en el diagrama E-R no siempre se refleja en el diagrama de navegación. Además la forma en que se comporta la herramienta es diferente si la ventana de navegación estaba o no abierta durante la modificación. Esto es un problema para obtener feedback y para hacer modificaciones en tiempo de diseño.

Para que fuese más sencillo su uso sería conveniente hacer explícito qué elementos no están activos y proporcionar información sobre la forma de enlazar con slices distintos del principal dentro de una entidad. También sería útil la posibilidad de copiar y pegar los elementos gráficos del diagrama.

Existe además una inconsistencia en la forma en que se pueden editar los enlaces. No tiene sentido que se pueda modificar el nombre de los enlaces unidireccionales y no el de los bidireccionales. Aunque en realidad el nombre de los enlaces unidireccionales es más irrelevante porque no aparece en tiempo de ejecución mientras que el nombre de los enlaces 1:n aparece como título de la "página" de índice que se activa al seleccionarlos.

Personalmente pienso que por consistencia con lo que ocurre con los enlaces 1:1 los índices que se crean por defecto a partir de las relaciones 1:N deberían tener como nombre por defecto en el diagrama de navegación el mismo que se le dio a la relación 1:N en el diagrama E-R aunque luego se permitiese su modificación.

La principal ventaja del diagrama de navegación es la posibilidad de crear varias primitivas de acceso (índices, circuitos guiados y circuitos guiados indexados) y la posibilidad de asignarles condiciones en las que se combinen distintas reglas para crear estructuras de acceso más complejas.

2.8.4 Prototipado

La fase de prototipado funciona bastante bien, la generación del prototipo es bastante rápida. En el prototipo hay dos tipos de páginas: las de índice y las que contienen información.

- Las páginas de índice contienen un título que corresponde con el nombre de la estructura de navegación a partir del que han sido generadas y un conjunto de botones (uno por cada elemento del índice). La etiqueta de los botones es el nombre de las entidades (cuando el índice relaciona una agrupación con distintas entidades) y es el nombre del atributo clave de la entidad (cuando se trata de relaciones entre dos entidades).
- Las páginas de información muestran el valor de un conjunto de atributos de una entidad concreta y se dividen en 4 áreas.
 - En el área principal se muestra un título y un conjunto de valores. El título consiste en el nombre de una entidad y el nombre del slice cuyos atributos se están mostrando. El conjunto de valores son los valores concretos de los atributos que agrupa el slice que aparece en el título.
 - En el margen derecho se muestran un conjunto de botones cuyas etiquetas son los nombres de otros slices de la misma entidad que se está mostrando. Cada uno de estos botones enlazan con la página de información en donde se muestra el contenido del slice que indica su etiqueta. No se muestran todos los slices de la entidad sino únicamente aquellos que estén relacionados (en el diagrama de slices) con el que se está mostrando en el área principal.
 - En el margen inferior se muestra un conjunto de botones que enlazan directamente con otras entidades (si la relación establecida es uno a uno) o con páginas de índice (si la relación es uno a muchos). La etiqueta de esos botones es el nombre del slice principal de la entidad con la que enlazan.
 - En la esquina inferior izquierda al igual que ocurre en las páginas de índice hay un par de botones que permiten en cualquier momento cerrar la ventana de prototipo o acceder a la página de inicio de la aplicación.

El principal problema del prototipo generado es que una vez que se está en una página no queda claro desde qué punto de la aplicación se ha llegado a ella. Únicamente existen mecanismos de vuelta atrás en el caso de los circuitos guiados que permiten pasar de una instancia a otra de forma secuencial hacia delante y hacia atrás (pero no cíclica).

2.9 Evaluación de RMCASE como herramienta de software.

2.9.1 Utilidad:

2.9.1.1 Cumplimiento de los objetivos propuesto por la herramienta.

En el epígrafe anterior se ha realizado un análisis de como la herramienta RMCASE soporta la metodología RMM, se ha visto que la herramienta sólo es aplicable en algunas etapas del desarrollo y no siempre se consigue el objetivo perseguido. Además desde el punto de vista cognitivo no soporta adecuadamente los diferentes estados mentales por los que atraviesa el diseñador en los procesos de creación, actualización y observación y evaluación de la aplicación. Por lo que uno de los principales problemas que presenta la herramienta es la falta de integración entre los distintos diseños que dificulta la realización de modificaciones en tiempo de diseño.

2.9.1.2 Capacidad de ser usado

La herramienta presenta algunos problemas. Por ejemplo existen ventanas modales (hasta que no se cierran no devuelven el control a la ventana principal desde la que fueron lanzadas), que no recuperan el foco al minimizar y volver a maximizar la ventana de aplicación. Un ejemplo ilustrativo es la ventana de diseño de slices. Esta ventana se lanza desde la ventana de diseño E-R. Si una vez abierto el diseño de slices se cambia de aplicación (ejemplo se comienza a editar con Word) y se regresa luego a la herramienta la ventana que mantiene el foco es la ventana principal (la del diagrama E-R) y no la ventana modal (diseño slices). Sin embargo en el diagrama E-R no se puede realizar ninguna acción ya que al no cerrarse la ventana de diseño de slices no se le ha devuelto el control, sin embargo como esa ventana no tiene el foco no se muestra.

Este mismo efecto aparece en todas las ventanas modales (ej. al editar una entidad, un atributo, una relación, etc.) Si maximizamos la aplicación desde la barra de herramientas de Windows no aparece la ventana modal, para activarla es necesario acceder a ella desde otro sitio (ejemplo cambiando de aplicación mediante alt+tab)

2.9.1.3 Facilidad de aprendizaje:

La principal dificultad en el aprendizaje de la herramienta radica en que carece de mecanismos de ayuda, aunque existen algunos documentos que describen las etapas de la metodología, no son suficientes ya que la herramienta no soporta todas las etapas y hay iconos propios de la herramienta que no han sido explicados.

2.9.1.4 Eficiencia en el uso

La creación de los distintos diagramas es rápida y la generación de los prototipos también es rápida. El principal problema es que no se actualizan adecuadamente los diferentes diagramas de modo que resulta difícil. Por ejemplo si se crea un grupo de entidades y las relaciones que tienen entre ellas y luego se borra el grupo no hay forma de borrar las relaciones que el grupo tenía con las entidades y es necesario empezar desde el principio.

2.9.1.5 Fácil de recordar

Aunque la herramienta no es sencilla de aprender una vez aprendida es fácil de recordar ya que hay pocas acciones permitidas en cada uno de los diagramas y la estructura en la que se crean es muy similar en todos ellos.

Para crear relaciones (dentro del modelo E-R o dentro del de navegación) siempre hay que pinchar sobre el triángulo que aparece dentro de cada elemento, del mismo modo para crear atributos siempre hay que hacer click en la elipse que aparece en los distintos elementos (Para crear los atributos de una entidad o para decir que atributos están dentro de un slice).

Aunque también existen algunas inconsistencias que dificultan recordar el aprendizaje como por ejemplo el hecho de que la creación de relaciones entre slices sea diferente a la creación de cualquier otro tipo de relación sin ningún motivo aparente.

2.9.1.6 Errores en la utilización de la herramienta.

Al principio la herramienta se presta a errores por parte del usuario ya que el funcionamiento de la herramienta debe aprenderse mediante prueba y error.

Algunos errores son debidos a la forma en la que está hecha la herramienta, por ejemplo la primera vez que se intenta crear una relación entre slices se intenta hacer del mismo modo que se crean las relaciones entre entidades ya que no hay nada que indique que hay que hacerlo de otro modo pero no es posible hacerlo así.

Otro ejemplo de error frecuente es que en el diagrama de navegación se intenten mover o editar algunos componentes que no son editables ya que en el resto de los diagramas todos los objetos lo son y no hay nada en el interfaz gráfico que indique que aquí no lo sean.

Algunos de estos errores pueden corregirse pero otros no.

En este sentido uno de los principales problemas de la herramienta es que permite crear elementos dentro del diagrama que luego no se pueden borrar.

También da lugar a confusión el hecho de que al instanciar los elementos y las entidades no avisan de posibles inconsistencias que se reflejan en tiempo de ejecución (ejemplo cuando aparece algún

elemento de la entidad como nulo) o simplemente da errores cuando se abre por segunda vez el diagrama de navegación.

2.9.1.7 Satisfacción del usuario

Los principales factores de insatisfacción por parte del usuario son la falta de mecanismos de ayuda para el aprendizaje de la herramienta. El hecho de que no se especifique claramente para qué tipo de aplicaciones es aplicable la metodología y en casos es útil el uso de la herramienta. El hecho de que no se pueda salvar el proyecto con diferentes nombres para aprovechar el contenido de una aplicación anterior. El que no se puedan borrar algunos de los elementos creados en los diferentes diagramas y que esto obligue a construir desde cero la herramienta, etc.

2.9.2 Coste

Con respecto al coste en recursos (tiempo), no al coste económico ya que se puede trabajar sobre una versión de evaluación.

El coste en tiempo es bastante elevado ya que puesto que la herramienta no permite borrar es necesario tener diseñados los distintos diagramas previamente en papel y tampoco reduce el coste al reutilizar ya que los proyectos no pueden ser salvados con otro nombre, sino que hay que modificarlos completamente o hacer una copia de fichero (fuera de la herramienta) o de los datos que "pueblan" la aplicación.

Las únicas formas de reutilización aplicable están fuera de la herramienta: copiar y guardar el fichero de proyecto en otra carpeta y abrirlo de nuevo para modificarlo (fichero.rmp). Lo mismo para los datos que pueblan la aplicación (fichero.ldb). Tampoco se pueden copiar y pegar elementos dentro de un diagrama.

2.9.3 Compatibilidad

Cómo no está implementada la fase de la herramienta que genera código HTML no es compatible con otros editores u otras herramientas de desarrollo de aplicaciones hipermedia como macweb ya que todos los ficheros que crea tienen formato propietario (ficheros.rmp y ficheros.ldb). En cuanto a la compatibilidad para distintas plataformas, se tiene una versión de la herramienta solo para Windows.

2.9.4 Fiabilidad

La herramienta es poco fiable ya que cualquier problema en el desarrollo de la aplicación hipermedia hace que se cierre la herramienta automáticamente y aunque los datos que guarda siempre son los más actualizados si contienen errores no permite guardar las vistas y es necesario empezar de nuevo desde cero.

2.10 Conclusiones

En este capítulo se abordaron temas relacionados con las fases de la metodología en la cual se puede ver que a metodología RMM permite hacer explícita la navegación al hacer el análisis, lo que permite, teóricamente, obtener una navegación más estructurada e intuitiva, y lo hace de una forma muy sencilla, añadiendo unas primitivas a un modelo entidad-relación tradicional. El concepto de slice es muy útil. También es interesante la primitiva de grupo, que permite mostrar la jerarquía de menús.

RMM representa el primer caso en el que se crea una metodología completa definiendo las distintas fases y no únicamente un modelo de datos. Además, se basa en un modelo de datos relacional, ajustándose así a la gran mayoría de las aplicaciones existentes. Sin embargo, los mecanismos de acceso a la información son excesivamente simples y valen para un problema con pocas entidades, pero el modelo se queda corto si hay gran número de ellas.

CAPÍTULO 3

Aplicación de RMM al Caso de Estudio.

3.1 Introducción

En este capítulo se aplica la metodología RMM al caso de estudio Multimedia Topografía, que corresponde a un software educativo que se utilizará como apoyo en el proceso de enseñanza - aprendizaje de la asignatura de Topografía de la Carrera de Ingeniería Civil en la Facultad de Ingeniería Civil del Centro Universitario José Antonio Echeverría (CUJAE).

3.2 Caso de estudio.

La Facultad de Ingeniería Civil del Centro Universitario José Antonio Echeverría (CUJAE) tiene entre sus asignaturas fundamentales la Topografía. Se presentan problemas con el nivel de asimilación de los estudiantes en las diferentes materias de la asignatura así como en el uso de los instrumentos de medición y nivelación del terreno. En este capítulo se realiza la aplicación de la metodología RMM al SWE Multimedia Topografía que servirá de apoyo en el proceso enseñanza - aprendizaje de los estudiantes de dicho centro universitario. En el proceso de análisis y diseño utilizando la metodología se debe garantizar que el SWE abarque información referente a la carrera de ingeniería civil, el programa de la disciplina de topografía que se imparte, la información del los instrumentos de medición y nivelación del terreno, los contenidos referentes a la asignatura de topografía y los métodos de comprobación, ejercicios, etc. También toda la bibliografía perteneciente a los contenidos de la asignatura y a los documentos que complementen el estudio de la misma.

3.3 Aplicación de la metodología RMM al caso de estudio Multimedia Topografía.

Como se expuso en el capítulo anterior la metodología cuenta con 7 etapas:

1. Diseño del modelo entidad-relación.
2. Diseño de los slice.
3. Diseño de la navegación.
4. Definir el protocolo de conversión.
5. Diseño de la interfaz de usuario.
6. Diseño del comportamiento en ejecución.
7. Construcción y prueba.

A continuación se procede a la aplicación de las etapas de la metodología RMM al SWE Multimedia Topografía, aunque se presta especial atención al diseño de los mecanismos de acceso.

3.3.1 Etapa 1: Diseño del Modelo Entidad-Relación (E-R).

Esta primera etapa suele ser de peso y de gran importancia para el período de diseño de la aplicación pues es donde se identifican las entidades correspondientes en la que se va a desglosar la información y se le dedica mucha atención. Además se definen las relaciones entre las entidades según correspondan.

A continuación se listan ordenadas alfabéticamente y descritas todas las entidades de la aplicación:

1. **Bibliografía:** Guardará toda la bibliografía ordenada alfabéticamente. Se incluirán vínculos directos con la Web para la bibliografía que tenga vínculos a Internet o la Intranet; o la bibliografía que sea más bien solo vínculo.
2. **Conferencias:** Contiene todas las conferencias desglosadas por sumario, contenido. En cada registro de esta entidad hay una conferencia.
3. **Ejemplos:** Contiene todos los ejemplos desglosados por las conferencias al cual pertenecen.
4. **Ejercicios:** Guía de ejercicios renovables con sus respuesta como forma mas de auto evaluación del estudiante.
5. **Imágenes:** Todas las imágenes incluidas por el administrador de sistema. Estas imágenes estarán organizadas por nombre y número.
6. **Información de la carrera Civil:** Explicación amplificada generalizada de la carrera de Ingeniería Civil.
7. **Información del equipo:** Breve explicación de forma cualitativa del instrumento y forma de usarlo.
8. **Laboratorios:** Se incluyen todos los laboratorios establecidos por el profesor o los requeridos por la asignatura vinculados a la conferencia a la que pertenece.
9. **Otras multimedia:** Contendrá vínculos directos con las otras multimedia de la carrera. Esta entidad tiene un objetivo futurista, pensando que las otras asignaturas pasen sus contenidos a formato digital.
10. **Programa disciplina Topografía:** Una explicación amplificada generalizada de la las asignatura topografía.
11. **Prueba:** Consta de varias pruebas desglosadas por temas y con sus respectivos ejercicios.
12. **Talleres:** Una forma más de práctica y auto preparación del estudiante son los talleres y que no se puede excluir de la aplicación. Seria según las conferencias impartidas requerido por la asignatura.
13. **Temas:** El estudiante tendrá la posibilidad de navegar en la hipermedia por temas sin tener que ir automáticamente por conferencias. El objetivo fundamental de esta entidad es que el estudiantado no se esquematice a ver los objetivos de la asignatura por conferencias sino que vea las mismas a cual tema perteneces, que se esta estudiando en ese momento.

14. **Videos:** Entidad que almacenara todos los videos incluidos en la multimedia aunque no se asocien a los tema o las conferencia.

Una vez que se identifican todas las entidades, se procede a identificar las relaciones entre ellas.

1. Un Temas tiene varias Conferencias por tanto existe una relación uno a muchos (1:N).
2. Un Conferencia tiene varios Ejemplos por tanto existe una relación de uno a muchos (1:N).
 - Tiene varias Bibliografías y una Bibliografía puede estar en varias Conferencias por tanto existe una relación muchos a muchos (N:M).
 - Tiene varias Imágenes y una Imagen puede aparecer en varias Conferencias por tanto existe una relación muchos a muchos (N:M).
 - Tiene varios Videos por tanto existe una relación uno a muchos (1:N).
3. Un Laboratorio evalúa varias Conferencia por tanto existe una relación de uno a muchos (1:N).
 - Tiene varios Ejercicios por tanto existe una relación de una a muchos (1:N).
4. Un Talleres tiene contenido de varias Conferencia por tanto existe una relación de uno a muchos (1:N).
5. Un Ejercicio tiene una Bibliografía por tanto existe una relación de uno a uno (1:1).
6. Un Ejemplo tiene una Bibliografía por tanto existe una relación de uno a uno (1:1).
7. Una Pruebas evalúa el contenido de un Temas por tanto existe una relación de uno a uno (1:1).
 - Tiene varios Ejercicios por tanto existe una relación de uno a muchos (1:N).
8. Un Ejercicio tiene varias Imágenes por tanto existe una relación de uno a muchos (1:N).
9. Un Ejemplo tiene varias Imágenes por tanto existe una relación de uno a muchos (1:N).
10. La Información de Instrumentos contiene varias Imágenes por tanto existe una relación de uno a muchos (1:N).
11. El Programa de Disciplina Topografía bibliografía por tanto existe una relación de uno a muchos (1:N).

Después de definidas todas las relaciones entre las entidades se procede a la elaboración del diagrama E-R, el cual se muestra a continuación:

3.3.1.1 Diagrama Entidad-Relación.

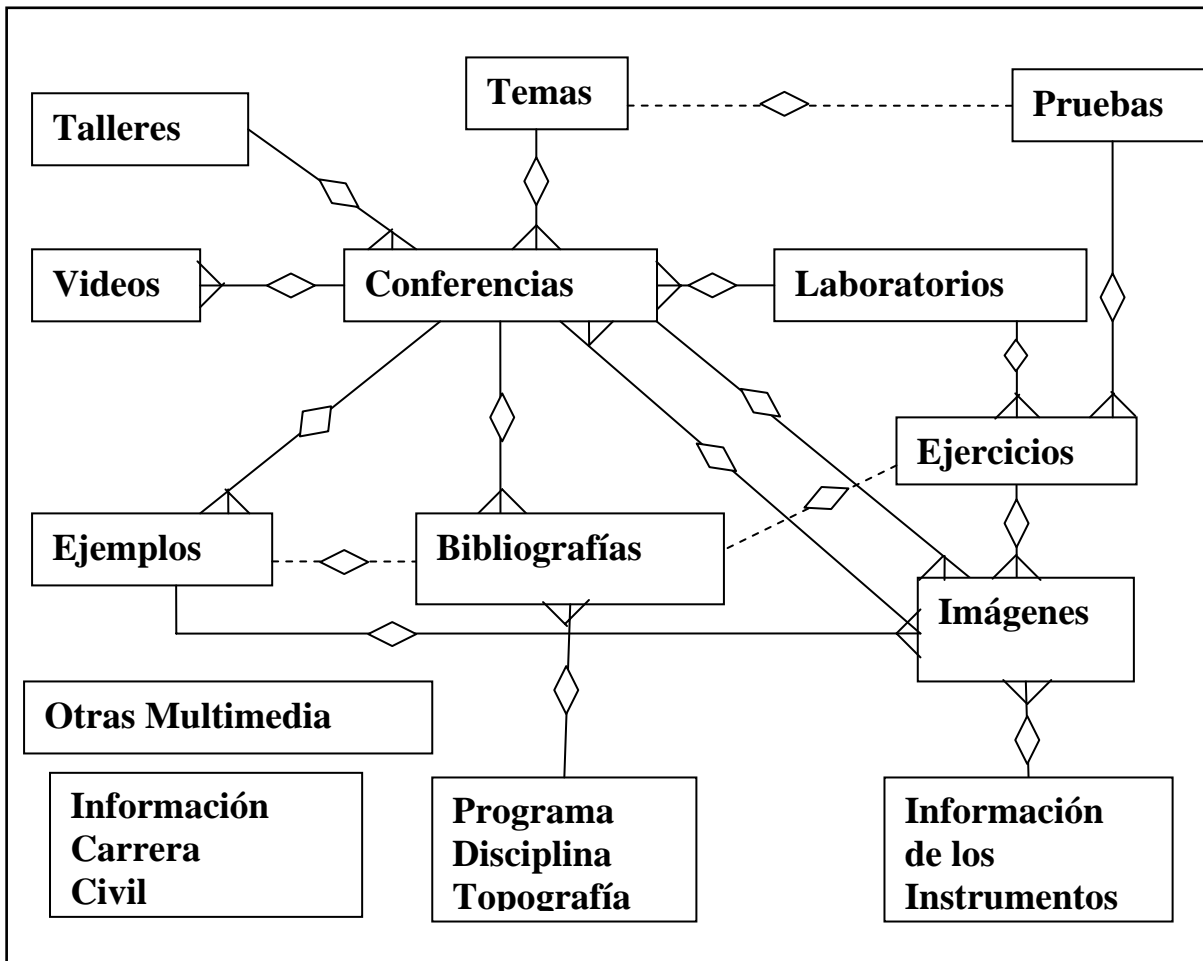


Figura 16. Diagrama E-R del SWE Multimedia Topografía.

3.3.2 Etapa 2: Diseño de los Slice.

En esta segunda etapa cada una de las entidades se dividen en slice y en los casos que sean entidades muy simples constarán de un solo slice, cada slice agrupará uno o más atributos de una entidad de tipos muy diferentes. Cada entidad tendrá su head, o slice principal, que se marca con un asterisco y este es el slice al que, por defecto, se accede a través de los mecanismos de navegación.

En la siguiente tabla se muestra la lista de las entidades divididas en sus slice y con sus correspondientes atributos.

Tabla 3. División en slice de las entidades del SWE Multimedia Topografía.

Entidad	Slice	Atributos
Bibliografía	Bibliografía 1 *(Head).	<ul style="list-style-type: none"> • Numero. • Nombre del documento. • Colectivo de autores. • Direcciones electrónicas.
Conferencia	Conferencia 1 *(Head)	<ul style="list-style-type: none"> • Numero. • Tema. • Bibliografía.
	Conferencia 2	<ul style="list-style-type: none"> • Introducción. • Desarrollo.
Ejemplo	Ejemplo 1 *(Head)	<ul style="list-style-type: none"> • Numero. • Tema • Conferencia • Bibliografía.
	Ejemplo 2	<ul style="list-style-type: none"> • Descripción del problema. • Desarrollo del Ejemplo. • Solución.
Ejercicios	Ejercicios 1 *(Head)	<ul style="list-style-type: none"> • Numero. • Tema • Bibliografía.
	Ejercicios 2	<ul style="list-style-type: none"> • Descripción del problema. • Desarrollo del ejercicio. • Respuesta.
Imágenes	Imágenes 1 *(Head)	<ul style="list-style-type: none"> • Numero. • Nombre • Imagen Pequeña.
	Imágenes 2	<ul style="list-style-type: none"> • Ampliación • Texto explicativo.
Información de la carrera civil	Información de la carrera *(Head)	<ul style="list-style-type: none"> • Resumen.
Información general del instrumental (nivel)	Información del equipo 1. *(Head)	<ul style="list-style-type: none"> • Resumen.

	Información del equipo 2.	<ul style="list-style-type: none"> Niveles y accesorios
Laboratorio	Laboratorio 1 *(Head)	<ul style="list-style-type: none"> Número. Contenido del laboratorio.
	Laboratorio 2	<ul style="list-style-type: none"> Ejercicios
Otras Multimedia	Multimedia *(Head)	<ul style="list-style-type: none"> Número. Nombre de la multimedia. Vinculo directo.
Programa disciplina topografía	Programa disciplina topografía 1 *(Head)	<ul style="list-style-type: none"> Resumen. Objetivos Generales Bibliografía.
	Programa disciplina topografía 2	<ul style="list-style-type: none"> Contenido de la disciplina.
	Programa disciplina topografía 3	<ul style="list-style-type: none"> Sistema de habilidades.
	Programa disciplina topografía 4	<ul style="list-style-type: none"> Evaluación.
Prueba	Prueba 1 *(Head)	<ul style="list-style-type: none"> Número Tema
	Prueba 2	<ul style="list-style-type: none"> Ejercicios
Talleres	Taller *(Head)	<ul style="list-style-type: none"> Número. Contenido.
Temas	Tema 1 *(Head)	<ul style="list-style-type: none"> Número. Nombre del tema.
	Tema 2	<ul style="list-style-type: none"> Conferencia Asociadas
Videos	Video 1 *(Head)	<ul style="list-style-type: none"> Número. Nombre
	Video 2	<ul style="list-style-type: none"> Video.

3.3.3 Etapa 3: Diseñar la Navegación.

En esta etapa se diseñan los caminos que permitirán la navegación por la aplicación. Cada relación de asociación del diagrama E-R+ se analiza, y si se desea hacerla accesible para la navegación, se reemplaza por una estructura de acceso RMDM. Como la metodología RMM se utiliza en dominios donde la información se actualiza con relativa frecuencia, todos los accesos navegacionales se especifican en términos genéricos. Esto quiere decir que los enlaces se especifican mediante propiedades de entidades y relaciones. Esto es posible gracias a los índices condicionales, visitas guiados condicionales, y visitas guiados indexados condicionales. De esta forma si se quiere indicar la relación entre un Tema y las Conferencias que a este pertenecen, no tenemos más que poner una visita guiado indexado condicional en lugar de la relación a la que pertenecen, incluyendo la condición pertenece (Tema, Conferencias), para que solo aparezcan las conferencias que pertenecen a este Tema.

Al final de esta etapa el diagrama E-R+ se ha convertido en un diagrama RMDM, que describe todas las estructuras de acceso que serán implementadas en el sistema.

A continuación para nuestro caso de estudio se obtiene el diagrama RMDM a partir de la estructura de menú que se definió para el SWE Multimedia Topografía.

3.3.3.1 Estructura del Menú (estructura del grupo)

Información General

- Programa Disciplina Topografía.
- Información de la Carrera Civil.
- Información de Instrumentos.

Docencia

- Temas.
- Conferencias.
- Ejemplos.
- Ejercicios.
- Talleres.
- Bibliografías.

Evaluación

- Laboratorios.
- Pruebas.

Galería

- Imágenes.
- Videos.

Otras multimedia

- Multimedia Disponibles.

A continuación representamos la estructura de menú del SWE Multimedia Topografía.

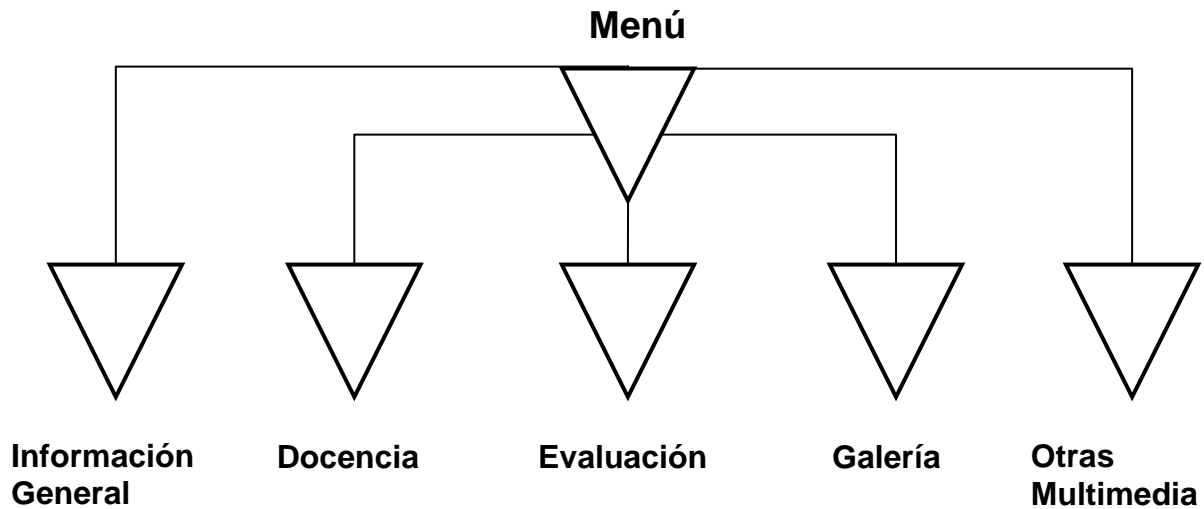


Figura 17. La estructura de menú del SWE Multimedia Topografía.

3.3.3.2 Diagrama RMDM

En el trabajo se muestra el diagrama RMDM dividido con el objetivo de que se observará con mayor claridad.

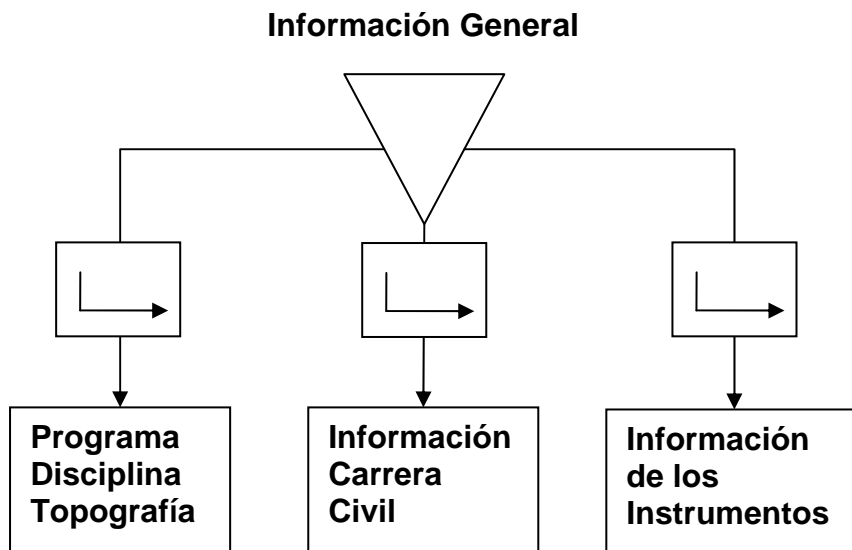


Figura 18. Información General es una parte de diagrama RMDM del SWE Multimedia Topografía.

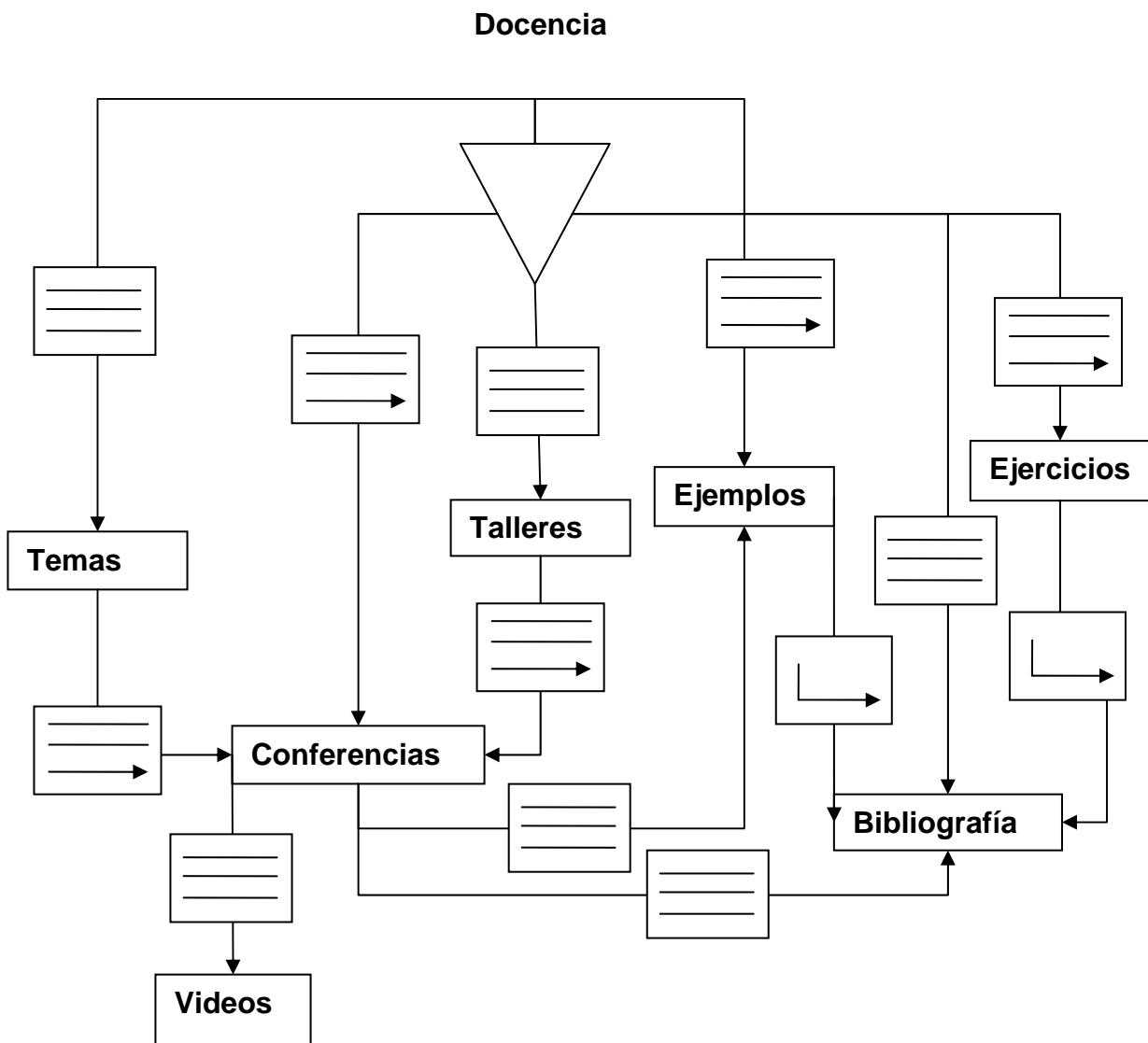


Figura 19. Docencia es una parte del diagrama RMDM del SWE Multimedia Topografía.

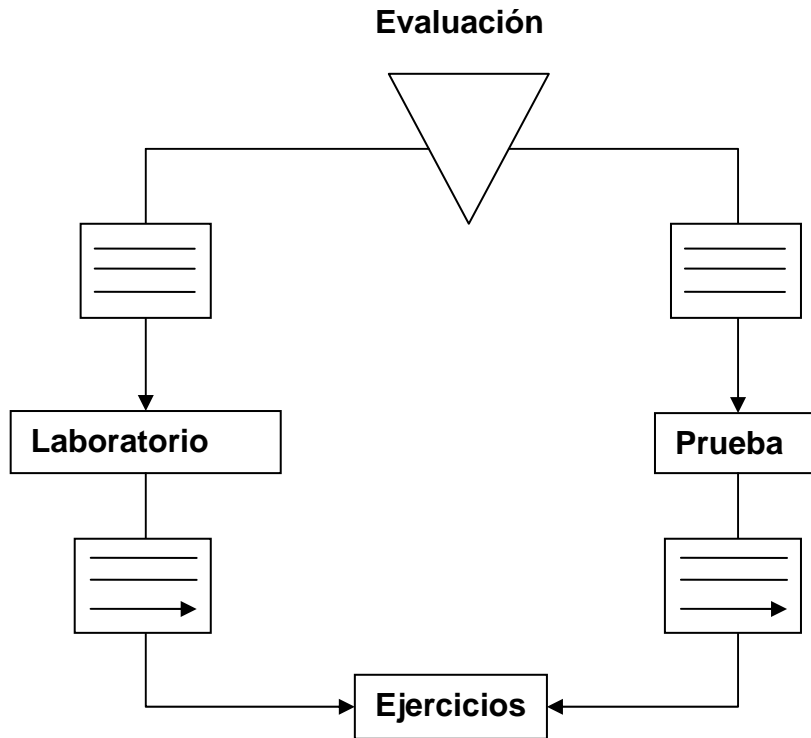


Figura 20. Evaluación es parte del diagrama RMDM del SWE Multimedia Topografía.

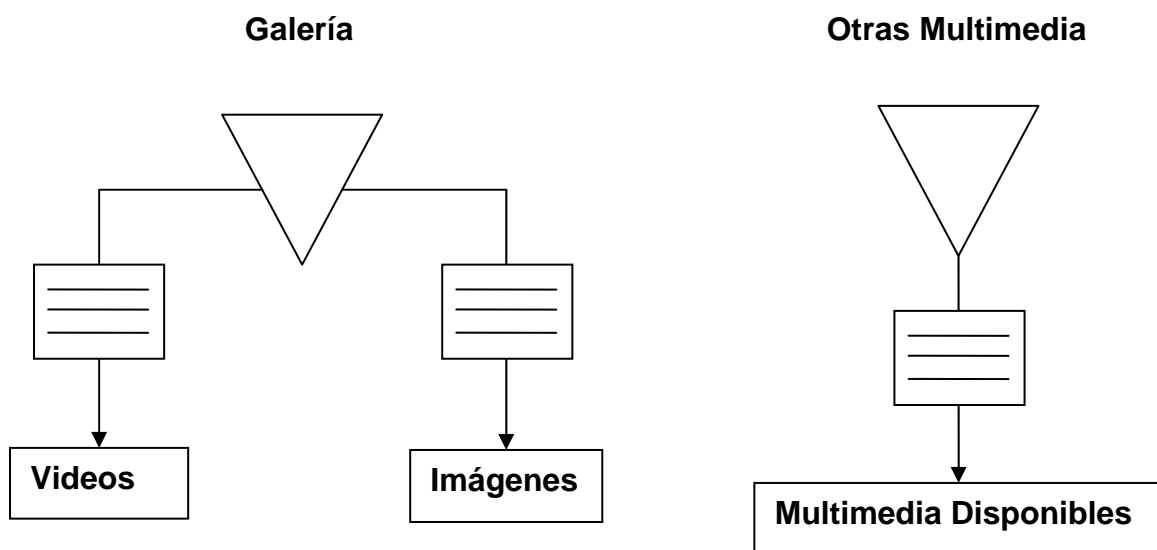


Figura 21. Galería y Otras Multimedia son partes del diagrama RMDM del SWE Multimedia Topografía.

3.3.4 Etapa 4: Definir el Protocolo de Conversión.

En esta etapa se crean reglas de conversión según la plataforma destino en la cual se desarrollará la aplicación, para traducir cada uno de los elementos del diagrama RMDM en objetos del lenguaje o sistema donde se vaya a implementar el diseño. Es decir transformar cada elemento del diagrama RMDM en un objeto de la plataforma destino.

3.3.5 Etapa 5: Diseño de Interfaz de Usuario.

En esta etapa se define la interfaz de usuario de cada una de las presentaciones según los slice de la segunda etapa. Además de determinar cuales serán las anclas y los mecanismos de navegación entre las entidades. Puede ser una buena solución el identificar cada una de las entidades mediante un icono. En este trabajo se utilizan iconos para las entidades:

- Conferencias
- Ejemplos
- Ejercicios
- Talleres
- Laboratorios
- Pruebas

A continuación se explican verbalmente para esta etapa como serian cada una de las interfaces de usuario.

Interfaz Topografía o (Principal)

Tendrá un diseño de presentación con un logo que represente la aplicación y un menú según la estructura definida en la Tercera Etapa, dicho menú permanecerá en todas las demás interfaces para dar una mayor facilidad de navegación al usuario.

Además una explicación de la aplicación, sus objetivos, como navegar y otros aspectos.

Interfaz Programa de la Disciplina Topografía 1* (Head)

Una introducción de la Asignatura, Objetivos Generales e índice con las Bibliografías principales.

Además de tres enlaces que pueden ser botones para acceder a:

- Contenido
- Sistemas de Habilidades
- Evaluación

Interfaz Programa de la Disciplina Topografía 2

Explicación detallada de de los contenidos que abarca la asignatura.

Interfaz Programa de la Disciplina Topografía 3

Explicación de los sistemas de habilidades que son objetivos de la asignatura.

Interfaz Programa de la Disciplina Topografía 4

Explicación del sistema evaluativo que tiene la asignatura.

Interfaz Información de la Carrera Civil

Explicación detallada de la Carrera Civil.

Interfaz Información del Instrumento

Información de los instrumentos que se utilizan para medir y nivelar el terreno, como utilizarlos y imágenes de los mismos accedidas mediante una visita guiada condicional.

Interfaz Temas 1 (Head)

Temas ordenados numéricamente e identificados por su nombre. Desde cada una de los temas existe la posibilidad de ver las conferencias que tiene asociadas, para ello se utiliza como ancla el nombre del Tema y se accederá mediante una visita guiada indexada condicional.

Interfaz Temas 2

Conferencias que pertenecen al Tema y con la posibilidad de acceder a los contenidos de las mismas utilizando como ancla su icono y mediante una visita guiada condicional.

Interfaz Conferencias 1 (Head)

Conferencias ordenada numéricamente y el Tema al cual pertenece, además un botón como ancla bibliografía que mediante un índice condicional mostrará las bibliografías correspondientes a dicha

conferencia. Desde cada una de las conferencias existe la posibilidad de acceder a su contenido utilizando como ancla su icono.

Interfaz Conferencia 2

Introducción y desarrollo de la conferencia. Se muestran los ejemplos que pertenecen a dicha conferencia mediante un índice condicional con los números de los ejemplos. Para ver los videos asociados a la conferencia se utiliza una índice condicional con los nombre de los videos.

Interfaz Ejemplo 1 (Head)

Ejemplos ordenados numéricamente con el tema y conferencia al cual pertenece. Vinculo a la bibliografía mediante una visita guiada condicional. Además existe la posibilidad de acceder al contenido del ejemplo utilizando como ancla su icono.

Interfaz Ejemplo 2

Se presenta la descripción de problema, el desarrollo y solución del mismo.

Interfaz Ejercicios 1 (Head)

Ejercicios ordenados numéricamente y el tema al cual pertenece. Vinculo a la bibliografía mediante una visita guiada condicional. Además existe la posibilidad de acceder al ejercicio tomando como ancla su icono.

Interfaz Ejercicios 2

Se presenta la descripción de problema, la forma en que se responderá y una solución para que el estudiante pueda autoevaluarse.

Interfaz Talleres 1 (Head)

Talleres ordenados numéricamente y la posibilidad de ver las conferencias impartidas para su realización utilizando como ancla un botón Conferencias y mediante una visita guiada indexada. Además existencia de acceso a su contenido utilizando como ancla su icono.

Interfaz Talleres 2

Contenido del taller.

Interfaz Bibliografía 1 *(Head)

Bibliografías ordenada numéricamente y con el nombre del documento, los autores y la dirección electrónica. Además de la posibilidad de acceder a la bibliografía tomando como ancla su dirección electrónica.

Interfaz Laboratorio1* (Head)

Laboratorios ordenados numéricamente y una breve reseña de los contenidos que evalúa. Además se accede mediante una visita guiada indexada condicional a los ejercicios para desarrollarlos utilizando su icono como ancla.

Interfaz laboratorio 2

Ejercicios cada uno con la descripción del problema, la forma en que se responderá y un botón para calificar al estudiantes cuantitativa y cualitativamente en los ejercicios resueltos.

Interfaz Prueba 1* (Head)

Pruebas ordenada por Tema al cual pertenece. Además de permitir el acceso tomando como ancla su icono a cada uno de las pruebas para responder los ejercicios mediante una visita guiada indexada condicional.

Interfaz Prueba 2

Ejercicios cada uno con la descripción del problema, la forma en que se responderá, además un botón para calificar al estudiante cuantitativa y cualitativamente en los ejercicios resueltos.

Interfaz Imágenes 1 *(Head)

Imágenes ordenadas numéricamente y con su nombre e imagen correspondiente, la cual se puede ver ampliada y con una explicación de su contenido utilizando como ancla la imagen.

Interfaz Imágenes 2

Imagen ampliada y texto con la explicación del contenido de la imagen.

Interfaz Videos 1*(Head)

Nombre de los videos ordenados y la posibilidad de verlo utilizando como ancla el nombre.

Interfaz Videos 2

Reproducción de videos con las posibilidades de realizar las funciones básicas de cualquier reproductor.

Interfaz Otras Multimedia 1* (Head)

Lista de las multimedia disponibles con su nombre y posibilidad de acceder utilizando como ancla el nombre.

3.3.6 Etapa 6: Diseño del Comportamiento en Ejecución.

La aplicación debe poseer un historial de ejecución de enlaces, herramientas como backtracking y similares y mecanismos de navegación como índices condicionales, visitas guiadas condicionales y visitas guiadas indexadas condicionales definiendo con precisión para cada enlaces el mecanismos correcto.

En la aplicación se tienen contenidos que se generan antes de la ejecución como por ejemplo la Información de la Carrera de Civil y otros que se generan dinámicamente en el tiempo de ejecución como la lista de Conferencias que pertenecen a un determinado Tema.

3.3.7 Etapa 7: Construcción y Prueba

Finalmente en la etapa de construcción y prueba se llevan a cabo las actividades tradicionales de ingeniería del software, prestando especial atención a la prueba de los caminos navegacionales.

3.4 Resultados de aplicar la metodología RMM.

Al aplicar la metodología RMM en el caso de estudio Multimedia Topografía se logró una navegación mas estructurada, regular e intuitiva.

Con el diseño del modelo entidad-relación se representa de forma coherente y bien estructurado todo el dominio de información de la aplicación, representado por las entidades y ampliado con relaciones asociativas las cuales permiten representar caminos navegacionales entre entidades puestos en evidencia en la fase de análisis.

Con la división en slice de las entidades se puede definir de forma muy clara las informaciones que serán mostradas al usuario en cada una de las pantallas.

Con la representación del modelo RMDM se define como se accederá a cada una de las informaciones con la utilización de las primitivas de acceso por lo que se obtiene una navegación bien estructura e intuitiva para los usuarios.

Al definir reglas de conversión para llevar cada uno de los elementos del modelo RMDM al lenguaje o plataforma destino donde se desarrollara la aplicación se logra estandarizar los objetos del dominio de la aplicación para facilitar su implementación dotando a los programadores de un lenguaje propio.

Cuando se diseñan las interfaz de usuario, se diseñan para cada una de las slice que serán presentados al usuario y que se han definida en la etapa 2 así se tiene bien definido como se presentaran cada uno de los contenidos al usuario final.

En cuanto al comportamiento en tiempo de ejecución es muy importante para los programadores saber que nodos y enlaces se generan antes de la ejecución o en tiempo de ejecución así se les facilita en gran medida la implementación.

Con respecto a la construcción de la aplicación, después de haber realizado un diseño tan detallado de la hipermedia que se desea obtener con una representación sencilla y fácil de comprender llevar todo a implementación es mucho más fácil y seguro.

De forma general se disminuye el tiempo de desarrollo y se transita por un ciclo de vida del software donde se especifica cada uno de los pasos, se realiza un diseño sencillo y fácil de llevar a implementación y se logra una navegación muy bien estructurada.

Las aportaciones con respecto a HDM residen en la descripción detallada de las etapas diferentes del proceso de concepción por una parte, y por otra parte, en la manera en que los diferentes elementos del modelo deben ser utilizados.

3.5 Conclusiones

En este capítulo se le aplicó al caso de estudio las etapas con que cuenta la metodología RMM, aunque se centra especial atención a las etapas de diseño que son las principales etapas en la metodología RMM. Se describen cada uno de los pasos y se muestran los principales diagramas obtenidos para el SWE Multimedia Topografía. Además se plasman los resultados obtenidos de aplicar la metodología RMM al caso de estudio.

Conclusiones

RMM a diferencia de su antecesora HDM, se puede considerar una metodología que asume un ciclo de vida completo compuesto de siete fases o etapas en las que el diseñador va modelando la estructura de la aplicación y las posibilidades de navegación de la misma.

La Metodología RMM se define como un proceso de análisis, diseño y desarrollo de aplicaciones hipermedia cuya estructura es estable y cuyo contenido sufre modificaciones frecuentes. Con la realización de este trabajo se ha podido dar cumplimiento al objetivo planteado logrando obtener a partir de las etapas de la metodología el análisis y diseño del SWE Multimedia Topografía.

Es muy importante tener presente que la utilización de esta depende del tipo de aplicación que se quiere obtener, es adecuada en aplicaciones que presentan un alto grado de estructuración en particular en aquellas que tienen por debajo una base de datos relacional, como la nuestra y que además requieren una actualización constante de la información.

El modelamiento conceptual de RMM se destaca por ser fácil de asimilar por parte de los diseñadores y sencillo de traspasar a implementación, cuando las aplicaciones son de un bajo nivel de complejidad.

Si se tiene una base de datos relacional y se quiere proporcionar acceso hipermedia de manera sencilla y elegante, nada mejor que utilizar la metodología RMM.

RMCASE es la herramienta de software que soporta algunas fases de RMM que aunque presenta algunas deficiencias brinda ciertas ventajas para la realización de los diagramas que se obtiene en las diferentes etapas de diseño.

Recomendaciones

Una vez concluido el desarrollo de este documento se recomienda:

- La utilización de la Metodología RMM en los procesos de desarrollo de software educativo que presentan una base de datos relacional, y que además requieren una actualización constante de la información.
- La utilización de la herramienta RMCASE que aunque no se utilizó en este trabajo de diploma y en realidad presenta algunas deficiencias, será de gran ayuda para la realización de los diagramas que la metodología requiere.
- La realización de posteriores estudio y análisis mas profundo en las fases de Definir el Protocolo de Conversión y de Diseño del Comportamiento en Ejecución en la metodología estudiada, las cuales no fueron profundamente abordadas en este trabajo.

Dar seguimiento a las investigaciones sobre las metodologías de desarrollo de software que se puedan utilizar en los procesos de desarrollo de SWE en nuestro país ya que cada software tiene características diferentes y otras metodologías podrían lograr mayor eficiencia en el desarrollo de determinadas aplicaciones y la obtención de productos con mayor calidad.

Bibliografía

1. Díaz, G. (2003). "Propuesta de una metodología de desarrollo y evaluación de software educativo bajo un enfoque de calidad sistémica." Caracas, Venezuela.
<http://www.academia-interactiva.com/ise.pdf>
2. Jacobson, I. (1998). "Applying UML in The Unified Process. Rational Software."
3. Juscado, J. (1996). "Procesos de construcción de software y ciclos de vida."
4. Kemmis, S. (1976). "The Educational Potential of Computer Assisted learning: Qualitative Evidence about Student Learning."
5. Maddison, R.N. (1983). "Information System methodologies."
6. Marqués P. (1995). "Metodología para la elaboración de software educativo."
7. Piattini, P. (1996). "Análisis y Diseño Detallado de Aplicaciones Informáticas de Gestión."
8. Pressman, R.S. (2006). "Software engineering Resources."
9. Sánchez, J. (1999). "Construyendo y Aprendiendo con el Computador."
10. Taylor, R. (1980). "The computer in the school: tutor, tool, tutel."
<http://dewey.uab.es/pmarques/funcion.htm>
11. Isakowitz, T. (1995). "RMM: a methodology of structured hypermedia design." New York University. <http://jmis.bentley.edu/rmm/papers/rmd.pdf>
12. Lagos, P.S. (2002). "Ingeniería de software educativo, teorías y metodologías que lo sustentan". <http://www.inf.udec.cl/%7Erevista/ediciones/edicion6/isetm.PDF>
13. Navarro, A. (2002). "Conceptualización, prototipado y proceso de aplicaciones hipermedia." Universidad Complutense de Madrid. <http://www.ucm.es/BUCM/tesis/mat/ucm-t25910.pdf>
14. Elmasri, R. (1997). "Sistemas de bases de datos: conceptos fundamentales."
15. Rodriguez, E. (2002). "La informática educativa."
16. Chen, P. (1976). "The entity-relationship approach: toward a unified view of data."
17. Gros, B. (1997). "Diseños y programas educativos." Barcelona.
<http://www.xtec.es/~pmarques/edusoft.htm#tipologias>
18. Pressman, R. (2002) "Ingeniería de Software: Un enfoque Práctico". McGraw Hill.
19. Marques, P. (1996). "El software educativo." Universidad Autónoma de Barcelona.
http://www.lmi.ub.es/te/any96/marques_software/#capitol13
20. Galvis, A., (2000) Ingeniería de software educativo. 2da. reimpresión. Universidad de Los Andes. Ediciones UNIANDES. Colombia.

Anexos

Anexo 1. Ejemplo de un modelo entidad-relación (E-R) con RMCASE.....	79
Anexo 2. Ejemplo de diseño de slice con RMCASE.	79
Anexo 3. Ejemplo de diseño de navegación con RMCASE.....	80
Anexo 4. Ejemplo de interfaz de usuario con RMCASE.	80
Anexo 5. Ejemplo de prototipo funcional con RMCASE.....	81

Anexo 1. Ejemplo de un modelo entidad-relación (E-R) con RMCCase.

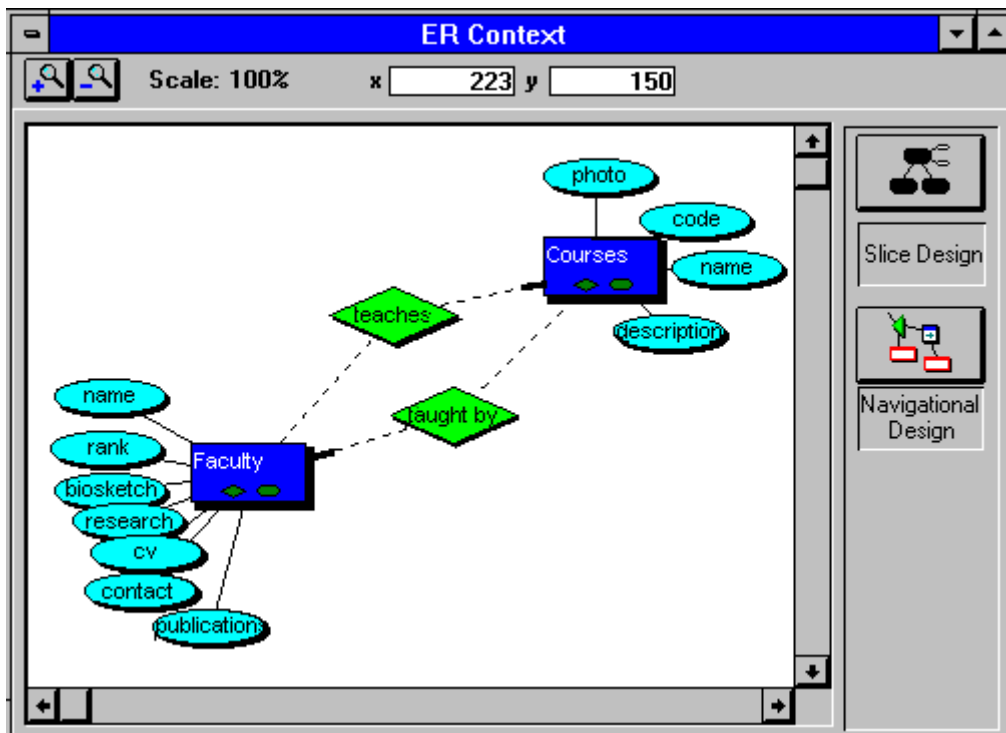


Figura 22. Interfaz de la herramienta RMCCase para el diseño del diagrama E-R.

Anexo 2. Ejemplo de diseño de slice con RMCCase.

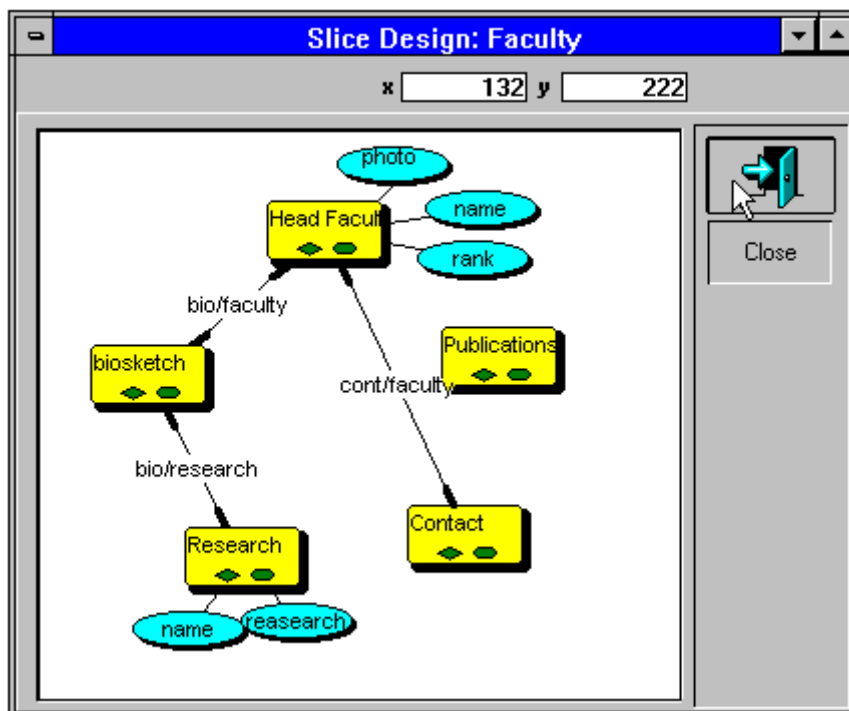


Figura 23. Interfaz de la herramienta RMCCase para el diseño de slice.

Anexo 3. Ejemplo de diseño de navegación con RMCCase.

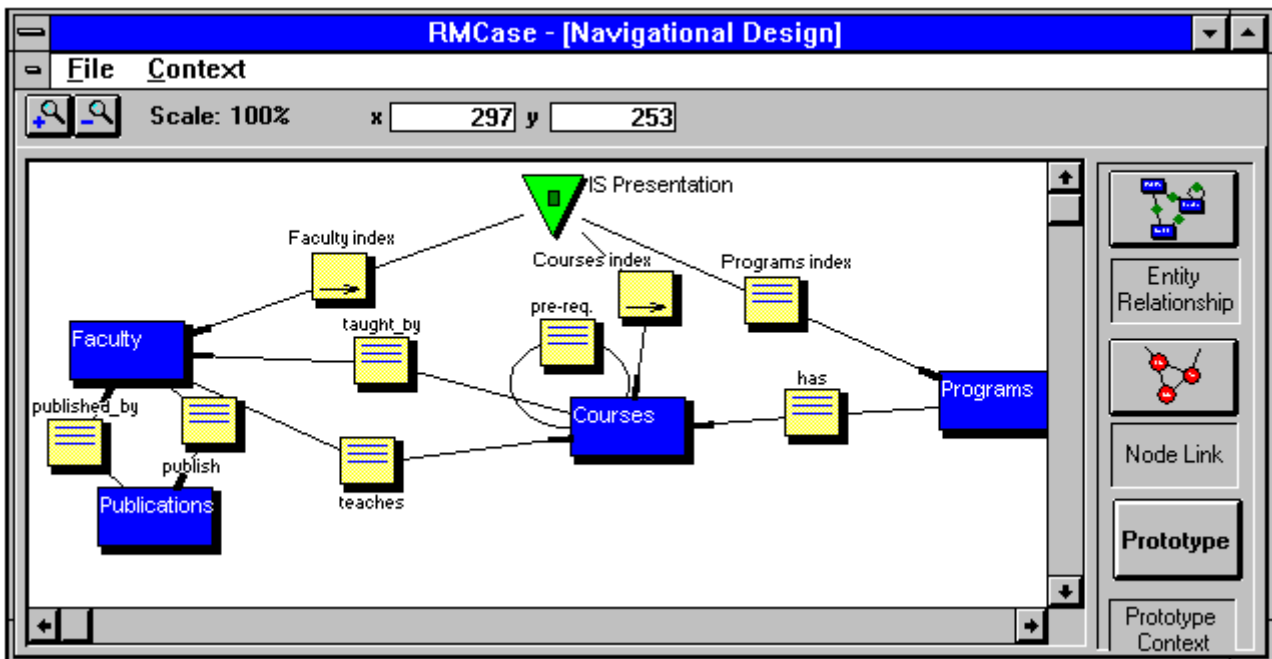


Figura 24. Interfaz de la herramienta RMCCase para diseñar el diagrama RMDM

Anexo 4. Ejemplo de interfaz de usuario con RMCCase.

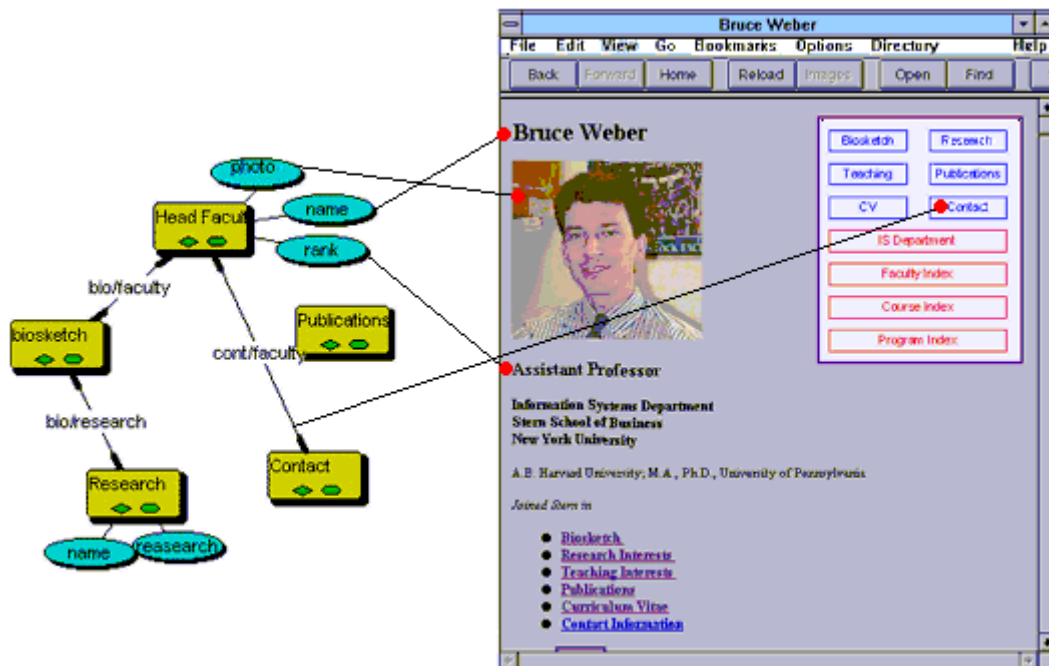


Figura 25. Diseño de una interfaz de usuario con RMCCase.

Anexo 5. Ejemplo de prototipo funcional con RMCas.

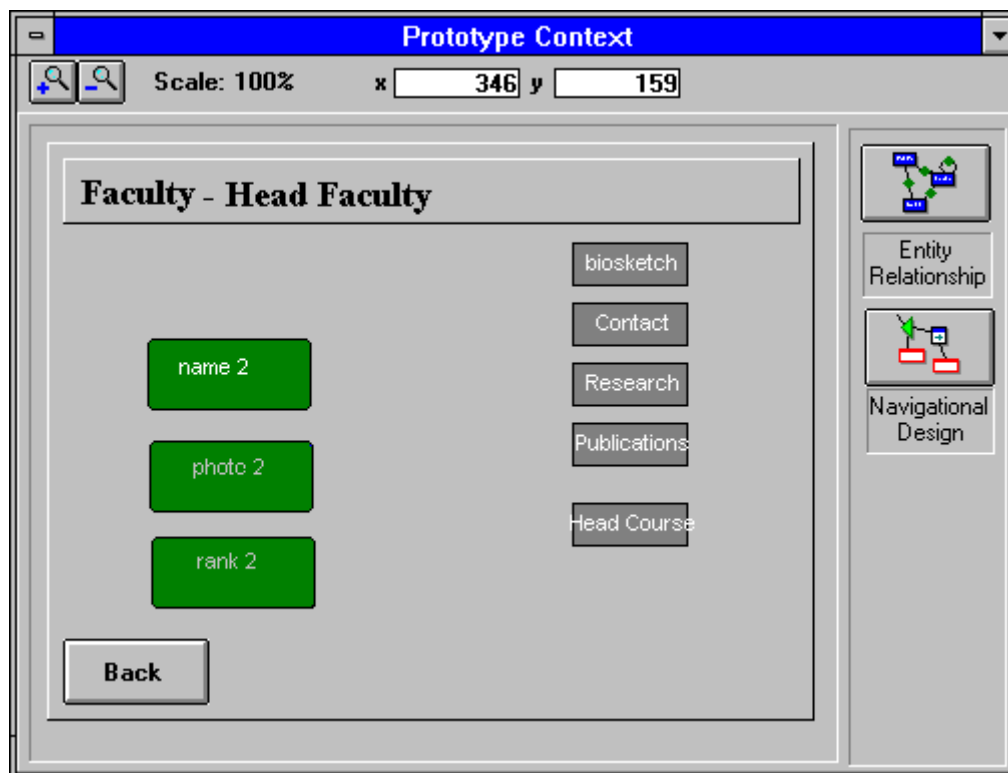


Figura 26. Prototipo de aplicación con RMCas.

Glosario

RMM: Relationship Management Methodology [Metodología de Administración de Relaciones] es una metodología de análisis, diseño y desarrollo de aplicaciones hipermedia.

HDM: Hypermedia Design Model [Método de Diseño Hipermedia] es un modelo para realizar el diseño de una aplicación de hipertexto (o hipermedia).

OOHDM: Object Oriented Hypermedia Design Methodology [Metodología de Diseño Hipermedia Orientada a Objetos] es una metodología de diseño basada en un modelo de proceso incremental o de prototipos. Cada paso se centra en una característica de diseño, construyéndose un modelo orientado a objetos.

Slice: Conjuntos de atributos de una entidad que se agrupan de forma lógica para ser presentados en una pantalla.

Entidad: Elemento conceptual del dominio de aplicación, caracterizado por un conjunto de atributos, se representa mediante un recuadro y su nombre.

Atributo: Representa una unidad de información.

Enlace: Relación de asociación que existe entre entidades o slice.

E-R: Diagrama Entidad-Relación, modelos de datos que muestra el dominio de información de la aplicación mediante sus entidades y las relaciones existente entre ellas.

RMDM: Relationship Management Data Model [Modelo de Datos Relacional], es el modelo de datos relacional que se obtiene en la metodología RMM donde se muestra el dominio de información de la aplicación y los caminos de navegación existentes entre ellas.

Software: Es la suma total de los programas de cómputo, procedimientos, reglas, documentación y datos asociados que forman parte de las operaciones de un sistema de cómputo.

Ingeniería de Software: Es una tecnología multicapa en la que, se pueden identificar: los métodos (indican cómo construir técnicamente el software), el proceso (es el fundamento de la Ingeniería de Software, es la unión que mantiene juntas las capas de la tecnología) y las herramientas (soporte automático o semiautomático para el proceso y los métodos)

Metodologías: Se refiere a los métodos de investigación en una ciencia. Se entiende como la parte del proceso de investigación que permite sistematizar los métodos y las técnicas necesarios para llevarla a cabo. Define Quién debe hacer, Qué, Cuándo y Cómo debe hacerlo.

Metodologías de Desarrollo: Se define como un conjunto de filosofías, etapas, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de sistemas de información.

Software Educativo (SWE.): Se define como un programa automatizado diseñado con una intencionalidad educativa para ser utilizado en el proceso de aprendizaje, utiliza procedimientos para que el estudiante aprenda, se fomenta el análisis de problemas, facilita el trabajo en grupo, provee soporte en actividades docentes y en el sentido más amplio, mejora las habilidades del pensamiento y la resolución de problemas.

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

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.

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.

Hipermedia: Es el resultado de la combinación del hipertexto y la multimedia.

Herramienta CASE: Conjunto de programas y ayudas que dan asistencia a los analistas, ingenieros de software y desarrolladores, durante todos los pasos del ciclo de vida de desarrollo de un software.

TIC: Tecnologías de la Información y las Comunicaciones.

IEEE: Corresponde a las siglas del Instituto de Ingenieros Eléctricos y Electrónicos. Es una asociación estadounidense dedicada a la estandarización internacional sin fines de lucro formada por profesionales de las nuevas tecnologías.