

**MINISTERIO DE EDUCACIÓN SUPERIOR
INSTITUTO SUPERIOR MINERO METALÚRGICO DE MOA
DR. ANTONIO NÚÑEZ JIMÉNEZ
Facultad de Metalurgia - Electromecánica
Departamento de Informática**



Tesis de grado en opción al título de Ingeniera Informática

GUÍA METODOLÓGICA PARA PROYECTOS DE SOFTWARE

**Autor: Roxana Díaz Jardínez
Tutores: Yadira Romero Rodriguez
Yessenia Rosario Ferrer**

**Moa, Cuba
Julio, 2008**

DECLARACIÓN DE AUTORÍA

Declaro que soy la única autora de este trabajo y autorizo al Instituto Superior Minero Metalúrgico de Moa "DR. ANTONIO NÚÑEZ JIMÉNEZ" para que hagan el uso que estimen pertinente con este trabajo.

Para que así conste firmo la presente a los _____ días del mes de _____ del 2008.

Roxana Díaz Jardínez _____

Nombre completo del primer autor

Yadira Romero Rodríguez

Nombre completo del primer tutor

Agradecimientos

A todas las personas que han estado conmigo en los momentos difíciles y han sabido comprenderme. Gracias.....

Dedicatoria

De Roxana

A mi hermana, Rosmery por estar ahí, comprenderme en mis momentos buenos y en los malos, espero que sigas el ejemplo y que logres coger la carrera que deseas.

A mi abuela por siempre preocuparse por mí, por soportarme mis malacrianzas, por siempre estar ahí para apoyarme, y por saber comprenderme. Te quiero mucho.

A mis padres: (Mabel y Roberto), por su confianza y dedicación, por estos 23 años de sacrificio y espera, gracias por ser los mejores padres del mundo...

A mis compañeros, gracias por soportarme, comprenderme y escucharme, especialmente a Xenia y Tony por sus constantes consejos.

A mis amigos Dailén y Jose por su colaboración y su ayuda.

A mis tutoras: Gracias por ayudarme en la tesis....

A familia, especialmente (mi abuela Yaya y mis tías) por su preocupación respecto a mi tesis....

RESUMEN

El creciente desarrollo en las Tecnologías de la Información y la Comunicación ha sido totalmente globalizado. La riqueza, variedad y potencia de los soportes y vías para transmitir y almacenar información han sido sustanciales para el desarrollo de la economía de nuestro país, es por esta razón que hemos tenido que ser capaces de crear nuestros propios sistemas de software y de esta forma cubrir nuestras necesidades. El objetivo principal del presente trabajo de diploma es posibilitar a cada desarrollador una vía de estudio para desarrollar un producto con calidad a estudiantes y profesores del Instituto Superior Minero Metalúrgico de Moa "Antonio Núñez Jiménez". Muchos desarrolladores de software crean sus sistemas y luego comienzan los problemas, esto se debe a la escasa documentación que existe de las metodologías de desarrollo de software y en la mayoría de los proyectos a desarrollar no se aplica la metodología apropiada. La Guía Metodológica para Proyectos de Software cubrirá la necesidad de muchos ingenieros a la hora de modelar cada tipo de aplicación. En la actualidad no existe ningún documento que de manera organizada recoja las metodologías para los diferentes tipos de proyectos y aplicaciones, toda la información aparece dispersa por lo que esta guía resolverá estos problemas. Nuestro trabajo propone un estudio de las diferentes metodologías para aplicaciones de escritorio, multimedia, y desarrollo Hipermedia y Web como proceso de Ingeniería. Y de estas metodologías definidas desarrollar una por cada tipo de aplicación con un ejemplo práctico de la misma.

Abstract

The growing development in the Information and Communication Technology has been totally globalized. The richness, variety and ability of the supports and means of transmitting and storing information have been substantial to the development of the economy of our country. It is for this reason that we have had to be able to create our own software systems and in this way cover our necessities. The main objective of the present thesis is to facilitate each developer with a study guide in the development of a quality product for the students and lecturers at the Higher Institute of Mining and Metallurgy of Moa "Antonio Núñez Jiménez." Many software developers create their systems and later on these have problems; this is due to the scarce documentation that exists on methodologies in software development and in most projects proposals an appropriate methodology is not applied. The Methodological Guide for Software Projects will cover the necessity of many engineers when modeling each application type. At the present time there is no document that sums up all the methodologies for different types of projects and applications in an organized way, as a consequence all the information appears dispersed. This guide intends solve such problems. Our work proposes a study of different methodologies for desktop applications, multimedia, Hypermedia and Web development as Engineering processes. For each of the aforementioned applications, a methodology for each type will then be developed, with a corresponding practical example.

	Pag
INTRODUCCIÓN.....	1
CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA.....	5
1.1. Metodologías de desarrollo de software.....	5
1.2. Desarrollo Hipermedia y Web.....	15
1.3. Aplicaciones multimedia.....	16
CAPÍTULO 2. METODOLOGÍAS PARA EL DESARROLLO DE APLICACIONES	18
MULTIMEDIA, HIPERMEDIA Y WEB.....	
2.1. Metodologías de desarrollo de Hipermedia y Web.....	19
2.2. Metodología propuesta para el desarrollo de aplicaciones Hipermedia y Web: Navigator Development technique ¹ (NDT).....	40
2.3. Metodología para el desarrollo de aplicaciones Multimedia.....	44
2.4. Metodología Multimet.....	51
CAPÍTULO 3. EJEMPLO PRÁCTICO DE LAS METODOLOGÍAS NDT Y	67
MULTIMET.....	
3.1. Ejemplo práctico de la metodología NDT.....	67
3.2 Ejemplo de MULTIMET.....	116
CONCLUSIONES.....	122
RECOMENDACIONES.....	123
BIBLIOGRAFÍA.....	124
GLOSARIO DE TÉRMINOS.....	125
ANEXOS.....	128

ÍNDICE DE TABLAS

	pag
Tabla 1.1. Clasificación de las metodologías de desarrollo de software.....	6
Tabla 2.1. Fases, actividades y tareas de NDT.....	43
Tabla 2.2. Modelo para la definición de los medios y sus objetivos.....	56
Tabla 2.3. Modelo para la confección del guión.....	61
Tabla 3.1- Patrón del objetivo OBJ-01.....	69
Tabla 3.2- Patrón del objetivo OBJ-02.....	69
Tabla 3.3- Patrón del requisito RA-01.....	70
Tabla 3.4- Patrón del requisito RA-02.....	71
Tabla 3.5 Patrón del requisito RA-03.....	72
Tabla 3.6- Patrón de la naturaleza NA-01.....	73
Tabla 3.7- Patrón de la naturaleza NA-02.....	73
Tabla 3.8- Patrón de la naturaleza NA-03.....	74
Tabla 3.9- Patrón del actor AC-01.....	74
Tabla 3.10- Patrón del actor AC-02.....	74
Tabla 3.11- Patrón del actor AC-03.....	75
Tabla 3.12- Patrón del actor AC-04.....	75
Tabla 3.13- Patrón del actor AC-05.....	76
Tabla 3.14- Matriz de incompatibilidad de actores.....	76
Tabla 3.15- Patrón del requisito RF-01.....	77
Tabla 3.16- Patrón del requisito RF-02.....	78
Tabla 3.17- Patrón del requisito RF-03.....	78
Tabla 3.18- Patrón de la frase FR-01.....	80
Tabla 3.19- Patrón de la frase FR-02.....	81
Tabla 3.20- Patrón para el prototipo PV-01.....	82
Tabla 3.21- Patrón para el prototipo PV-02.....	83
Tabla 3.22- Patrón para el prototipo PV-03.....	84
Tabla 3.23- Patrón para el prototipo PV-01 fusionando PV-01 y PV-3.....	85
Tabla 3.24- Patrón para el prototipo PV-04.....	86
Tabla 3.25- Patrón para el prototipo PV-05.....	87
Tabla 3.26 Matriz de rastreabilidad.....	88

Tabla 3.27- Patrón para la definición de la clase CL-01.....	90
Tabla 3.28- Patrón para la definición de la clase CL-02.....	91
Tabla 4.29- Patrón para la definición de la clase CL-03.....	91
Tabla 3.30- Patrón para la definición de la asociación AS-01.....	92
Tabla 3.31- Patrón para la definición de la asociación AS-02.....	93
Tabla 3.32- Patrón para la definición de la asociación AS-03.....	93
Tabla 3.33- Patrón para la definición de la asociación AS-04.....	94
Tabla 3.34- Patrón para la definición de la asociación AS-05.....	94
Tabla 3.35- Patrón para la definición de la asociación AS-06.....	94
Tabla 3.36- Patrón para la definición de la asociación AS-07.....	95
Tabla 3.37- Patrón para la definición de la asociación AS-08.....	95
Tabla 3.38- Patrón para la definición de la asociación AS-09.....	96
Tabla 3.39- Patrón para la definición de la asociación AS-10.....	96
Tabla 3.40- Patrón para la definición de la asociación AS-11.....	96
Tabla 3.41- Patrón para la definición de la asociación AS-12.....	97
Tabla 3.42- Patrón para la definición de la asociación AS-13.....	97
Tabla 3.43- Patrón para la definición de la asociación AS-14.....	98
Tabla 3.44- Patrón para la definición de la clase CLn-01.....	98
Tabla 3.45- Patrón para la definición de la clase CLn-02	99
Tabla 3.46- Patrón para la definición de la clase CLn-03.....	99
Tabla 3.47- Patrón para la definición del paquete PQ-01.....	100
Tabla 3.48- Patrón para la definición del paquete PQ-02.....	101
Tabla 3.49- Patrón para la definición de la clase CL-04.....	101
Tabla 3.50- Patrón para la definición de la clase CL-01.....	102
Tabla 3.51- Patrón para la definición de la clase CL-02.....	103
Tabla 3.52- Patrón para la definición de la asociación AS-02	104
Tabla 3.53- Patrón para la definición de la asociación AS-03.....	105
Tabla 3.54- Patrón para la definición de la asociación AS-04.....	105
Tabla 3.55- Patrón para la definición de la asociación AS-05.....	106
Tabla 3.56- Patrón para la definición de la asociación AS-07.....	106
Tabla 3.57- Matriz de descripción de actores en estudio.....	107
Tabla 3.58- Patrón para la definición del nodo NO-01.....	108
Tabla 3.59- Patrón para la definición del nodo NO-02.....	109

Tabla 3.60- Patrón para describir el enlace EN-01.....	109
Tabla 3.61- Patrón para describir el índice IN-01.....	110
Tabla 3.62- Patrón para describir la query QU-01.....	110
Tabla 3.63- Patrón para describir la query QU-02.....	111
Tabla 3.64- Patrón para describir el enlace EN-02.....	111
Tabla 3.65- Patrón para describir el enlace EN-03.....	112
Tabla 3.66- Patrón para describir el enlace EN-01.....	112
Tabla 3.67- Patrón para describir el enlace EN-04.....	113
Tabla 3.68- Patrón para describir el enlace EN-05.....	113
Tabla 3.69- Patrón para describir el menú principal.....	114
Tabla 3.70- Patrón para describir el enlace EN-06.....	115
Tabla 3.71- Patrón para describir el enlace EN-05.....	115

Índice de figuras

	pag
Figura 2.1. Relaciones entre las diferentes metodologías.....	31
Figura 2.2. Relación entre los elementos de HDM.....	33
Figura 3.1. Generalización de actores.....	86
Figura 3.2. Diagrama de caso de uso del sistema.....	88
Figura 3.3. Modelo conceptual básico.....	112
Figura 3.4. Modelo conceptual final.....	114
Figura 3.5. Grafo navegacional para los turistas.....	124
Figura 3.6. Modelo de navegación final para los turistas.....	127

INTRODUCCIÓN

Desarrollar productos de software no solo es escribir instrucciones de programación juntas y ejecutarlas en un ordenador, es preciso obtener un producto en tiempo, con pocos errores y con buena calidad.

A nivel mundial en la Industria de Software hay tendencia al crecimiento del volumen y complejidad de los productos, los proyectos están excesivamente tardes, se exige mayor calidad y productividad en menos tiempo y hay insuficiente personal calificado.

Uno de los principales problemas en el desarrollo de sistemas es el poco conocimiento que se tiene de cómo desarrollarlo de forma eficiente, es por eso que para los ingenieros de software es imprescindible tener bastos conocimientos de Ingeniería de Software, pues esta disciplina le permite obtener software de modo rentable y con calidad, asegurando que los resultados se adecuen a las necesidades reales del usuario.

Durante años, muchas aplicaciones han fallado porque existieron incongruencias entre lo que el usuario quería, lo que realmente necesitaba, lo que interpretaba cada miembro del equipo de proyecto y lo que realmente se obtiene.

Por todos estos problemas ha sido necesario realizar un estudio de los principios y metodologías para el desarrollo y mantenimiento de sistemas de software. Un paso útil para desarrollar estos sistemas de software es el de crear modelos que organicen y comuniquen los detalles importantes del problema de la vida real con que se relacionan y del sistema a construir.

Muchas veces en el desarrollo de un software no se toma en cuenta el utilizar una metodología adecuada, sobre todo cuando se trata de proyectos pequeños. Cuando los proyectos que se van a desarrollar son de mayor envergadura, ahí si toma sentido el basarnos en una metodología de desarrollo, y empezamos a buscar cual sería la más apropiada para nuestro caso. Lo cierto es que muchas veces no encontramos la más adecuada y terminamos por hacer o diseñar nuestra propia metodología, algo que por supuesto no está mal, siempre y cuando cumpla con el objetivo.

La Informatización de la Sociedad Cubana a partir de los años 90 llevó consigo la creación de Centros de Estudio de Software en los diferentes Institutos Superiores del país y la instauración de bancos de software para organizar las producciones existentes. En la actualidad nuestro país se ha visto inmerso en el desarrollo de sistemas modernos que logren resolver nuestros problemas económicos y sociales. Las nuevas tecnologías han dado paso a que sea necesario modelar sistemas para obtener productos que satisfagan las necesidades de los clientes.

Esta revolución tecnológica se ha generalizado para dar paso a un arduo estudio con el que se le puede dar solución a una serie de problemas que serían imposibles resolver sin su automatización. La mayoría de nuestras empresas de software han tenido que definir una metodología para la creación de sus sistemas y es ahí, a la hora de desarrollar un proyecto, donde se hace tan necesario la aplicación de la ingeniería de software desde que iniciamos el mismo.

Esta investigación se realiza debido al poco conocimiento que los desarrolladores de software tienen de las metodologías de desarrollo para cada tipo de aplicación. Muchas de estas personas hacen sus programas sin metodología, lo que conlleva muchas veces a obtener productos con errores y usuarios insatisfechos.

Para conocer el nivel de conocimiento de los desarrolladores se realizaron encuestas en empresas y centros universitarios.

Descripción de la Encuesta

Para hacer un diagnóstico del estado actual y nivel de conocimiento de los desarrolladores a la hora de utilizar metodologías de desarrollo para el crear Hipermedia y Web, multimedia y otros tipos de aplicaciones, fue necesario aplicar encuestas para obtener información cuantitativa y cualitativa que ayudaran a justificar esta investigación.

Las preguntas de la encuesta fueron diseñadas para conocer el nivel de conocimiento del personal que desarrolla proyectos de software, en cuanto a las metodologías de desarrollo de Hipermedia y Web, multimedia y otro tipo de aplicaciones.

En la elaboración de la encuesta se combinaron los dos tipos de preguntas fundamentales, abierta y cerradas. Las preguntas cerradas se hicieron con el interés de conocer la información cuantitativa con respecto al tema y las preguntas abiertas con el objetivo de saber la opinión del encuestado acerca de las metodologías que proponían para el desarrollo de cada aplicaciones plasmada en las encuestas.

Análisis de las encuestas

Las encuestas fueron aplicadas en empresas que desarrollan software, como son el caso de Desoft (Holguín), Serconi (Moa), y EMNI (Moa). También se realizaron las mismas en la Universidad de Holguín (UHO) y la Universidad de Moa (ISMM).

Total de encuestados: 26

Universidad de Holguín: 6

Desoft (Holguín): 6

Serconi: 6

ISMM: 5

EMNI: 3

Como metodología más conocida y utilizada en estos centros, está RUP, la que utilizan para cualquier tipo de proyecto sin que importe el tamaño, ni el grupo de trabajo con el que cuenten. Es por esto que muchos proyectos fracasan debido a que no definen la metodología adecuada para crear su aplicación.

El resultado de las encuestas lo podemos encontrar en los anexos 2 y 3.

Problema

La inexistencia de un sistema informativo acerca de las metodologías de desarrollo de aplicaciones multimedia, Hipermedia y Web

Objeto de estudio

Las metodologías de desarrollo de software.

Campo de acción

El proceso de desarrollo de Hipermedia y Web, aplicaciones multimedia.

Idea a defender

La elaboración de una guía metodológica para proyectos de software, favorecerá el desarrollo de aplicaciones multimedia, Hipermedia y Web.

Objetivo General

Elaborar una guía metodológica para el desarrollo de aplicaciones multimedia, Hipermedia y Web, con la finalidad de que pueda ser empleada satisfactoriamente por los desarrolladores de software.

Objetivos específicos

1. Seleccionar las diferentes metodologías que existen para desarrollo de aplicaciones multimedia, Hipermedia y Web.
2. Proponer y desarrollar una metodología para cada tipo de aplicación.
3. Validar la presente guía metodológica a través un ejemplo práctico para cada tipo de aplicación.

Tareas:

1. Realizar un estudio que abarque las tendencias actuales de la aplicación de las diferentes metodologías para el desarrollo de proyectos de software a nivel mundial y nacional.
2. Comprender los principios y bases científicas de los tipos de aplicaciones tratadas en la tesis.
3. Realizar encuestas a desarrolladores de software, líderes de proyecto y procesar los resultados de las encuestas.
4. Seleccionar y desarrollar una metodología apropiada para una determinada Aplicación.
5. Realizar un ejemplo de una de las metodologías propuestas en cada tipo de Aplicación.

CAPÍTULO I. FUNDAMENTACIÓN TEÓRICA

1.1. Metodologías de desarrollo de software

En la actualidad debido al desarrollo tecnológico la mayoría de los países se ven con la necesidad de crear sistemas complejos basados en computadoras. Estos sistemas facilitan el trabajo que con otros medios no se podría realizar. Debido al auge en la informatización surgió una rama dentro de su contexto llamada Ingeniería de Software para darle solución a todo el proceso de desarrollo de software.

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 del proyecto y de la aplicación, los controles y las entregas a realizar.

Uno de los problemas más importantes con los que se enfrentan los ingenieros en software y los programadores en el momento de desarrollar una aplicación, es la falta de marcos teóricos comunes que puedan ser usados por todas las personas que participan en el desarrollo del proyecto informático para aplicaciones generales.

1.1.1. Definición de metodología de desarrollo de software

En el manual de García Rubio, F. O. y Bravo Santos, C. (2008) se definen tres conceptos claros sobre metodologías de desarrollo de software; ellos son:

1. Conjunto de pasos y procedimientos que deben seguirse para el desarrollo de software.
2. Conjunto de filosofías, fases, procedimientos, reglas, técnicas, herramientas, documentación y aspectos de formación para los desarrolladores de SI (Maddison, 1983).
3. Conjunto de procedimientos, técnicas, herramientas y soporte documental que ayuda a los desarrolladores a realizar nuevo software.

En un proyecto de desarrollo de software la metodología define quién debe hacer qué, cuándo y cómo debe hacerlo. Es un proceso y está compuesta por:

- Cómo dividir un proyecto en etapas
- Qué tareas se llevan a cabo en cada etapa
- Qué restricciones deben aplicarse
- Qué técnicas y herramientas se emplean
- Cómo se controla y gestiona un proyecto.

1.1.2. Objetivos de las metodologías de desarrollo de software

Según García Rubio, F. O. y Bravo Santos, C. (2008) los objetivos de las metodologías de desarrollo de software son los siguientes:

- Obtener mejores aplicaciones, conducentes a una mejor calidad
- Obtener un mejor Proceso de Desarrollo controlado que identifique salidas (o productos intermedios) de cada fase de forma que se pueda planificar y controlar el proyecto
- Obtener un Proceso Estándar en la organización, no dependiente del personal.

1.1.3. Clasificación de las metodologías

García Rubio, F. O. y Bravo Santos, C. (2008) muestran una clasificación en función de los procedimientos metodológicos para el desarrollo de software, considerando el enfoque, tipo de sistema y formalidad. Según esta clasificación ellas pueden ser estructuradas y orientadas a objetos (tabla 1.1).

Tabla 1.1. Clasificación de las metodologías de desarrollo de software (García Rubio, F. O. y Bravo Santos, C. (2008).

ENFOQUE	TIPO DE SISTEMA	FORMALIDAD
ESTRUCTURADAS <ul style="list-style-type: none"> ➤ Orientadas a procesos ➤ Orientadas a datos <ul style="list-style-type: none"> • Jerárquicos • No jerárquicos ➤ Mixtas 	GESTIÓN	NO FORMAL
ORIENTADAS A OBJETOS	TIEMPO REAL	FORMAL

➤ **Metodologías estructuradas.**

Estas metodologías cuentan con la siguiente especificación estructurada:

- Diagramas de Flujo de Datos
- Diccionario de Datos
- Especificaciones de procesos

Las metodologías estructuradas proponen la creación de modelos del sistema que representan:

- Los procesos
- Los flujos
- La estructura de los Datos

Dentro de las metodologías estructuradas se encuentran:

I. Metodologías orientadas a procesos



Elas se apoyan en técnicas gráficas para obtener:

- Modelo gráfico, particionado, descendente y jerárquico de los procesos del sistema y de los datos utilizados por éstos.
- Componentes:
 - Diagrama de Flujo de Datos
 - Diccionario de Datos
 - Especificaciones de Procesos

II. Metodologías orientadas a datos jerárquicos



- La estructura de control del programa debe ser jerárquica y se debe derivar de la estructura de datos del programa.
- El proceso de diseño consiste en definir primero las estructuras de los datos de entrada y salida, mezclarlas todas en una estructura jerárquica de

programa y después ordenar detalladamente la lógica procedimental para que se ajuste a esta estructura.

- El diseño lógico debe preceder y estar separado del diseño físico.

III. Metodologías orientadas a datos no Jerárquicos



Metodología “Ingeniería de la Información”

- **Planificación:** construir una arquitectura de la Información y una estrategia que soporte los objetivos de la organización.
- **Análisis:** comprender las áreas del negocio y determinar los requisitos del sistema.
- **Diseño:** establecer el comportamiento del sistema deseado por el usuario y que sea alcanzable por la tecnología.
- **Construcción:** construir sistemas que cumplan los tres niveles anteriores.

➤ Metodologías Orientadas a Objetos

La orientación a objetos (OO) unifica procesos y datos encapsulándolos en el concepto de objetos. Tiene dos enfoques distintos:

1. **“Revolucionarios” o “Puros u ortodoxo”** Rompen con las metodologías tradicionales.
 - La OO se entiende como un cambio profundo de las metodologías estructuradas que se ven como obsoletas.
 - Ejemplos: metodolog'ia OOD (Booch), CRC/RDD (Wirfs-Brock)
2. **“Sintetistas” o “evolutivos”**
 - Análisis y Diseño Estructurado se consideran como la base para el desarrollo OO.
 - Ejemplo : Metodología OMT de Rumbaugh, UML

1.1.4. Características Deseables de una Metodología

Según García Rubio, F. O. y Bravo Santos, C. (2008) las características deseables de una metodología son las siguientes:

- Existencia de reglas predefinidas
- Cobertura total del ciclo de desarrollo
- Verificaciones intermedias
- Planificación y control
- Comunicación efectiva
- Utilización sobre un abanico amplio de proyectos
- Fácil formación
- Herramientas CASE
- Actividades que mejoren el proceso de desarrollo
- Soporte al mantenimiento
- Soporte de la reutilización de software.

1.1.5. Elementos que hay que tener en cuenta para construir o elegir una metodología.

Según Rafael Menéndez-Barzanallana Asensio (Extraído de: <http://www.um.es/docencia/barzana/IAGP/lagp3.html>) para construir la metodología hay que tener en cuenta los siguientes elementos:

- **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.

- **La metodología debe cubrir el ciclo entero de desarrollo de software**

Para ello la metodología ha de realizar siguientes etapas: Investigación, análisis de requisitos y diseño.

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

Dentro de este aspecto Rafael Menéndez-Barzanallana resalta la rastreabilidad y la fácil interacción entre etapas del ciclo de desarrollo. Relacionado con el primero 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. Con relación al segundo es necesario 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.

- **La metodología debe de incluir la realización de validación**

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.

- **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. Esto implica una comunicación entre usuario y técnico amigable y sencillo, exento de consideraciones técnicas.

- **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.

- **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 la 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 una buena técnica de diagramación.

- **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.

- **La metodología debe 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, 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 de departamentos o varias empresas.

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

- **La metodología debe ser capaz de 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.

- **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.

- **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 (En: <http://www.um.es/docencia/barzana/IAGP/lagp3.html>. Capítulo 3. IAGP 2005/06. Metodologías usadas en ingeniería del software).

1.1.6. Aspectos para cada metodología

La elaboración de una metodología pasa inevitablemente por las fases de desarrollo, que básicamente son: requisitos, análisis, diseño preliminar, diseño detallado, codificación y pruebas. A continuación se detallan cada componente de esta.

1. Requisitos: Los requisitos son una descripción de las necesidades o deseos de un producto. La meta primaria de la fase de requerimiento es identificar y documentar lo que en realidad se necesita, en una forma que claramente se lo comunique al cliente y a los miembros del equipo de desarrollo (Craig, L., 2004).

2. Análisis: Durante el análisis se va a documentar lo que vamos a definir más claramente qué es lo que va a hacer nuestro programa. Debemos hacer varias cosas principalmente:

- **Identificar actores.** En lenguaje UML, actores son los usuarios y cualesquiera otros sistemas con los que se pueda comunicar nuestro programa.
- **Identificar casos de uso.** Un caso de uso es algo que un actor quiera hacer con nuestro sistema.
- **Detallar casos de uso.** En un texto vamos poniendo varios puntos. En cada punto ponemos sentencias del estilo "el usuario hace tal cosa y el programa hace tal otra". Es decir, explicamos por escrito, desde el punto de vista del usuario, qué es lo que tiene que hacer y qué es lo que va a hacer el ordenador.

- **Diagrama de clases del negocio.** Es un diagrama de clases de objetos que tienen sentido para el usuario. Este diagrama a este nivel tiene dos utilidades: Asegurar que los que saben del tema y los que hacen el programa están hablando de lo mismo, y servir de base para el diseño detallado orientado a objetos.

3. Diseño Preliminar: Aquí ya empezamos a pensar en cómo vamos a hacer las cosas. En el diseño preliminar tratamos de establecer la arquitectura de nuestro programa. La arquitectura es un esquema de en qué módulos/paquetes vamos a dividir nuestro programa, qué librerías. Si el programa es suficientemente grande, quizás vaya en varios ejecutables, una arquitectura cliente/servidor, etc. Viendo los casos de usos, deberíamos ver qué cosas podemos hacer comunes o como librerías aparte, que podamos reutilizar. Es importante en este paso definir las interfaces y relaciones entre paquetes. Para ello puede servir de ayuda hacer los diagramas de secuencia de los casos de uso mostrando los actores, los paquetes y los mensajes entre ellos.

4. Diseño detallado: En el diseño detallado ya se entra a nivel de clases y métodos. Por cada paquete que hayamos extraído en el paso anterior y siguiendo siempre los casos de uso, debemos ir detallando las clases que vamos a implementar y los métodos que van a tener. Detallamos aun más los casos de uso y las interfaces de las clases.

En este punto pueden ser de ayuda los patrones de diseño. Algunos de los problemas salen con mucha frecuencia y se han establecido soluciones "estándar", que funcionan bien. Dichas soluciones se llaman patrones. No está de más leer algo sobre patrones de diseño orientados a objetos para tener una lista de "soluciones" a aplicar. Hay incluso patrones a nivel de arquitectura. Es bastante conocido, por ejemplo, el patrón separación modelo-vista.

5. Implementación y Pruebas: Pues a ello. Hay muchas cosas que se podrían contar aquí, pero no son de diseño. De las pruebas podemos decir que hay que escribir los "casos de prueba". Básicamente son como la descripción de los casos de

uso, pero indicando datos concretos que el operador va a introducir y qué resultados exactos debe dar nuestro programa.

1.1.7. Metodología más adecuada

Es aquella que permite conseguir el éxito del proyecto (en: <http://seis.wordpress.com/2007/06/10/metodologia/>). Teniendo en cuenta los múltiples factores externos e internos que intervienen en el trabajo del equipo del proyecto, en ocasiones es necesario disponer de diversas metodologías para las distintas áreas del proyecto; por ejemplo, en un desarrollo de software, una metodología de gestión de proyectos y una metodología de desarrollo de software pueden ensamblarse para llevar a cabo los objetivos del proyecto. En muchas ocasiones, la aplicación de una metodología suscita rechazo, por lo que hay que tener presentes los **beneficios** que proporciona:

- Permite **enfocar** el proyecto, al tener presente desde el primer momento todos los procesos que se llevarán a cabo
- Aumenta las posibilidades de éxito, ya que la metodología se basa en el **aprendizaje** obtenido en proyectos anteriores
- Hace hincapié en las **interdependencias** entre procesos, actividades, equipos distintos, etc., proponiendo formas de actuación
- **Alinea** los recursos, capacidades y necesidades del proyecto
- Es una receta **preventiva**, que busca anticiparse a los problemas y atajar los riesgos lo antes posible
- Promueve la **involucración** de la organización en el proyecto
- **Facilita** que el proyecto termine en las fechas marcadas, con los recursos asignados y con el coste previsto
- **Capitaliza la experiencia** del proyecto de forma que pueda ser empleada en el futuro.

1.1.8. Importancia de las metodologías

La importancia de las diferentes metodologías reside en que integran un conjunto de conceptos, principios y técnicas, para ofrecer una respuesta coherente a las grandes dificultades que tradicionalmente encuentran los proyectos de construcción de

sistemas, entre los cuales los más comunes son (En: <http://www.rhernando.net/modules/tutorials/viewexttutorial.php?tid=15>):

- Problemas de comunicación
- Confusión entre el qué y el cómo
- Dificultad para manejar la complejidad
- Participación del usuario.

1.2. Desarrollo Hipermedia y Web.

La Hipermedia se ha convertido en una tecnología madura de la que existe una base teórica y una establecida comunidad de investigadores que trabajan en aspectos de muy diversa índole, desde temas puramente técnicos, como pueden ser la definición de nuevas arquitecturas, la aplicación de técnicas de ingeniería del software para producir sistemas de calidad, la especificación de modelos de referencia que recojan las características y el funcionamiento de este tipo de sistemas, la integración eficiente de material multimedia o el desarrollo de interfaces de usuario avanzadas; hasta otros de carácter más literario como son la exploración de nuevos modelos de lectura y escritura para mejorar la comprensión de la información .

Desde el punto de vista de la aplicación, los sistemas Web pueden considerarse como un subconjunto de los sistemas Hipermedia. En tal sentido, el concepto de sistema Web está englobado en el de hipermedia, a sabiendas de que existen múltiples aspectos adicionales, como aquellos relacionados con los protocolos de comunicación o los servicios de bajo nivel ofrecidos por los servidores de información, que constituyen de por sí una interesante y activa área de investigación.

1.2.1. Hipermedia

Hipermedia es el término con que se designa al conjunto de métodos o procedimientos para escribir, diseñar, o componer contenidos que tengan texto, video, audio, mapas u otros medios, y que además tenga la posibilidad de interactuar con los usuarios (en: <http://es.wikipedia.org/wiki/Hipermedia>).

Hipermedia =Estructura hipertexto+Información Multimedia

Es cuando se proporciona una estructura ligados a través de los cuales el usuario puede navegar, entonces, multimedia interactiva se convierte en Hipermedia.

1.2.2. Tipos de Hipermedia

Entre los tipos de hipermedia se encuentran (disponibles en: <http://es.wikipedia.org/wiki/Hipermedia>)

- Hipertexto
- Hiperfilmes
- Hipergrama.

1.3. Desarrollo de Aplicaciones Multimedia

1.3.1. ¿Qué es Multimedia?

Multimedia se puede definir desde los tres puntos de vista siguientes:

1. Como integración de hardware: Estamos hablando de las características que debe tener un ordenador para poder responder al reto multimedia. Vídeo, sonido, CD-ROM, comunicación con escáneres y cámaras digitales, etc.
2. Como integración de software: Los programas informáticos cada vez se interrelacionan mejor, los archivos y datos son intercambiables. Los programas de autor permiten la integración de sonido, vídeo, texto, animación, imagen...
3. Como integración de medios: Cada vez más, los medios clásicos de comunicación se interrelacionan. Por ejemplo, podemos oír la radio por Internet, recibir el correo electrónico en el teléfono móvil, leer la prensa en la pantalla del ordenador... Es decir, los medios no son independientes, sino que necesitan unos de otros y se prestan mutuamente sus posibilidades.

1.3.2. ¿Qué son las aplicaciones multimedia?

Una aplicación Multimedia (En: <http://claseinscoop.wordpress.com/2008/02/09/que-son-aplicaciones-multimedia/>) es un programa que permite interactuar con imágenes, vídeo, sonido, animación y texto. Todos estos elementos han de organizarse y controlarse de forma que lleguen al usuario final como una aplicación cómoda de utilizar, en la que puede desplazarse a voluntad por los distintos elementos que forman la aplicación. Los programas que permiten crear estas aplicaciones se llaman

herramientas de desarrollo Multimedia; entre los más conocidos están: Macromedia Director, Macromedia Authorware, Asymetrix ToolBook, Scala Multimedia, Neobook de NeoSoft, etc.

1.3.3. Usos más frecuentes de aplicaciones multimedia

La utilización de las aplicaciones Multimedia es hoy en día y, gracias a la potencia gráfica, velocidad de los ordenadores actuales, casi infinita. Podemos crear aplicaciones interactivas para negocios, juegos, cursos, presentaciones de un producto, publicidad, catálogos, mercadotecnia, educación y entretenimiento con ayuda de computadoras.

1.3.4. Tipos de información Multimedia

Los tipos de información multimedia son (En: <http://es.wikipedia.org/wiki/Multimedia>)

Texto: (palabra, oración, párrafo, capítulo (letra pura)) sin formatear, formateado, lineal e hipertexto.

Gráficos: utilizados para representar esquemas, planos, dibujos lineales.

Imágenes: son documentos formados por pixeles. Pueden generarse por copia del entorno (escaneado, fotografía digital) y tienden a ser ficheros muy voluminosos.

Animación: presentación de un número de gráficos por segundo que genera en el observador la sensación de movimiento.

Vídeo: Presentación de un número de imágenes por segundo, que crean en el observador la sensación de movimiento. Pueden ser sintetizadas o captadas.

Sonido: música, voz, ruidos u otros sonidos.

CAPÍTULO II. METODOLOGÍAS PARA EL DESARROLLO APLICACIONES MULTIMEDIA DE HIPERMEDIA Y WEB.

El desarrollo de aplicaciones Web involucra decisiones no triviales de diseño e implementación que inevitablemente influyen en todo el proceso de desarrollo, afectando la división de tareas. Los problemas involucrados, como el diseño del modelo del dominio y la construcción de la interfaz de usuario, tiene requerimientos disjuntos que deben ser tratados por separado.

Los principales problemas que nos encontramos es la falta de fiabilidad, seguridad, escalabilidad, mantenimiento, integración y la alta dependencia para su desarrollo e implantación junto con la falta de estándares. Para todo esto se han desarrollado metodologías que permiten estructurar comunicar, entender, simplificar y formalizar tanto el dominio como las decisiones de diseño, así como disponer de documentación detallada para posibles cambios del software.

En los últimos años, ha aparecido un interés creciente en el desarrollo de los sistemas de información en la Web, lo que mediante la práctica ha demostrado que los mismo requieren un tratamiento especial durante el proceso de desarrollo, siendo necesario proponer nuevos modelos y técnicas que den soporte al equipo de trabajo asegurando la calidad del producto resultante.

En este capítulo se presenta un resumen de las metodologías más utilizadas en el desarrollo de aplicaciones Hipermedia y Web. Teniendo como base la necesidad de elevar el conocimiento estas para este tipo de aplicaciones se propone a Navigational Development Techniques (NDT) como una metodología adecuada para el desarrollo Hipermedia y Web teniendo en cuenta que una técnica que permite realizar la especificación, el análisis y el diseño de la navegación de un sistema y que ofrece una guía sistemática para el tratamiento de la navegación y la interfaz.

En el epígrafe 2.4, último del capítulo 2 de la presente tesis de grado se desarrolla la metodología MULTIMET para lo cual se elaboró un ejemplo práctico de su aplicación en el capítulo 3.

2.1 Metodologías de desarrollo Web

En la actualidad la mayoría de las empresas y organismos han encontrado en Internet, una vía de trabajo que les permite contactar con cualquier persona y poder acceder a sus informaciones desde el lugar donde se encuentre. Todo esto, unido al avance de las comunicaciones y al aumento en las prestaciones de los equipos, redes y vías de transmisión, ha producido que muchos de los sistemas actuales hayan sido desarrollados o emigrados a entornos Web.

Desde que se comenzó a plantear el desarrollo de sistemas software en la Web, la comunidad científica detectó la necesidad de proponer nuevos entornos metodológicos, técnicas y modelos que ofreciesen un marco de referencia adecuados a las nuevas características de este entorno.

El desarrollo de aplicaciones Web involucra decisiones no triviales de diseño e implementación que inevitablemente influyen en todo el proceso de desarrollo, afectando la división de tareas.

Los principales problemas que nos encontramos en las aplicaciones Web son la falta de fiabilidad, seguridad, escalabilidad, mantenimiento, integración y la alta dependencia para su desarrollo e implantación junto con la falta de estándares. Para todo esto se han desarrollado metodologías que permiten estructurar, comunicar, entender, simplificar y formalizar tanto el dominio como las decisiones de diseño, así como disponer de documentación detallada para posibles cambios del software.

Las relaciones entre las diferentes metodologías se muestran en la figura 2 (Escalona Cuaresma, M.J. (2004)). En esta figura se ofrece una visión esquemática de ellas y el orden de aparición que han ido teniendo. Algunas de estas propuestas en la actualidad no se utilizan, en el caso de HDM o RMM, que son las únicas orientadas dentro del paradigma estructurado.

La figura 2.1 muestra las relaciones que se producen entre estas propuestas. A pesar de que las propuestas son diferentes y las ideas y formas de concebir el proceso de ingeniería Web difiere de un grupo a otro, muchas propuestas han asumido ideas y experiencias de grupos y metodologías anteriores para definir sus

aproximaciones. Esta relación se muestra en la figura mediante líneas continuas dirigidas. De esta forma, si se analiza la propuesta de UWE se puede observar que tiene grandes influencias de parte de OOHDM.

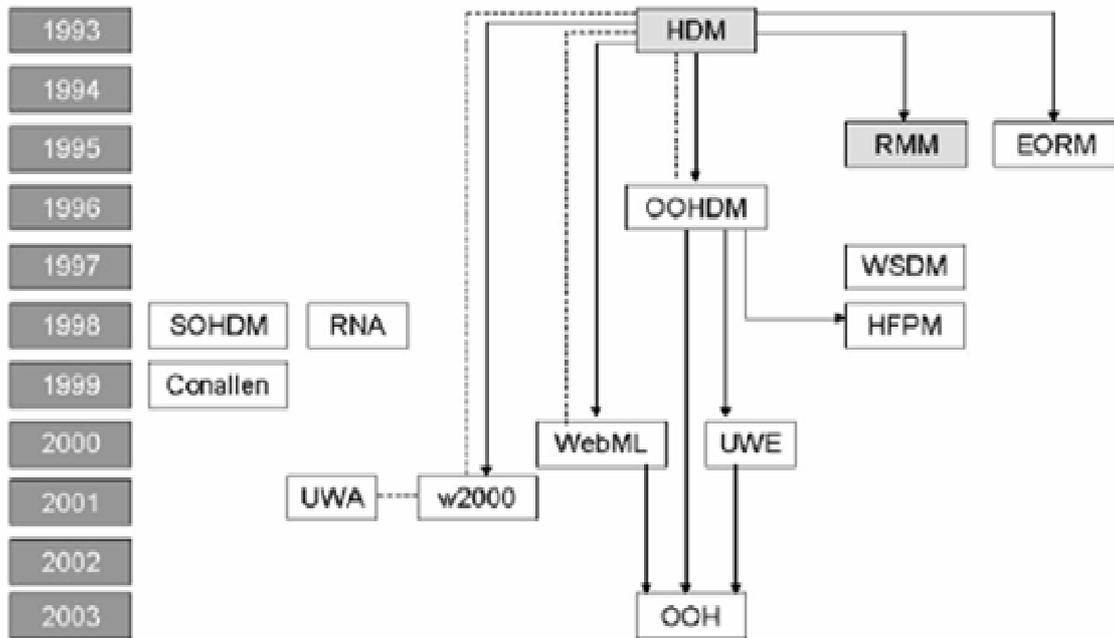


Figura 2.1. Relaciones entre las diferentes metodologías (Según Escalona Cuaresma, M.J., 2004).

En otros casos, la relación entre los entornos metodológicos resulta más fuerte en el sentido de que incluso parte del equipo de desarrollo de una propuesta forman parte del equipo de desarrollo de otra, o incluso, hay propuestas como el proyecto UWA que incorpora totalmente una o varias fases de otras propuestas como W2000. Esta relación viene representada mediante una línea discontinua.

Puede observarse en la figura 2.1 que la aparición de HDM ha sido un hito en el entorno de las metodologías para la Web. A pesar de que HDM no se utiliza, la necesidad que plantea de ofrecer nuevas vías de trabajo dentro del entorno de la Hipermedia, proponiendo nuevos elementos de modelado para el trabajo con la Hipermedia, resultó una fuente de inspiración para muchas propuestas. Las inquietudes propuestas y analizadas en HDM han servido para nutrir metodologías

tan importantes como OOHDM, W2000 o WebML, que a su vez, también han servido como fuente de inspiración para propuestas posteriores.

Otro hito destacable es OOHDM y la idea de separación de conceptos que propone, separando el tratamiento de lo conceptual, de la navegación y de la interfaz, así como el uso de diagramas de clases navegacionales, son de las propuestas más aceptadas dentro de los grupos de investigación. De hecho, propuestas como UWE, OOH o HFPM las han tomado como válidas en sus propuestas de desarrollo.

Las metodologías de desarrollo de Hipermedia y Web se explican de forma resumida a continuación.

2.1.1. Hypermedia Design Model (HDM)

HDM [Garzotto et al. 1993: En: Escalona Cuaresma, M.J. (2004).] fue la primera aproximación que se hizo para dar soporte al desarrollo de sistemas de información Hipermedia. El objetivo era crear un modelo que fuera de utilidad para realizar el diseño de una aplicación de hipertexto. No es realmente una metodología de desarrollo. En realidad, HDM es una extensión del modelo entidad-relación, en el que se incluyen nuevos aspectos para el modelado de sistemas hipermedia.

HDM propone un conjunto de elementos que incorporados al modelo entidad-relación (MER) permiten al diseñador especificar una aplicación. Estos elementos son las entidades, los componentes, las perspectivas, las unidades y los enlaces. Una entidad es la unidad mínima autónoma de cualquier modelo HDM.

La entidad está compuesta por una jerarquía de componentes que heredan las propiedades de dicha entidad.

Los componentes no tienen razón de ser sin que exista la entidad de la que dependen. Los componentes son, por su parte, abstracciones para diseñar un conjunto de unidades o nodos que representan un mismo conjunto de información de la entidad.

Una unidad, es pues un depósito de la información contenida en una aplicación. Una unidad representa un fragmento del contenido de una entidad presentada bajo una perspectiva particular.

La perspectiva permite representar la multiplicidad de presentaciones de un mismo contenido de información (por ejemplo, la presentación de un documento en múltiples lenguas).

En HDM las entidades son agrupadas en tipos de entidad. Los tipos de entidad se caracterizan por un nombre, por un conjunto de perspectivas bajo las que se pueden presentar su contenido y un conjunto de enlaces de aplicación por los que se puede navegar (fig. 2.2)

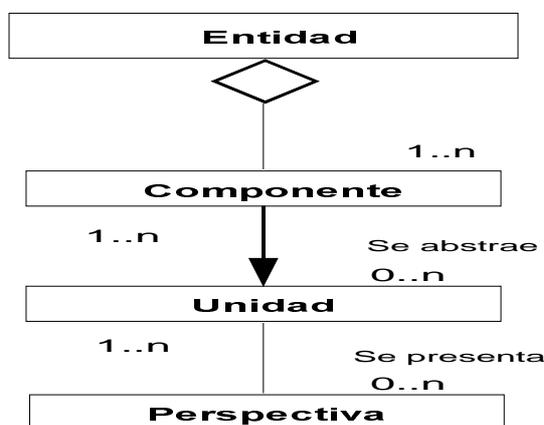


Figura 2.2. Relación entre los elementos de HDM

En la actualidad HDM no se usa, principalmente porque el paradigma estructurado ha dejado paso al paradigma de la orientación a objetos y no ofrece un proceso metodológico completo. HDM ha sido el referente para muchas propuestas tales como RMM, OOHDM o W2000 que han aceptado y asumido las ideas de su propuesta.

2.1.2. Relationship Management Method (RMM)

RMM [Izakowitz et al. 1995 en: Escalona Cuaresma, M.J. (2004).], 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 propuesta está basada en el modelo entidad-relación y en HDM. Partiendo de ellos define un nuevo modelo, el modelo RMDM, que propone un lenguaje que permite describir los objetos del dominio, sus interrelaciones y los mecanismos de navegación hipertexto de la aplicación.

Fases:

Fase 1. Realización del modelo E-R

Fase 2. Diseño de slices.

Fase 3. Diseño de la navegación

Fase 4. Definición del protocolo de conversión

Fase 5. Diseño de la interfaz

Fase 6. Implementación de la aplicación

Fase 7. Fase de pruebas de la aplicación

El proceso comienza con la realización del modelo E-R sin entrar en detalles de navegación o presentación y tratando el sistema hipertexto como a cualquier sistema clásico. Tras esto se hace el diseño de slices. Un slice es un subconjunto de atributos de una entidad que van a ser presentados de forma agrupada al usuario, es lo que se podría llamar una vista del sistema.

El diseño de slices es enriquecido durante la siguiente fase de diseño de la navegación, en la que se indica como se podrá pasar de una entidad a otra. El diseño de slices y el enriquecimiento del mismo con los aspectos de la navegación generan el modelo RMDM. Basándose en este modelo, en la próxima fase se define el protocolo de conversión que va a describir el proceso a seguir para pasar del modelo RMDM a la plataforma de desarrollo concreta. Tras esto, se realiza el diseño de la interfaz en el que se definen las pantallas tal y como se presentarán al usuario. Conocida la interfaz, se pasa a la implementación de la aplicación en el lenguaje concreto elegido. Y por último, en la se realizan las pruebas de la aplicación.

2.1.3. Enhanced Object Relationship Methodology (EORM)

EORM [Lange, 1995 en: Escalona Cuaresma, M.J. (2004).] es una de las primeras propuestas para la Web centrada en el paradigma orientado a objetos (OO).

Estructura el proceso iterativo de desarrollo en tres fases: análisis, diseño e implementación.

Fases:

Fase 1. Análisis: Consiste en hacer un modelo orientado a objetos, siguiendo las pautas y nomenclatura de OMT para representar la aplicación. En esta fase aspectos como la navegación o la interfaz no son tenidos en cuenta.

Fase 2. Diseño: Se procede a modificar el modelo de objetos obtenido en la fase anterior añadiendo semántica suficiente a las relaciones para representar los enlaces.

Fase 3. Implementación: Los aspectos obtenidos en las fases anteriores son traducidos al lenguaje de programación.

2.1.4. Object-Oriented Hypermedia Design Method (OOHDM)

Esta propuesta nace basada en HDM pero dentro del paradigma de la orientación a objetos. OOHDM [Rossi, 1996 en: Escalona Cuaresma, M.J. (2004).] es una propuesta metodológica ampliamente aceptada para el desarrollo de aplicaciones Web.

OOHDM es un método basado en modelos para el desarrollo de sitios Web, sistemas de información, presentaciones multimediales, quioscos de información, etc. Se trata por separado el diseño de componentes, el modelo de navegación y el diseño de la interfaz.

Otra de las ideas que OOHDM propuso y que ha tenido mucha aceptación ha sido el hecho de separar el modelado de los aspectos de los sistemas hipermedia. El modelar lo conceptual, lo navegacional y la interfaz abstracta de manera independiente ha dado muy buenos resultados y ha sido asumido en muchas propuestas posteriores como se analiza más adelante.

Fases:

Fase 1. Diseño conceptual o análisis de dominio: Esta fase utiliza el método de análisis Orientado objeto para obtener esquemas conceptuales de las clases y de las relaciones entre las mismas). Durante esta actividad se construye un esquema

conceptual representado por los objetos del dominio, las relaciones y colaboraciones existentes establecidas entre ellos. Mediante un diagrama de clases, se representa la estructura estática del sistema. En las aplicaciones hipermedia convencionales, cuyos componentes de hipermedia no son modificados durante la ejecución, se podría usar un modelo de datos semántica estructural (como el modelo E/R). De este modo, en los casos en que la información base pueda cambiar dinámicamente o se intente ejecutar cálculos complejos, se necesitará enriquecer el comportamiento del modelo de objetos.

El análisis de requisitos entra en esta fase, requisitos funcionales y no funcionales.

Fase 2. Diseño navegacional: En esta fase se diseña el modelo de navegación. Este modelo ofrece una vista del modelo conceptual y expresa cómo se podrá navegar a través de la información representada en el modelo conceptual. Un modelo navegacional es construido como una *vista* sobre un diseño conceptual, admitiendo la construcción de modelos diferentes de acuerdo con los diferentes perfiles de usuarios. Cada modelo navegacional provee una vista subjetiva del diseño conceptual. En OOHDM la navegación es considerada un paso crítico en el diseño de aplicaciones.

Fase 3. Diseño de interfaz abstracta: Esta fase se apoya en un modelo orientado a objeto para especificar la estructura y el comportamiento de la interfaz del sistema. Este modelo se crea a través de 3 tipos de diagramas: diagrama abstracto para cada clase, diagrama de configuración para reflejar los eventos externos y diagramas de estados para señalar el comportamiento dinámico. Este modelo ofrece una vista de cómo se va a presentar la información al usuario. De esta forma, el modelo de navegación es una vista del conceptual y el de interfaz abstracta es una vista del navegacional.

En OOHDM se utiliza el diseño de interfaz abstracta para describir la interfaz de usuario de aplicación de hipermedia. El modelo de interfaz ADVs (Vista de Datos Abstracta) especifica la organización y comportamiento de la interfaz, pero la apariencia física real o de los atributos, y la disposición de las propiedades de las

ADVs. en la pantalla real son hechas en fase de implementación. En la especificación de la interfaz se define cómo van a ser los objetos navegacionales.

Fase 4. Implementación: Se realiza la construcción de los programas en programación orientada a objetos. En la misma, el diseñador debe implementar el diseño. Hasta ahora, todos los modelos fueron construidos en forma independiente de la plataforma de implementación; en esta fase es tenido en cuenta el entorno particular en el cual se va a correr la aplicación. Al llegar a esta fase, el primer paso que debe realizar el diseñador es definir los ítems de información que son parte del dominio del problema. Debe identificar también, cómo son organizados los ítems de acuerdo con el perfil del usuario y su tarea; decidir qué interfaz debería ver y como debería comportarse. A fin de implementar todo en un entorno Web, el diseñador debe decidir además que información debe ser almacenada.

En todas estas fases, OOHDM ofrece lenguajes de modelado específicos para representar estos modelos. Algunos de ellos, como los contextos de navegación o las vistas abstractas de datos han caído en desuso en la actualidad. Sin embargo, otros como el hecho de representar la navegación mediante un diagrama de clases especiales denominadas *clases de navegación*, ha sido ampliamente aceptada por la comunidad investigadora.

2.1.5. Web Site Design Method (WSDM)

WSDM (De Troyer & Leune 1998 en: Escalona Cuaresma, M.J., 2004) es una propuesta para el desarrollo de sitios Web, en la que el sistema se define en base a los grupos de usuarios. Esta aproximación define una aplicación Web a partir de los diferentes grupos de usuarios que vaya a reconocer el sistema. Propone cuatro etapas: modelado de usuario, diseño conceptual, diseño de la implementación e implementación.

Fases:

Fase 1. Modelado de usuario: Durante la fase de realización del modelo de usuario, se estudian los diferentes roles de usuario que van a interactuar con el sistema y se clasifican según las relaciones que existan entre ellos. El modelo de usuario se

divide en dos subfases: clasificación y descripción. El tratamiento de requisitos se identifican y clasifican los usuarios que van a hacer uso de la aplicación Web y se describen los requisitos de cada grupo de usuarios.

Fase 2. Diseño conceptual: En esta fase, y basados en la clasificación de usuarios realizada, se realiza el modelado conceptual del sistema. El modelado conceptual no tiene el mismo significado que en OOHDM. Durante el modelado conceptual se realizan dos tareas a la vez: el modelado de objetos, que es lo que en OOHDM se llama modelo conceptual, y el diseño de la navegación, que coincide con la idea del diseño navegacional de OOHDM. En WSDM puede existir más de un modelo de navegación, dependiendo de los roles de usuario detectados durante la primera.

Fase 3. Diseño de la implementación: Se modela la interfaz para cada rol de usuario. Y por último, en la fase de implementación se codifican todos estos aspectos en el lenguaje concreto que se haya seleccionado. WSDM es también una propuesta viva que está cambiando y adaptándose a nuevos requisitos.

2.1.6. Scenario-based Object-oriented Hypermedia Design Methodology (SOHDM)

SOHDM (Lee et al.1998; Suh & Lee, 2001. En Escalona Cuaresma, M.J., 2004) es una de las primeras propuestas para la web que más importancia da a la tarea de tratamiento de requisitos. Esta propuesta presenta la necesidad de disponer de un proceso que permita capturar las necesidades del sistema. Para ello, propone el uso de escenarios.

SOHDM se inicia con la definición de un diagrama de contexto donde se identifican las entidades externas que se comunican con el sistema y los eventos que dicha comunicación provoca. Por cada uno de estos eventos se propone elaborar un escenario, en el cual se describe el proceso de interacción entre el usuario y el sistema especificando el flujo de actividades, los objetos involucrados y las transacciones realizadas.

A partir de estos escenarios SOHDM propone un proceso para la obtención del modelo conceptual del sistema, el cual es representado mediante un diagrama de clases.

Se caracteriza principalmente porque su ciclo de vida comienza con la aplicación de los escenarios como técnica de elicitación y definición de requisitos.

Fases:

Fase 1. Análisis: Se debe realizar un estudio de las necesidades de la aplicación, del entorno de trabajo y de los actores. La finalidad principal de esta fase es conseguir los escenarios que representen las actividades que se pueden llevar a cabo en el sistema. Éstos son representados gráficamente mediante los denominados *Scenario Activity Chart* (SACs) y describe el proceso de interacción entre el usuario y el sistema cuando se produce un evento determinado, especificando el flujo de actividades, los objetos involucrados y las transacciones realizadas.

Fase 2. Modelo de objetos: se realiza un modelado de objetos en el que se desarrolla un diagrama de clases que representa la estructura conceptual del sistema.

Fase 3. Diseño de vistas: los objetos son reorganizados en unidades navegacionales que representan una vista de los objetos del sistema.

Fase 4. Diseño navegacional: En esta fase se enriquecen dichas vistas definiendo los enlaces e hiperenlaces que existen en el sistema.

Fase 5. Diseño de la implementación: esta es la fase en la que se diseñan las páginas, la interfaz y la base de datos del sistema.

Fase 6. Construcción: es en la que se implementa la aplicación.

2.1.7. Relationship-Navigational Análisis (RNA)

RNA (Bieber, Galnares & Lu, 1998; En: Escalona Cuaresma, M.J. ,2004) plantea una secuencia de pasos para el desarrollo de aplicaciones Web, centrándose fundamentalmente en el flujo de trabajo de análisis.

A diferencia de otras propuestas, RNA no cubre varias fases del ciclo de vida y centra todo su proceso de desarrollo en la fase de análisis del sistema.

RNA se centra fundamentalmente en el flujo de trabajo del análisis y propone la realización de cinco pasos: Análisis del entorno, elementos de interés, análisis del conocimiento, análisis de la navegación e implementación del análisis.

En esta propuesta los requisitos son tratados en los dos primeros pasos. En el primero de ellos se realiza una tarea de identificación y clasificación de usuarios, mientras que en el segundo, se listan los elementos de interés de la aplicación como documentos, las pantallas que se van a requerir, información, etc.

Fases:

Fase 1- Análisis del entorno: el propósito de esta fase es el de estudiar las características de la audiencia. Consiste en determinar y clasificar a los usuarios finales de la aplicación en grupos según sus perfiles.

Fase 2- Definir los elementos de interés: en esta fase se listan todos los elementos de interés de la aplicación. Por elementos de interés se entienden los documentos, las pantallas que se van a requerir, la información, etc.

Fase 3- Análisis del conocimiento: esta fase consiste en desarrollar un esquema que represente a la aplicación. Para ello RNA propone identificar los objetos, los procesos y las operaciones que se van a poder realizar en la aplicación, así como las relaciones que se producen entre estos elementos.

Fase 4- Análisis de la navegación: en esta fase el esquema obtenido en la fase anterior es enriquecido con las posibilidades de navegación dentro de la aplicación.

Fase 5- Implementación del análisis: una vez obtenido el esquema final en el que ya se encuentran incluidos los aspectos de navegación, se pasa el esquema a un lenguaje entendible por la máquina.

Se puede concluir tras la presentación de su proceso de desarrollo, que el concepto de análisis no coincide exactamente con el de otras propuestas y casi se podría decir que cubre tareas de la especificación de requisitos, el análisis, el diseño o incluso la implementación.

La propuesta de RNA es quizás una de las que más ha resaltado la necesidad de trabajar con la especificación de requisitos, incluyendo tareas como el análisis del entorno y de los elementos de interés. Además, resulta interesante pues plantea la necesidad de analizar los requisitos conceptuales de manera independiente a los navegacionales.

2.1.8. Object Oriented Web Solution (OOWS)

OOWS (Fons et al. 2003. En: Escalona Cuaresma, M.J., 2004) es un método de desarrollo de aplicaciones Web que permite especificar sistemas de información para ambientes Web mediante la extensión de un método orientado a objeto (OO). Los sistemas de información Web representan un tipo de aplicación que tiene una base común con las aplicaciones de software tradicional, tales como la funcionalidad del sistema y la interacción con el usuario; pero además, estas aplicaciones introducen nuevas características navegacionales las cuales deben ser recogidas para representar el sistema de una manera más precisa y aproximada.

OOWS introduce estos nuevos conceptos de OO para permitir la noción de semántica navegacional y de presentación, las cuales corresponden a primitivas del modelo. El método utilizado como base para definir la aproximación fue OO-Method (Método OO de desarrollo de Software) el cual fue extendido adecuadamente permitiendo captar las propiedades funcionales del sistema que son consideradas relevantes para permitir construir una especificación textual OO y formal de manera automática. La especificación formal OO permite obtener un repositorio de información de alto nivel del sistema, el cual se utilizará posteriormente como entrada a un compilador (automático) de modelos conceptuales cuyo nombre es *model compiler*.

La definición del modelo de ejecución conjuntamente con la estrategia basada en patrones de traducción es lo que hace posible pasar de la especificación a la implementación operacional generando un prototipo completo del sistema.

OOWS es una propuesta para el tratamiento de requisitos y análisis de la navegación en la Web.

Fases:

Fase 1. Especificación del problema: En esta fase se captura el comportamiento del sistema que debe ofrecer. Para esta fase se capturan los requisitos mediante la aproximación de casos de uso y posteriormente las actividades de modelado conceptual del sistema. Para la parte del modelado conceptual las abstracciones derivadas del problema se especifican en términos de clases y de su estructura, comportamiento y funcionalidad, para lo cual se construyen los siguientes modelos: de objetos, dinámico, funcional, navegacional y de presentación. Esta fase además realiza un estudio de los tipos de usuarios que pueden interactuar con el sistema. En esta fase se deben capturar las peculiaridades y el comportamiento que el sistema debe ofrecer para satisfacer los requisitos de usuarios que han sido identificados. Para esta fase se capturan los requisitos mediante la aproximación de casos de uso y a posterior las actividades de modelado conceptual del sistema.

Fase 2. Desarrollo de la solución: En esta fase se propone una estrategia de generación de código basada en componentes que permite integrar la solución propuesta en ambientes Web. Esta fase se desarrolla de manera totalmente automática por intermedio de un compilador de modelos conceptuales (*Model Compiler*) el cual construye un sistema de software que recoge los requisitos de la aplicación que fue modelada. Esta estrategia permite facilidad de las tareas de mantenimiento y evolución, dado por la generación automática basada en patrones se realiza utilizando soluciones previamente probadas y validadas. Esta filosofía permite obtener de manera muy rápida aplicaciones finales con calidad, evitando la fase de pruebas (*testing*) del sistema.

Esquema Conceptual: El esquema conceptual de OOWS se compone de cinco modelos: el diagrama de clases, que define la estructura estática, los modelos dinámicos y funcional, que definen el comportamiento de sistema, el modelo de navegación, que definen los aspectos navegacionales y el modelo de presentación, que especifica las características de presentación de información.

2.1.9. Hypermedia Flexible Process Modelling Strategy (HFPM)

HFPM (Olsina 1998. En: Escalona Cuaresma, M.J., 2004) es quizás la propuesta más completa que se ha realizado en cuanto a las fases del ciclo de vida que cubre.

HFPM no se caracteriza por proponer nuevos modelos o técnicas y asume muchas de las ideas de OOHDM. Sin embargo, sí que cubre todo el proceso de desarrollo completo, incluso hasta la generación de los documentos a realizar. Esta propuesta define el tratamiento de requisitos con gran detalle; sin embargo, no ofrece técnicas concretas para llevar a cabo cada una de las tareas que propone, especialmente en la descripción de requisitos no funcionales.

Fases:

Fase 1. Modelado de los requisitos: El proceso comienza con un modelado de los requisitos que se deben cubrir y con la realización de la planificación. Tras esto, se realizan las fases de modelado conceptual, modelado navegacional y modelado de interfaz abstracta, que son similares a los de OOHDM.

Fase 2. Diseño del entorno: Se realiza el diseño del entorno, en el que se enriquecen los modelos anteriores mediante patrones de diseño. En la siguiente fase, de captura y edición de elementos multimedia, se deben plantear los múltiples medios con los que se va a trabajar, así como los sistemas de almacenamiento que se usarán en los mismos.

Fase 3. Implementación del sistema: Con todo este conocimiento, se pasa a la implementación del sistema.

Fase 4. Fase de validación y verificación: Y tras ello, a una fase de validación y verificación. En este punto, sobre el sistema se aplican una serie de métricas que permiten evaluar la calidad del mismo. Y para terminar, se contemplan las fases de mantenimiento y de generación de documentación como el manual de usuario o los documentos resultantes de cada fase.

2.1.10. Building Web Applications with UML

La propuesta de Conallen (Conallen 1999. En: Escalona Cuaresma, M.J., 2004), ha adquirido gran importancia en los últimos años. Se caracteriza por basarse en UML y en el proceso unificado en todo su ciclo de vida. Realiza muchas ampliaciones y definiciones de estereotipos propios para la Web, usando las posibilidades de extensión de UML.

Su ciclo de vida comienza con una fase previa de planificación. Tras ella, se comienza un proceso cíclico que cubre las fases de tratamiento de requisitos, análisis, diseño, implementación, pruebas y evaluación. Una vez aceptado el sistema, se pasa a la explotación del mismo.

Básicamente, los modelos y técnicas que ofrece están heredados de UML, pero los enriquece y define muchos nuevos elementos de modelado específicos para la web como la posibilidad de representar en los modelos de clase unidades java beans, frames, botones, etc. Estos nuevos estereotipos pueden incorporarse en la herramienta Rational Rose y así utilizarlos fácilmente durante la realización de los modelos.

2.1.11. Web Modelling Language (WebML)

WebML (Ceri et al. 2000;Ceri et al. 2003. En: Escalona Cuaresma, M.J., 2004) es una notación para especificar complejos sitios web en el ámbito conceptual. Permite una descripción de los sitios web desde distintos puntos de vista: el conceptual, el navegacional, el de presentación, etc.

Fases:

Fase 1. Modelado Conceptual: Se representa la estructura estática del sistema mediante un lenguaje de modelado.

Fase 2. Modelado de Hipertexto: Se describen los hipertextos que pueden ser publicados en el sitio Web. Cada uno de ellos define una vista del sitio. La descripción de los hipertextos se realiza mediante dos modelos: el modelo de composición, que define las páginas que componen el sistema, y el modelo de navegación, que describe cómo se podrá navegar a través de ellas.

Fase 3. Modelado de presentación: Define la apariencia física de las páginas.

Fase 4. Modelado de personalización: Finalmente, se define como debe adaptarse el sistema a los diferentes roles de usuario.

Uno de los aportes más interesantes de WebML es que ofrece una herramienta CASE que permite aplicar las técnicas propuestas y conseguir los resultados [WebRatio 2004]

2.1.12. UML Based Web Engineering (UWE)

La Ingeniería Web basada en UML (UWE) (Koch 2001. En: Escalona Cuaresma, M.J., 2004) es un proceso del desarrollo para aplicaciones Web enfocado sobre el diseño sistemático, la personalización y la generación semiautomática de escenarios que guíen el proceso de desarrollo de una aplicación Web.

UWE describe una metodología de diseño sistemática, basada en las técnicas de UML, la notación de UML y los mecanismos de extensión de UML. La propuesta de Ingeniería Web basada en UML, es una metodología detallada para el proceso de autoría de aplicaciones con una definición exhaustiva del proceso de diseño que debe ser utilizado. Este proceso, iterativo e incremental, incluye flujos de trabajo y puntos de control, y sus fases coinciden con las propuestas en el Proceso Unificado de Modelado.

UWE cubre todo el ciclo de vida de este tipo de aplicaciones centrandose además su atención en aplicaciones personalizadas o adaptativas. Esta metodología está especializada en la especificación de aplicaciones adaptativas, y por tanto hace especial hincapié en características de personalización, como es la definición de un modelo de usuario o una etapa de definición de características adaptativas de la navegación en función de las preferencias, conocimiento o tareas de usuario.

Para satisfacer los aspectos especiales de las aplicaciones Web, UWE define vistas especiales representadas gráficamente por diagramas en UML, tales como el modelo de navegación y el modelo de presentación. Además, UWE permite que un diseñador Web pueda también hacer uso de otra técnica de modelado UML que agreguen otras vistas de la aplicación, en otras palabras, UWE no limita el número

de vistas posibles de una aplicación. Las técnicas de modelado en UML abarcan la construcción de vistas estáticas y dinámicas de los sistemas de software: diagramas del objeto y de clase, diagramas de componentes, diagramas de casos de uso, diagramas de estado y de actividades, de secuencia y diagramas de la colaboración.

Además de esta colección de diagramas, el UML proporciona mecanismos de extensión basados en estereotipos. Estos mecanismos de extensión son los que UWE utiliza para definir estereotipos que son lo que finalmente se utilizarán en las vistas especiales para el modelado de aplicaciones Web. De esta manera, se obtiene una notación UML adecuada a un dominio en específico a la cual se le conoce como Perfil UML.

Durante el proceso de desarrollo de la UWE las fases de la gestión de proyectos y de la gestión de calidad son fundamentales, pues son ejecutadas para analizar los riesgos inherentes a su proceso de desarrollo. La gestión de proyectos logra el feedback de todos los pasos ejecutados durante el proyecto y les ajusta a los objetivos exigidos para realizar los próximos pasos del proyecto. Por su vez, la gestión de calidad da una atención especial a lo apresamiento de los requisitos, validación, verificación, *tests* y mantenimiento.

Los modelos de UWE están estructurados en paquetes. Cada paquete contiene un conjunto de clases y enlaces entre las clases. El modelo de casos de uso es el único usado para requisitos (CIM) en UWE. Los demás son los modelos de análisis-diseño.

Esta metodología distingue entre la tarea de elicitar requisitos, definir y validar los requisitos. El resultado final de la captura de requisitos en UWE es un modelo de casos de uso acompañado de documentación que describe los usuarios del sistema, las reglas de adaptación, los casos de uso y la interfaz. UWE clasifica los requisitos en dos grandes grupos: funcionales y no funcionales.

El modelo de navegación se construye en dos fases. En una primera etapa se desarrolla un modelo de espacio de la navegación, construido como vista del modelo conceptual y que indica cuáles son las clases y modelos visitables. Se representa

mediante un modelo de clases especiales denominadas clases navegacionales que no son más que clases de UML estereotipadas para indicar su semántica.

Fases:

Fase 1. Requisitos: propone comenzar con la identificación de los usuarios y la elicitación de los requisitos. Trata de diferente forma las necesidades de información, las necesidades de navegación, las necesidades de adaptación y las de interfaz de usuario, así como algunos requisitos adicionales relacionados, por ejemplo, con las restricciones hardware o la seguridad. Tras esto, centra el trabajo en el estudio de los casos de uso, la generación de los glosarios y el prototipado de la interfaz de usuario.

Fase 2. Análisis y diseño es similar a la de OOHDM. Sin embargo, UWE engloba más aspectos que OOHDM. De hecho, UWE distingue entre diseño conceptual, de modelo de usuario, de navegación, de presentación, de adaptación, de la arquitectura, en el diseño detallado de las clases y en la definición de los subsistemas e interfaces.

Fase 3. Implementación, UWE incluye todas las tareas que llevan a la implementación de los modelos aceptados: implementación de la arquitectura, implementación de la estructura del hiperespacio, implementación del modelo de usuario, implementación de la interfaz de usuario, implementación de los mecanismos adaptativos y las tareas referentes a la integración de todas estas implementaciones.

2.1.13 W2000

La propuesta de W2000 (Baresi et al. 2001. En: Escalona Cuaresma, M.J., 2004) ha sido la evolución a la orientación a objetos de su antecesora HDM. Con respecto a HDM tiene dos diferencias básicas. La primera de ellas es que HDM no era realmente una propuesta metodológica sino un modelo enriquecido del diagrama Entidad-Relación, W2000 sí que propone un ciclo de vida para el desarrollo de sistemas Web.

Una característica más importante de W2000 es que separa el aspecto de la navegación y la estructura de la información desde las primeras fases del ciclo de vida.

Esta metodología supone una propuesta que amplía la notación de UML con conceptos para modelar elementos de multimedia heredados de la propuesta HDM.

Fases:

Fase 1. Análisis de requisitos: El proceso comienza con esta fase, basado principalmente en los casos de uso. La especificación de requisitos en W2000 se divide en dos subactividades: análisis de requisitos funcionales y análisis de requisitos navegacionales. La especificación de requisitos comienza haciendo un estudio de los diferentes roles de usuario que van a interactuar con el sistema. Cada actor potencialmente distinto tendrá su modelo de requisitos de navegación y de requisitos funcionales. El modelo de requisitos funcionales es representado como un modelo de casos de uso tal y como se propone en UML. En él se representa la funcionalidad principal asociada a cada rol y las interacciones que se producen entre el sistema y cada rol. El segundo modelo consiste en otro diagrama de casos de uso pero que no representa funcionalidad sino posibilidades de navegación de cada actor. La representación gráfica es realizada con una extensión de UML.

Fase 2. Diseño de Hipermedia: Con el conocimiento adquirido durante la fase de requisitos, se pasa a esta fase. En ella, se realizan dos modelos: el *modelo conceptual* y el *modelo navegacional*. Para ello, los autores han modificado y extendido algunos modelos de UML como el diagrama de clases o el diagrama de estados.

Fase 3. Diseño funcional. Por último, se pasa a esta fase, en la que se ha adaptado el diagrama de secuencias para expresar la funcionalidad del sistema.

2.1.14 Proyecto UWA- Ubiquitous Web Applications

El proyecto UWA (UWA 2001. En: Escalona Cuaresma, M.J., 2004) es un proyecto en el que participan diversos grupos de investigación que desde hace años han trabajado en el mundo de la ingeniería Web.

De hecho, muchas de las propuestas que se hacen en su ciclo de vida y de las técnicas y modelos que utilizan han sido heredadas de otras propuestas como W2000. UWA ha nacido de la colaboración entre diferentes grupos de trabajo, por lo que resulta realmente una agrupación de propuestas y técnicas. En concreto, la propuesta de W2000 se encuentra incluida en UWA. Sin embargo, W2000 ha sido incluida en UWA sólo en la fase de diseño hipermedia, siendo ambas propuestas diferentes en la fase de definición de requisitos.

Los requisitos son clasificados en varios tipos: de contenido, de estructura de contenido, de acceso, de navegación, de presentación, de operaciones de usuario y de operaciones del sistema.

Para definir los objetivos, UWA propone una notación propia, basada en una plantilla. La definición de los actores y la relación con los objetivos se hace usando un diagrama basado en casos de uso. Por último, para definir y refinar los subobjetivos y los requisitos, utilizan una notación gráfica propia que denominan grafo de refinamiento de objetivos, el refinamiento de este grafo permite ir representando la relación entre los requisitos y hacer un seguimiento para validar la consecución de los objetivos del sistema. Una vez que los requisitos son detectados, hacen uso de XML para definirlos de una manera formal.

Fases:

Fase 1. La elicitación de requisitos: En esta fase se decide qué debe hacer la aplicación. Es una elicitación de requisitos basada en objetivos. El proceso comienza marcando los objetivos a alto nivel y a partir de ellos, se van concretando las necesidades que acaban definiendo los requisitos del sistema.

Fase 2. Diseño hipermedia: Modela los datos, la navegación y presentación de los mismos y los servicios que se ofrecen al usuario. Esta fase viene heredada de W2000. De hecho propone tres modelos para modelar la hipermedia: el modelo de información, el modelo de navegación y el modelo de presentación.

Fase 3. Diseño de transacciones: En esta fase se modelan las transacciones desde dos puntos de vista: el punto de vista estático, que estudia qué transacciones

se pueden producir, y el punto de vista dinámico, que estudia las consecuencias que cada transacción puede producir en el sistema.

Fase 4. El diseño de la personalización: el diseño de la personalización en el proyecto UWA se hace mediante componentes.

2.1.15. Object-Oriented Hypermedia Method (OOH)

OOH (Cachero 2003. En: Escalona Cuaresma, M.J., 2004) nace con la orientación de, utilizando una aproximación orientada a objetos, capturar la semántica necesaria para el modelado eficiente de interfaces de usuario y su implantación en web. La idea de OOH es ofrecer un marco metodológico que permita modelar de manera sistemática interfaces adaptativas y adecuadas al conjunto de usuarios del sistema.

Características de la Metodología:

1. OOH se caracteriza por diseñar interfaces en las que se implementa la navegación del sistema adaptado a cada rol de usuario a las necesidades que cada uno de ellos plantea.
2. Otra de las características más importantes de OOH es que lleva asociada una herramienta que soporta su ciclo de vida. Toda la información de VisualWADE está disponible en (VisualWADE 2004).

Fases:

Su ciclo de vida comienza con un **análisis de requisitos** que da paso a la **fase de ingeniería**. En esta fase de ingeniería se realiza el análisis y el diseño del dominio y de la navegación, así como el diseño de la presentación. Tras esto se pasa a una **fase de construcción y adaptación** en la que se obtiene, en base a un conjunto de plantillas, el sistema final. La última fase incluye la evaluación de la interfaz por parte del cliente. Todas estas fases se centran mucho en los roles de usuario.

2.2 Metodología propuesta para el desarrollo de aplicaciones Hipermedia y Web: Navigational Development Technique¹ (NDT)

En este acápite se presenta a NDT (En: Escalona Cuaresma, M.J., 2004) como una propuesta adecuada para el desarrollo de aplicaciones Hipermedia y Web porque permite modelar los requisitos navegacionales, sin que ofrezca nuevos lenguajes de modelado, define un proceso de desarrollo con actividades y tareas específicas, usa lenguajes de modelado estándar, como en el caso de los diagramas de clase para el modelo conceptual, o basados en estándares, como el caso del diagrama de clases navegacionales adquirido de UWE, permite que, con los resultados de NDT se pueda continuar el ciclo de vida con cualquier otra metodología.

Estos modelos al ser estándar pueden ser asumidos en cualquier propuesta basada en este tipo de estándares. Por último, decir que la aportación más importante de NDT es que ofrece una guía sistemática para el tratamiento de la navegación. Este tratamiento sistemático viene derivado de la definición formal de los modelos de ingeniería de requisitos y análisis así como de las relaciones que se establecen entre ellos. Los aspectos más importantes de NDT se resumen en las siguientes viñetas:

- Creado M.J. Escalona en 2004 para su tesis doctoral.
- Metodología para obtener y modelar requisitos para sistemas navegacionales: técnicas y notación
- Herramienta CASE: NDT-Tool
- Captura de requisitos mediante plantillas
- NDT-Tool produce prototipos visuales
- Basado en ideas de :
 - OOHDM: tratar modelo conceptual y modelo navegacional aparte
 - UWE: metamodelo para navegación, usar UML
 - REM: requisitos de almacenamiento, funcionales, etc.

¹ Técnicas de desarrollo navegacional, se toma el nombre en ingles pues es el propuesto por el autor.

Características de la metodología:

Orientada al Proceso: NDT describe de manera detallada todos los pasos que hay que realizar para tratar los requisitos y, a partir de ellos, conseguir los modelos de análisis.

Orientada a la Técnica: en todo el proceso NDT se indica qué técnicas se deben usar, el modelo de aplicación y el resultado que hay que obtener.

Orientada al Resultado: tras la aplicación de las técnicas definidas por NDT se obtienen resultados y modelos cuya nomenclatura y estructura están completamente detalladas.

Fases:

Fase 1. Ingeniería de Requisitos

NDT es una propuesta que da mucha importancia a esta etapa. El proceso de desarrollo que propone NDT se podría catalogar como un proceso bottom-up, en el sentido de que defiende el realizar un gran esfuerzo en la tarea de captura, definición y tratamiento de requisitos, con idea de sentar una base adecuada y correcta, con el menor número de errores posibles.

El proceso comienza con una primera actividad en la que se hace un estudio del entorno del futuro sistema, se preparan y realizan las entrevistas, determinando cuáles son los objetivos del mismo. Tras esta primera actividad, se suceden un conjunto de actividades en las que se captura y definen los diferentes requisitos del sistema.

NDT contempla 5 tipos de requisitos:

- Requisitos de almacenamiento de información
- Requisitos de actores.
- Requisitos funcionales
- Requisitos de interacción
- Requisitos no funcionales

Tras el estudio de los requisitos, NDT propone realizar una validación de los mismos.

Si al validar los requisitos no se detectan errores o incongruencias, se generará el documento de requisitos del sistema (DRS), que es el resultado de la fase y que, además de ser el producto de la misma, es la base para la realización del análisis.

En la tabla 2.1 se presentan todas las actividades y las tareas que componen la fase de ingeniería de requisitos. Estas se describen en el ejemplo realizado de esta propuesta en el capítulo 3. Como resultado de esta fase se obtiene el documento de requisitos del sistema.

Fase 2. Análisis

El proceso de análisis en NDT tiene como objetivo obtener los modelos de análisis de un sistema Web. NDT asume que para realizar el análisis de un sistema web es necesario realizar tres modelos:

- Modelo conceptual, que representa la estructura estática del sistema,
- Modelo de navegación, que representa la estructura de navegación del sistema
- Modelo de prototipos del sistema

El modelo conceptual se compone de un único diagrama. Sin embargo, el modelo de navegación y los prototipos puede ser muy diferente dependiendo del rol del usuario que en cada momento interactúe con el sistema. Por ello, estos modelos se van a adaptar dependiendo del actor en estudio. Un actor en estudio es un rol, o un conjunto de roles, de usuario que comparten la misma navegación.

La realización de estos modelos se hace en NDT de una manera sistemática desde los requisitos. Partiendo de los resultados en la fase anterior, NDT propone una serie de procesos que permiten generar estos modelos. Sin embargo, los modelos generados desde los requisitos deben ser revisados para detectar posibles errores o incongruencias en fases anteriores, o para adecuarlos mejor a la realidad del sistema. Los cambios y modificaciones posibles son controlados por NDT.

El análisis se compone de cuatro fases. La primera de ellas permite obtener el modelo conceptual, la segunda el modelo de navegación, la tercera los prototipos y, en la última, se genera el documento de análisis del sistema, resultado de esta fase.

Las tres primeras actividades, a su vez, se dividen en dos tareas. En las primeras, se consiguen los modelos básicos, de manera sistemática desde los requisitos. En las segundas, el grupo de analista puede aplicar su experiencia para mejorar los resultados, pero los cambios y modificaciones que pueden realizar son controlados por NDT. Nótese que desde estas segundas tareas se puede volver a actividades anteriores para resolver problemas o incongruencias.

En la tabla 2.1 se presentan todas las actividades y las tareas que componen la fase de análisis. Estas se describen en el ejemplo realizado de esta propuesta en el capítulo 3. Como resultado de esta fase se obtiene el documento de análisis del sistema.

Tabla 2.1. Fases, actividades y tareas de NDT.

Fases	Actividades	Tareas
Ingeniería de requisitos	Obtener información sobre el entorno y definir objetivos	Obtener información sobre el dominio de problema
		Preparar y realizar reuniones y entrevistas
		Identificar y definir los objetivos
	Identificar y definir los requisitos de almacenamiento de información	Identificar y definir los requisitos de almacenamiento de información
		Identificar y definir las nuevas naturalezas
	Identificar y definir los actores	Identificar y definir los actores básicos
		Identificar y definir la generalización de actores
		Identificar y definir la incompatibilidad de actores
		Identificar y definir los actores derivados
Identificar y definir los requisitos funcionales	Diseñar los diagramas de casos de uso	
	Describir los casos de uso	
Identificar y definir los requisitos de interacción	Identificar y definir las frases	
	Identificar y definir los prototipos de visualización	
Identificar y definir los requisitos no funcionales	Identificar y definir los requisitos no funcionales	
Validar los requisitos	Validar los requisitos	
Generar el documento de requisitos del sistema	Generar el documento de requisitos del sistema	
Análisis	Realizar el modelo conceptual	Realizar el modelo conceptual básico
		Realizar el modelo conceptual final
	Realizar el modelo de navegación	Definir los actores en estudio
		Realizar el modelo de navegación básico
		Realizar el modelo de navegación final
	Realizar y validar los prototipos	Realizar los prototipos
		Validar los prototipos
Generar el documento de análisis del sistema	Generar el documento de análisis del sistema	

Resultados de la aplicación de NDT

Los resultados de la aplicación de NDT son:

1. Requisitos del sistema (de almacenamiento de información, de actores, funcionales, de interacción, no funcionales)
2. Análisis del sistema (modelo conceptual, modelo de navegación)
3. Prototipos del sistema

Es importante decir que esta metodología no incluye las fases de diseño e implementación, pero con los resultados de ella, se dispone de toda la información necesaria para el desarrollo de las fases de diseño e implementación.

Para una mejor comprensión de esta metodología en el Capítulo 3 se describe la aplicación de la metodología NDT mediante un ejemplo real.

2.3. Metodologías para el desarrollo de Aplicaciones Multimedia

A finales de los 80's y principios de los 90's el término Multimedia comienza a popularizarse, como significado de la integración de elementos (textos, imágenes estáticas y animadas, sonido) a las tecnologías de computación.

Muchos desarrolladores de software Multimedia abogan por el desarrollo de principios y métodos de ingeniería de software para la construcción de sistemas Multimedia. Al mismo tiempo como profundización de estos anhelos, forma parte de la demanda de los creadores de este tipo de aplicación, el desarrollo de notaciones precisas semánticamente, y al mismo tiempo usables sintácticamente, que soporten las diferentes vistas y niveles de abstracción.

Debido al estudio realizado de las metodologías de desarrollo se definieron las siguientes, las cuales le facilitará el trabajo al que la utilice.

1. Hypermedia Design Model(HDM)
2. Relationship Management Method (RMM)
3. Enhanced Object Relationship Methodology (EORM)
4. Object-Oriented Hypermedia Design Method(OOHDM)
5. Scenario-based Object-oriented Hypermedia Design Methodology(SOHDM)

6. Hypermedia Flexible Process Modelling Strategy (HFPM)
7. Metodología para elaborar multimedia a nivel sistémico (MEMS)
8. OMMMA
9. RUP (OMMMA-L)
10. MULTIMET

En el capítulo 2 "Desarrollo de hipermedia y Web como proceso de ingeniería", se hace un resumen de las metodologías: HDM, RRM, EORM, OOHDM, SOHDM, HFPM las que son utilizadas además para desarrollo de aplicaciones Multimedia.

2.3.1 Metodología para elaborar Multimedia a nivel sistémico (MEMS)

Multimedia en el concepto de sistema en MEMS

Cuando MEMS (Muñoz Henríquez, L.M., 2000) está destinada a la educación, existe un énfasis en tres aspectos fundamentales. El concepto de sistema, el concepto de comunicación y el concepto de evaluación. La educación es una forma de comunicación, y de esta manera está considerada en MEMS, como también está considerado en ella incluir los valores como parte ineludible del aprendizaje.

En MEMS destinada a la educación, se cree que un soporte Multimedia siempre deberá constituir un sistema en tanto está dirigido a potenciar el aprendizaje. Por ello, debería incluir necesariamente, aquel material de complemento que le asegure su efectividad. Por tanto, en su desarrollo MEMS considera tres tipos de soportes electrónicos o digitales (Muñoz Henríquez, L.M., 2000):

1. Un sistema Multimedia educativo (SME) equivalente a una aplicación o documento off-line en CD-ROM, DVD u otro; además de, como mínimo, una guía de enseñanza - aprendizaje.
2. Un sistema Hipermedia educativo (SHE) equivalente un documento on-line en página o sitio Web que, de igual modo debe incluir, como mínimo, una guía de complementos.
3. Un sistema modular Multimedia educativo (SMME) equivalente a una estructura de enseñanza-aprendizaje on-line que puede consistir en un módulo de aprendizaje o un sistema modular. Un módulo puede estar compuesto por más de

un soporte en interconexión.

Fases:

Fase 1: ORGANIZACIÓN: En un proyecto Multimedia es imprescindible una buena organización. Concluir esta etapa, en MEMS significa haber detectado la necesidad Identificando y verificando una serie de aspectos.

Fase 2. PLANIFICACION: Planificar en MEMS, significa profundizar en detalles, en cálculos precisos sobre la base de la organización hecha con antelación. Significa analizar y estructurar el problema ya detectado para establecer finalmente un orden y control de la situación.

Fase 3. DISEÑO: Cuando se desarrolla Multimedia educativo, diseño es un concepto muy amplio en MEMS. Tiene varias acepciones: Diseño como creación de un lenguaje Audiovisual-Cinético; como creación de un lenguaje sistémico y como creación de un lenguaje pedagógico, y muchas de sus fases corresponden, necesariamente, a actividades paralelas. Todo esto es Diseño en MEMS.

Fase 4. DESARROLLO: es el complemento de las etapas anteriores. Con la recopilación de contenidos, la digitalización y la edición se posibilita el avance hacia la etapa de producción, y funciona totalmente en forma paralela a las etapas que le anteceden. Tanto la recopilación de contenidos como la digitalización son actividades similares en cualquier método de desarrollo multimedia por lo que no nos referiremos a ellas en esta oportunidad.

Fase 5. PRODUCCION: La Producción en MEMS consiste en reunir todo el trabajo en un todo integrado y cohesionado, llamado Prototipo.

Producir requiere tres fases: programar, integrar los componentes y, obtener el 1er. prototipo.

Fase 6. VALIDACION: En MEMS la verificación de efectividad de un SME, SHE o SMME es una fase imprescindible para el logro de las metas propuestas en la solución de la problemática planteada. En esta etapa, se considera la valoración de expertos especialistas en el tema y también, la de quienes se relacionan con la

educación. Por tanto, MEMS, en la validación siempre considerará necesarios como expertos, además del especialista, a un comunicador audiovisual, un pedagogo o diseñador instructivo, un psicólogo, un ingeniero o técnico en Multimedia, y un sociólogo cuando se trate de un soporte para un ambiente de aprendizaje masivo. Todos ellos, sin embargo, han de ser profesionales ajenos al equipo de trabajo.

Fase 7. EVALUACION: La etapa de evaluación se relaciona sólo con el educando y comprende tres fases: La evaluación de los conceptos en el educando para establecer el mapa conceptual, la etapa de diseño como lenguaje audiovisual-cinético, con el pre-test de transparencia aplicado al educando y la evaluación del 2º prototipo. Estas fases se hacen en Red para hacer observaciones de campo más reales.

2.3.2. RUP con OMMMA-L

Para el desarrollo de aplicaciones Multimedia (Sonlezal Fernández, G y Díaz Catalá,S. (2006)) se puede usar como metodología RUP, utilizando el lenguaje de modelado orientado a objetos de aplicaciones Multimedia (OMMMA-L), que es una propuesta para la integración de especificaciones de sistemas Multimedia basados en el paradigma orientado a objetos.

OMMMA-L se sustenta en cuatro vistas fundamentales, a cada una se asocia un tipo de diagrama en particular. Estas vistas son: la lógica, la de presentación espacial, la de comportamiento temporal y la de interacción.

OMMMA – L, no es un lenguaje nuevo, sino una extensión de UML, por lo que no es necesario aprenderlo, sino interpretar las características extendidas, centrados a la lógica de funcionamiento de una multimedia. Muestra análisis similares a otras metodologías y no se especializa en una clasificación de producto, sino que se generaliza a través del uso de la semántica original de UML.

2.3.3 OMMMA

Los desarrolladores de aplicaciones Multimedia tienen que considerar un rango extenso de aspectos cuando desarrollan un sistema de información multimedia. No solamente tienen que tener en cuenta la estructura de la aplicación y su

funcionalidad sino también la interfaz de usuario y los aspectos del sistema tales como la distribución, control de flujo(*stream control*), gestión de recursos o bases de datos y recursos de la red (Kirschner, J., 2003).

Aspectos esenciales de las aplicaciones multimedia para la modelación.

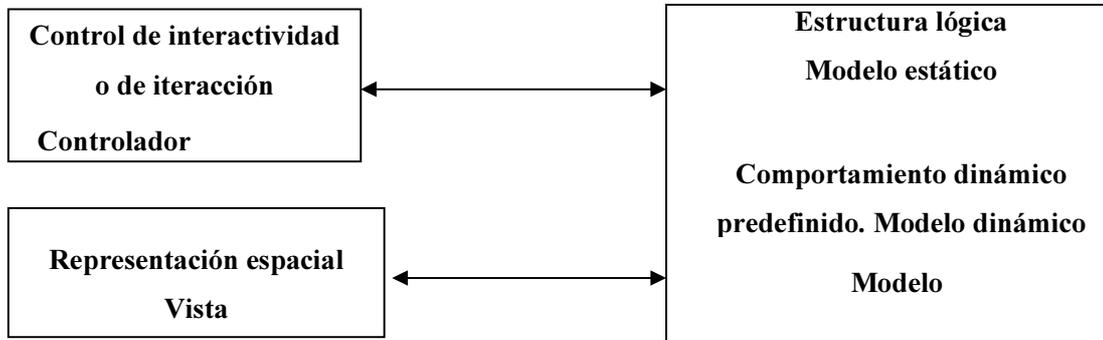
Una característica fundamental de las aplicaciones multimedia es la composición de diferentes aspectos. Además de las características estáticas del contenido (el concepto clásico de modelación de datos para un sistema de bases de datos), los tipos de medios utilizados, el comportamiento predefinido e interactivamente controlado y también el diseño de interfaz de usuario deben de ser considerado. Además de estos aspectos, el vínculo con un sistema de base de datos o el acceso a recursos distribuidos dentro de un sistema de red son de vital importancia para los sistemas de información Multimedia. Todos estos aspectos deben de ser especificados dentro de un modelo integral y consistente durante el desarrollo de una aplicación Multimedia. De ahí podemos identificar los aspectos esenciales de una aplicación Multimedia a modelar:

1. Estructura lógica
2. Tipos de medios asociados y los objetos del dominio de aplicación
3. Comportamiento temporal predefinido
4. Disposición espacial de los elementos de la presentación
5. Interactividad o comportamiento dinámico guiado por eventos

Esta clasificación lleva a una arquitectura específica de una aplicación Multimedia que será presentado como una extensión del paradigma estándar de modelo vista controlador (MVC_{mm}).

Se propone la utilización de la arquitectura MVC (patrón de diseño) que distingue el comportamiento y los datos, los componentes visuales para mostrar información a usuarios y componentes controladores para manejar la entrada de los usuarios. Estos elementos han sido enriquecidos para el modelado de multimedia separando el modelo estático de la aplicación y el modelo dinámico.

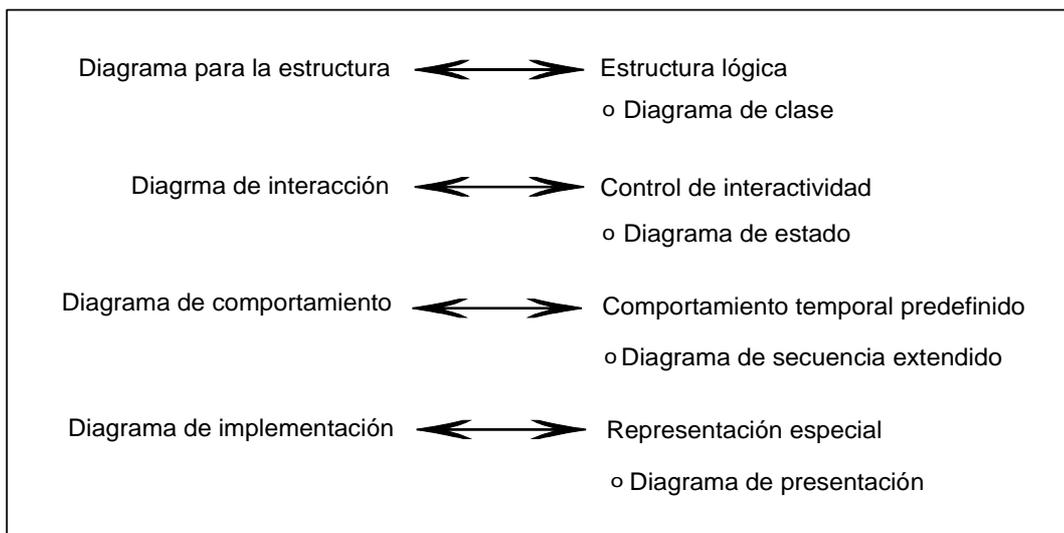
Este patrón es utilizado en la fase de análisis, lo que resulta una representación de las propiedades distintivas de cada componente.



La metodología OMMMA es una propuesta para el desarrollo de aplicaciones multimedia dentro del paradigma orientado a objeto, cubre las fases de especificación de requisitos, análisis, diseño e implementación. Está soportada por una herramienta case llamada OMMMA-TOOL.

Los modelos de análisis y diseño se realizan utilizando el lenguaje OMMMA-L.

OMMMA-L despliega diagrama de clases y de estado de UML tanto como un diagrama de secuencia extendido y un diagrama de presentación recientemente introducido para especificar adecuadamente la presentación visual dentro de un sistema de información Multimedia.



OMMMA-L se encuentra sustentado en cuatro vistas fundamentales, donde cada una se asocia a un tipo de diagrama en particular (disponible en: <http://www.ilustrados.com/publicaciones/EEAkulVyEVyKTwmnEi.php>)

- **-Vista Lógica:** modelada a través del Diagrama de Clases de OMMMA-L, extendido del Diagrama de Clases de UML, utilizando las mismas notaciones, pero incorporando las clases correspondientes a las medias: media continua y media discreta, generalizadas en una clase medias. Divide en dos áreas dicho diagrama: una para la jerarquía de los tipos de media y otra para la modelación de la estructura lógica del dominio de la aplicación.
- **-Vista de Presentación espacial:** modelada a través de los Diagramas de Presentación de OMMMA-L, los cuales son de nueva aparición en la extensión de UML, dado que este último no contiene un diagrama apropiado para esta tarea. Estos diagramas tienen el propósito de declarar las interfaces de usuario con un conjunto de estructuras delimitadas en tamaño y área, dividiéndose en objetos de visualización (texto, gráfico, video, animación) e interacción (barras de menú, botones, campos de entrada y salida, scrolls, hipertextos con hipervínculos).
- **Vista de Comportamiento temporal predefinido:** modelada por el Diagrama de Secuencia de OMMMA-L, extendido a partir del diagrama de secuencia de UML. El Diagrama de secuencia modela una secuencia de una presentación predefinida dentro de una escena, donde todos los objetos dentro de un diagrama se relacionan al mismo eje del tiempo. En este diagrama se hace un refinamiento del eje del tiempo con la introducción de marcas de tiempo a través de diferentes tipos de intervalos; marcas de inicio y fin de ejecución que permite soportar su reusabilidad; marcas de activación y desactivación de demoras en objetos de tipo media, posibilitando la modelación de las tolerancias de la variación de las restricciones de sincronización para los objetos media; activación compuesta de objetos media para la agrupación de objetos concurrentemente activos.
- **Vista de Control Interactivo:** modelado a través del Diagrama de Estado, extendido a partir del diagrama de estado de UML, sintácticamente igual a este último, mas con la diferencia semántica de que en el orden de unir los controles

interactivos y predefinidos, no interrumpidos de los objetos, las acciones internas de estados simples tienen que llevar nombres de diagrama de secuencia en vez de diagramas de estado empotrados; queriendo esto decir que el comportamiento especificado por el diagrama de secuencia se provoca automáticamente cuando se entra al estado correspondiente donde se hace referencia.

2.4. Metodología Multimet

2.4.1 Introducción

La metodología Multimet (Colectivo de autores, 2000) se emplea para el desarrollo de aplicaciones Multimedia. Ella cubre todas las etapas para la organización de un proyecto informático de este tipo. Las etapas están bien delimitadas y su objetivo es que cada especialista componente del equipo de desarrollo en cada proyecto, conozca la aplicación de forma integral y pueda dirigir su trabajo hacia un fin común.

2.4.2 Fases

Las etapas y fases propuestas en la metodología son:

1. Estudio preliminar:
 - 1.1. Definición del producto.
 - 1.2. Elaboración del plan de desarrollo.
 - 1.3. Estudio de factibilidad.
2. Definición del contenido de la aplicación:
 - 2.1. Definición de los objetivos.
 - 2.2. Identificación de la audiencia.
 - 2.3. Especificación del contenido.
 - 2.4. Definición de los medios y sus objetivos.
 - 2.5. Establecimiento de normas de diseño.
3. Especificación del contenido de la aplicación:
 - 3.1. Recopilación y preparación de los medios.
 - 3.2. Elaboración del diagrama de flujo.
 - 3.3. Confección del Guión.
4. Desarrollo de la aplicación:

- 4.1. Comprobación del Diagrama de flujo y acciones de acuerdo al guión.
- 4.2. Selección del Lenguaje de programación o Sistema de Autor.
- 4.3. Integración del contenido y los medios en su forma final.
5. Prueba de la aplicación:
 - 5.1. Elaboración de protocolo de prueba.
 - 5.2. Revisión y comprobación por el usuario.
6. Preparación para la distribución:
 - 6.1. Determinación de la forma de distribución.
 - 6.2. Diseño de empaquetadura.
 - 6.3. Preparación para su producción.
 - 6.4. Elaboración de documentos comerciales.

Para cada etapa están definidos los pasos a seguir y cada una debe desarrollarse teniendo en cuenta el refinamiento a etapas anteriores, por tanto el trabajo del productor debe tener en cuenta todos los elementos posibles para que las diferentes partes puedan trabajar por separado y a su vez en paralelo con los otros, disponiendo de los medios necesarios en cada momento, esto hace necesario elaborar un calendario de trabajo exacto que debe cumplir cada especialista y revisar cada paso antes de pasar al próximo.

A continuación se realiza un comentario sobre cada una de ellas.

1. Estudio preliminar

1.1. Definición del producto

En este punto deben quedar definidos algunos elementos básicos relacionados con las necesidades de los usuarios, elementos necesarios para el desarrollo y para la ejecución del producto; puede contemplar los siguientes puntos:

- Necesidades del usuario y objetivos del producto: ¿Por qué surge el producto y en función de esto que problemas debe resolver?.
- Tecnología necesaria para el desarrollo del producto: Que equipamiento se requiere para el desarrollo, tener en cuenta las necesidades para la preparación de cada uno de los medios que pudieran estar presentes en la aplicación.

- Tecnología necesaria para la ejecución del producto: De que medios debe disponer el usuario final para poder ejecutar satisfactoriamente la aplicación.
- Personal que trabajará en el desarrollo del producto: Esto dependerá de la cantidad de medios a incluir y de la complejidad de la aplicación, se recomienda que estén presentes:
 - Productor: Controla todo el desarrollo del diseño y de la aplicación, máximo responsable.
 - Experto en el contenido
 - Diseñador de la interface gráfica
 - Diseñador de animaciones
 - Técnico de audio
 - Técnico de vídeo
 - Programador
 - Especialista en didáctica

Si el proceso de edición de textos es largo y complejo puede ser necesario alguien que apoye el trabajo de edición y corrección y contar con un personal para la realización de las pruebas, que no haya tenido parte en el desarrollo del producto.

Si se trata de un producto para la venta debe estudiarse el mercado potencial y la estrategia de comercialización.

1.2. Elaboración del plan de desarrollo

Se confecciona un plan que incluya todas las etapas del desarrollo con fecha de inicio y de terminación y responsable. Puede ser que hasta este momento no se tengan todos los elementos para definir una etapa en particular, en este caso, se va completando a medida que se avanza.

En este momento se debe precisar el personal necesario para llevar a cabo el proceso con el cual se cuenta, cual debe ser contratado y cuales servicios se solicitarán, es el momento de definir el grupo multidisciplinario que acometerá el trabajo.

1.3. Estudio de factibilidad

Para hacer el estudio de factibilidad debemos tener en cuenta dos elementos:

- La factibilidad económica.
- La factibilidad técnica

Factibilidad económica:

En este punto deben analizarse varios factores, uno de los más importantes es la relación costos - beneficios, el impacto del producto final, costo de los elementos que hacen falta para el desarrollo y crecimiento potencial en el mercado. [Alvarez,1995]

El análisis de costos - beneficios debe tener en cuenta los beneficios tanto económicos como sociales que tendrá el producto. Por supuesto, en software educativo prevalecerá el beneficio social aunque puede tratarse de recuperar la inversión en que se incurra si una comercialización del mismo lo aconseja.

En los costos se debe incluir la inversión en la adquisición de equipamiento que en el caso de las aplicaciones Multimedia suelen ser necesarios, además los costos de los servicios técnicos y de la mano de obra que sea necesario contratar. Además incluir los costos en la producción de software donde se tienen en cuenta la digitalización de imágenes, textos, videos, sonido, desarrollo de animaciones, etc.

Factibilidad técnica:

Lo más importante a considerar es:

- Si es posible disponer de todo el personal técnico.
- Si se dispone de la tecnología necesaria, tanto desde el punto de vista de hardware como de software.

Como resultado debe quedar claro si es factible o no desarrollar el producto y continuar con el resto de las etapas.

Los principales resultados que deben quedar esclarecidos en esta primera etapa son:

- Por qué surge el producto y su importancia.
- De qué se dispone para su desarrollo.

- Cual es el personal del equipo técnico.
- Que necesitan los usuarios finales para ejecutar la aplicación.
- De que tiempo se dispone para la realización del producto.
- Será comercializado.

2. Definición del contenido de la aplicación

2.1. Definición de los objetivos de la aplicación

En este caso se definen los objetivos desde el punto de vista de la aplicación propiamente dicho, teniendo en cuenta si es educativa, demostrativa, informativa, etc.

2.2. Identificación de la audiencia

Uno de los aspectos más importantes es la correcta identificación del usuario final del sistema que puede resolverse respondiendo a la pregunta, a quién va dirigida la aplicación?, se debe tener en cuenta que los criterios de diseño están en función de satisfacerlos y un correcto análisis en este aspecto permitirá el cumplimiento de los objetivos antes señalados y definir que contenido incluir y como hacerlo, para ello debe tenerse en cuenta un grupo de factores, a los cuales ya nos hemos referido, mencionemos entre otros:

- Habilidades en el uso de la computadora.
- Conocimiento del tema.
- Como utilizará la información que se presenta.
- Necesidad que tiene del producto.
- Frecuencia de consulta a la información.
- En que ambiente se ejecutará la aplicación.

Una audiencia heterogénea puede conducir a que la aplicación tenga que ser estructurada en varios niveles.

2.3. Especificación del contenido

Aquí deben destacarse los temas que serán tratados, en su orden de aparición y teniendo en cuenta para cada uno el nivel de detalle y la forma en que será

estructurado. Igualmente ya debe tenerse la estrategia pedagógica a seguir para presentar y que a partir de este momento tendrá que empezar a tenerse en cuenta en la presentación de la aplicación.

2.4. Definición de los medios y sus objetivos

Para cada tema o parte de él debe tratar de definirse con que medios va a ser representado, y para cada medio utilizado debe quedar claro con que objetivo aparecerá, todo esto determina la importancia que tienen su presencia en la aplicación. Los objetivos que pueden señalarse son: educar, informar, persuadir, entretener, complementar o una mezcla de ellos, también hace falta conocer la disponibilidad de cada medio y la fuente de obtención de cada uno. Para esto proponemos utilizar el formato de la tabla 2.2.

Tabla 2.2. Modelo para la definición de los medios y sus objetivos

Tema	Medio	Objetivo	Disponible	Fuente
	<ul style="list-style-type: none"> • • • 			

Se considera disponible cuando se cuenta con el texto, la imagen, el sonido, el vídeo, en cualquiera de los medios de almacenamiento comunes para cada uno de ellos, en caso de que deba crearse, no estará disponible y en ese caso la fuente será el medio para obtenerlo.

2.5. Establecer normas de diseño

Para cada medio debe quedar claro que forma tendrá dentro de la aplicación, esto es lo que garantiza la uniformidad. Estas características o parámetros son específicos para cada medio utilizado.

Textos:

- Porcentaje máximo de ocupación de pantallas
- Fuentes utilizadas para títulos
- Fuentes utilizadas para texto normal.

Imágenes: Los parámetros de cada una están muy ligados con los objetivos que tienen en la aplicación, pero deben quedar claras las normas generales sobre todo teniendo en cuenta el espacio en disco de que se dispone, deben fijarse:

- Tamaño máximo y mínimo.
- Profundidad del color
- Resolución de la imagen.

Sonido: Pueden ser utilizados diferentes tipos de sonido como: música de fondo, locución, efectos, cada uno debe ser tratado por sus características e importancia definiendo:

- Frecuencia de muestreo
- Precisión del valor de cada muestra.

Vídeo / Animación: Ocupan un importante volumen de disco, luego para la definición de sus características se tendrán en cuenta sus objetivos. No deben incluirse videos de muy larga duración porque esto influye en la capacidad y en la calidad. Los parámetros a fijar son:

- Duración
- Parámetros de cada imagen
- Parámetros del sonido
- Cantidad de cuadros por segundo.

Como resultado de esta etapa deben quedar perfectamente definidas:

- Características del usuario a quien va dirigida la aplicación
- Relación de temas que aparecerán reflejados con su estructuración
- Para cada tema, medios a emplear, objetivos, disponibilidad y fuentes
- Normas generales de diseño para cada medio.

3. Especificación del contenido de la aplicación

3.1 Recopilación y preparación de los medios

De acuerdo con las fuentes para obtener los medios definidos anteriormente, se procede a recopilar cada uno de ellos y luego a su preparación que en cada uno tendrá características especiales:

Textos: El texto puede ser almacenado en caracteres o como imagen, para definir esto debemos tener en cuenta las posibilidades de operación con la información y el espacio en disco. Las características generales de la aplicación son importantes.

Para almacenar un texto en caracteres, si la fuente de que se dispone es material escrito se deberá teclear o usar un programa en su digitalización y posteriormente realizar un exhaustivo proceso de revisión que implica un gran esfuerzo y personal, sin embargo en este caso los procesos de búsqueda pueden ser amplios y la información puede particionarse según las necesidades, para vincularse con otros elementos cuando lo demande la aplicación, el volumen de espacio que ocupan es pequeño.

Para almacenar un texto como una imagen solo se necesita digitalizar las imágenes y el proceso de revisión es más sencillo, sin embargo este texto solo puede mostrarlo, no puede realizar procesos de búsqueda y requiere mucho más espacio para ser almacenado [Perez,1995]. Con esta variante se respeta el diseño de documentos originales que hayan sido utilizados.

El volumen de información a incluir puede ser muy grande y si no es bien tratado desviarán la atención del usuario, el análisis del contenido de textos que aparecerán en cada pantalla debe ser bien estudiado, si tenemos en cuenta que en muchos casos resulta imposible sustituir los textos en las aplicaciones.

Se recomienda que para incluir la información en la aplicación tenga en cuenta:

- Separar subtemas de interés.
- Seleccionar los textos que deben aparecer en cada pantalla de acuerdo a las normas.

- Seleccionar los conceptos e ideas generales que pueden ser extraídos sin que se afecte la idea central y ser luego consultados como información adicional e incluso ser compartidos desde otras partes de la aplicación.
- Selección para cada pantalla de textos las palabras calientes que forman parte del Hipertexto.

Los textos que serán tratados como caracteres deberán ser siempre revisados y para esto utilizará un Editor de textos con facilidades, que permita obtener el texto con el formato de diseño definido y almacenar en ficheros con el formato compatible con el resto de la aplicación.

Imágenes: La preparación de las imágenes requiere un nivel de especialización, por los conocimientos de diseño que requiere y las facilidades en el manejo de herramientas especializadas. Si la fuente es papel u otro similar impreso, el primer paso será la digitalización de la imagen utilizando un Escáner y un software apropiado para esto. Después, y válido también para aquellas imágenes disponibles en disco, se debe utilizar un editor o procesador para su edición. El objetivo es obtener la máxima calidad ajustando parámetros y tamaño.

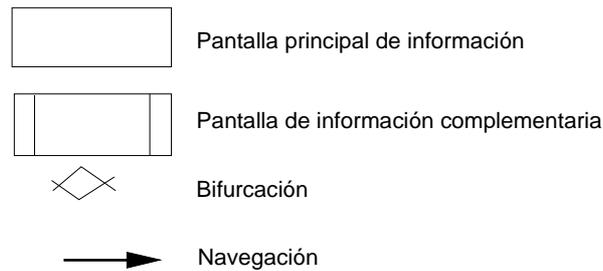
Sonido: El sonido puede ser recuperado de disco o estar almacenado en un medio externo a la computadora como cinta CD musical, una locución, etc. Si lo tenemos en uno de los medios externos debemos como primer paso digitalizarlos, siguiendo las normas de diseño establecidas y con el uso de software especializado para la digitalización. Posteriormente, y en esto se incluye el sonido que está en disco, se debe pasar al proceso de edición que permitirá ajustar el sonido según las necesidades, acotarlo, mezclarlo, limpiarlo de ruidos, etc. este es un trabajo que requiere de un especialista para su culminación exitosa.

Videos / Animación: La fuente de los videos generalmente es una cinta, por lo tanto deben digitalizarse, para ello debe tener una máquina con tarjeta digitalizadora de vídeo y un software apropiado para ello, posteriormente se pasa al proceso de edición. De igual forma para crear las animaciones deben tenerse en cuenta las

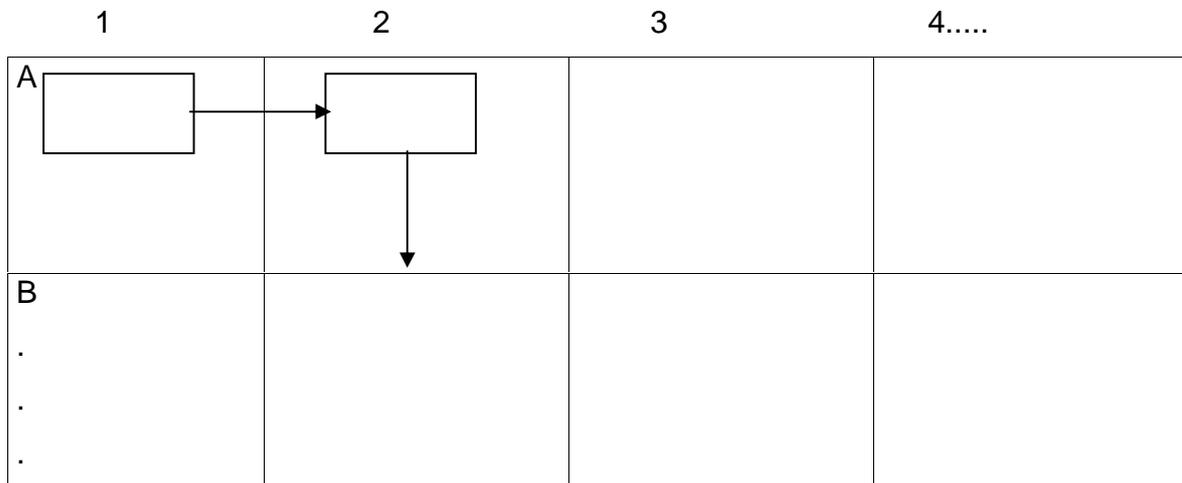
normas de diseño. El equipo que se dedique a la actividad de diseño y preparación de los medios debe tener gran capacidad de almacenamiento y procesamiento.

3.2. Elaboración del diagrama de flujo

Este diagrama se propone sea igual al de cualquier programa, los símbolos a utilizar son:



Nos parece acertada la idea mostrada por otros autores de que los diseñadores puedan definir otros símbolos siempre que sean bien explicados. Para facilitar la referencia posterior a cualquier elemento dentro del diagrama utilizaremos una hoja con cuadrícula de letras y números, donde se hace referencia a cada elemento señalando la fila y columna a la que pertenece, ejemplo A1. Cada elemento dentro del diagrama se corresponde con una pantalla o elemento de información.



El diagrama muestra la secuencia en que interactúa el sistema y las acciones posibles. Debe ser confeccionado desde la perspectiva del usuario y su interacción con el sistema. Cada línea hacia un elemento implica la posibilidad de retorno.

3.3. Confección del guión:

Con el diagrama se obtiene una idea del funcionamiento general e integral del sistema, sin embargo existen aspectos a tener en cuenta y que no se colocan en el diagrama como son:

- Tema tratado en cada elemento.
- Información que aparece en la pantalla.
- Acciones del usuario que determinan las respuestas del sistema.
- Respuesta del sistema a cada acción.
- Tratamiento de errores.

Para ello podemos establecer un guión, el cual puede ser elaborado a través de la tabla 2.3 para facilitar el trabajo:

Tabla 2.3. Modelo para la confección del guión

Tema	Posición Diagrama	Entradas	Alcance Información	Acciones usuarios	Respuesta sistema	Trat. Errores
	<ul style="list-style-type: none">••					

Para cada elemento del diagrama se deben indicar todas las acciones posibles del usuario, a través de la selección de botones, palabras de Hipertexto (subrayadas), presionando alguna tecla y para cada una que acción toma el sistema.

Al finalizar esta etapa debe quedar disponible:

- Biblioteca de medios asociados con los elementos del diagrama.
- Diagrama de Flujo de interacción.
- Guión

4. Desarrollo de la aplicación

Cuando se llega a este punto en el desarrollo ya está preparada toda la información a incluir y diseñado el funcionamiento integral del sistema desde el punto de vista de las acciones del usuario, queda lo relacionado con la integración de todos los medios a partir de una prueba exitosa del guión y el diagrama de flujo.

En este paso participan fundamentalmente los programadores aunque el resto del equipo debe estar disponible para posibles ajustes e imprevistos que surgieran.

4.1. Comprobación del Diagrama de flujo y acciones de acuerdo al guión

Debe existir total correspondencia entre lo que ilustra el diagrama de flujo y lo que aparece en el guión, esto debe ser revisado cuidadosamente teniendo en cuenta que la secuencia Acción - Respuesta par cada elemento del diagrama tenga sentido en todos los casos y esté correctamente expresado en el guión. Hay que verificar que todas las acciones imprevistas han sido tenidas en cuenta. Debe tenerse en cuenta el orden de aparición de los temas según lo expresado en el paso número 2 de la metodología.

4.2. Selección del lenguaje de programación o sistema de autor

Aquí se trata de seleccionar la herramienta de ensamblaje de la aplicación que debe cumplir algunos requisitos como son:

- Programación visual para garantizar eficiencia.
- Facilidades para la manipulación de recursos Multimedia.

Estas herramientas pueden ser:

- Lenguaje de programación.

Permiten la programación visual de aplicaciones, incidiéndose en el uso de bibliotecas de control de dispositivos. Son más flexibles y eficientes en la ejecución de las aplicaciones pero requieren de un mayor nivel de habilidad técnica y se consume más tiempo en el desarrollo.

- Sistema de autor.

Pueden ser de diferentes tipos:

- Basados en líneas de tiempo: El esquema de control es basado en un horizonte de tiempo, sobre el cual existen diferentes canales en los cuales se hace control de las intervenciones de cada uno de los dispositivos. Escaso control de las acciones del usuario. Ejemplo Mediablitz.
- Basados en diagramas de flujo: La lógica de la aplicación se estructura a manera de un diagrama de flujo lógico, en el cual se visualizan las trayectorias y los elementos que forman parte de ellas. Permiten el control directo de la interacción

con el usuario, identificando sus respuestas y definiendo líneas de acción ante cada una de ellas. Ejemplo Author Ware Star.

- Basados en páginas: La aplicación se estructura en páginas en las cuales existen tanto criterios de linealidad como rutas alternativas, definidas en función de acciones del usuario. Permiten el control de estas acciones. Ejemplo Multimedia ToolBook de Asymetrix.

Estos Sistemas de autor contienen un conjunto de herramientas que brindan entornos de trabajo y bloques básicos prefabricados para la creación de las aplicaciones Multimedia muchos de los cuales pueden ser utilizados incluso por los no programadores. Se recomiendan para aplicaciones con gran variedad de contenidos. Entre sus beneficios tenemos ciclos de desarrollo más cortos, características más predecibles y mayor confiabilidad del producto final.

4.3. Integración del contenido y los medios en su forma final

Una vez expresados todos los elementos del diseño y seleccionado el lenguaje de programación, se ensamblan todos los elementos desarrollando un producto de software, esta tarea es responsabilidad de los programadores que deben ajustarse al guión y utilizar las normas de diseño definidas y las bibliotecas con la información. Debe tenerse en cuenta por parte de los especialistas la confección de la Ayuda del sistema.

Una vez finalizada la etapa debe quedar definido :

- Que lenguaje de programación o sistema de autor será utilizado.
- Documentación asociada con el programa de computación.

5. Pruebas de la aplicación

En ningún proyecto informático debe pasarse por alto el proceso de pruebas que es el que garantiza la salida de un producto de calidad. Un software Multimedia debe revisarse desde dos puntos de vista:

- Solidez de la información
- Adecuado funcionamiento.

Solidez de la información: Toda la información contenida en la aplicación debe ser verificada en cuanto a:

- No existencia de errores ortográficos.
- Calidad de los medios que se muestran.
- Correspondencia entre el tema tratado, el texto y el resto de los medios que aparecen en cada pantalla.
- Cumplimiento de las normas de diseño.

Adecuado funcionamiento: En este caso se trata de comprobar que cada acción del usuario tenga una respuesta correcta del sistema y que no ocurran errores imprevistos. Es necesario revisar lo previsto en el diseño de la interfaz.

5.1 Elaborar protocolo de pruebas

Cada uno de los puntos de vista antes expresado puede ser revisado a la vez o por separado, esto depende de la complejidad del sistema, el tamaño de la aplicación o la conveniencia a la hora de efectuar las pruebas.

Solidez de la información: Revisión de cada pantalla verificando los tipos de errores de acuerdo a lo ya tratado y comprobando con las fuentes y las normas de diseño establecidas.

Adecuado funcionamiento: En una aplicación Hipermedia como ya se ha expresado en la fundamentación teórica de este trabajo, el usuario del sistema puede acceder a la información de forma no secuencial, o sea, satisfaciendo su curiosidad o necesidad según sea el caso. Si se entrega un producto de este tipo para ser probado, los probadores en su interacción podrían dejar de recorrer algún camino y por tanto dejar el sistema vulnerable en zonas no revisadas.

De acuerdo con el diagrama de flujo y el guión debe elaborarse un protocolo de pruebas que garantice el recorrido a todas las vías posibles y que además permita comprobar si el programa se ajusta completamente a lo deseado.

5.2 Revisión y comprobación por el usuario

Crear un grupo externo al del proyecto para realizar las pruebas que se ejecutarán de acuerdo al calendario fijado y guiados por el protocolo.

Como resultado de esta etapa debe aparecer:

- Protocolo de prueba utilizado.
- Resultado de las pruebas.

6. Preparación para su distribución

Un producto que utiliza técnicas de Multimedia en muchos casos puede resultar de interés comercial, por lo que si se decidió su comercialización debe prestársele la máxima atención.

6.1. Determinación de la forma de distribución

Debe decidirse si se distribuirá utilizando disquete o CD-ROM, la tendencia actual es a distribuir los productos en CD-ROM como ya se ha planteado. Pueden aparecer causas que determinen utilizar los disquetes, como por ejemplo: la no tenencia de lectores de CD entre los usuarios potenciales, no ocupar gran capacidad de memoria la aplicación que no justifica los gastos, etc.

6.2. Diseño de la empaquetadura

Se entiende como empaquetadura, el medio que se utilizará para contener los discos que componen el producto. Debe incluirse:

- Para disquetes: Diseño de la etiqueta.
- Para CD: Diseño de carátula.

El tipo de empaquetadura (cajas plásticas, sobres de nylon, cartulina, etc.) se selecciona teniendo en cuenta la disponibilidad en el mercado, el tipo de producto, el presupuesto de que se dispone. Cualquiera sea el tipo seleccionado deben diseñarse los materiales asociados a la misma y una portada.

6.3. Preparación para su producción

Si son disquetes: Preparación de instalador.

Preparación del primer juego de discos.

Prueba de la aplicación preparada desde discos.

Envío a producción.

Si es CD-ROM: Preparación de instalador

Preparación en disco duro de una simulación del contenido del CD.

Quema del premaster.

Prueba de la aplicación desde el premaster.

Envío a producción.

6.4. Elaboración de documentos comerciales

Si se decidió producir una aplicación para su comercialización, debe trabajarse y en paralelo con el resto de las etapas en las líneas y estrategias de comercialización con el estudio de todos los clientes potenciales y la preparación de todos los materiales que permitirán promocionar el producto.

Como resultado debe quedar:

- Producto final empaquetado.
- Documentos comerciales.

CAPÍTULO 3. EJEMPLO PRÁCTICO DE LAS METODOLOGÍAS NDT Y MULTIMET

3.1 Ejemplo práctico de la metodología NDT

3.1.1. Introducción

En este ejemplo se describe el proceso de aplicación de NDT. No se muestran los resultados de la aplicación de NDT, éstos serían el documento de requisitos y el documento de análisis, así como los prototipos generados. El ejemplo muestra el trabajo realizado durante el proceso de desarrollo utilizando NDT, es decir, el trabajo realizado durante la especificación de requisitos y el análisis, las vías seguidas para la aplicación de los procesos sistemáticos y derivación y la forma en la que se han conseguido los resultados finales de una pequeña parte de dicho sistema.

3.1.2. El sistema para la generación de itinerarios culturales

El interés que el patrimonio histórico ha adquirido en los últimos años, hace que la demanda de información por parte de turistas, visitantes, interesados en patrimonio o investigadores, sea cada vez más grande. Por ello, han buscado nuevos sistemas y medios que faciliten y agilicen el trabajo de dar información a los consumidores de la misma. Entre éstos se encuentra el sistema de generación de itinerarios culturales que tiene como objetivo ofrecer una herramienta vía Web que permita a cualquier persona interesada en el patrimonio histórico, generar el itinerario cultural más adecuado a su perfil. El sistema, desde su propia definición, plantea una serie de características complejas; la primera de ellas es que el sistema va a recibir peticiones de usuarios con diferente nivel de conocimiento del área patrimonial. De la misma forma puede acceder al sistema un simple turista que desea visitar monumentos que investigadores de todo el mundo interesados en estudiar un estilo pictórico muy concreto. Por ello, el sistema debe adecuarse a los diferentes usuarios que lo utilicen.

Se pretende realizar la fase de requisitos y análisis para un sistema de generación de itinerarios culturales. En él, el usuario que se conecte a través de Internet debe mostrar, mediante una serie de pantallas de consulta, las características de los

monumentos que desea visitar. El sistema determinará los más adecuados según el perfil del usuario y mostrará, bajo un mapa la situación de los mismos, dando la posibilidad al usuario de que obtenga mayor información de los monumentos mostrados. Como perfiles de los usuarios se tendrá a los turistas en general y a los investigadores. Dentro de los investigadores se hace diferencia también entre si éstos son arqueólogos, etnólogos o artistas. El sistema varía la información que maneja en función de cada uno de ellos.

Con todo esto, en este capítulo se muestran cómo se van consiguiendo los resultados y modelos y cómo se van aplicando las técnicas de NDT.

Fase Ingeniería de requisitos

En la fase de ingeniería de requisitos se realizan tareas para la generación de itinerarios culturales. Las mismas serán explicadas a continuación.

- **Obtener información sobre el entorno y definir los objetivos**

El sistema resulta novedoso porque no sólo está orientado a personas expertas en patrimonio, sino que además se dirige a visitantes anónimos por lo que supone un reto importante en cuanto a la definición de la interfaz. Por ello, una vez definido el sistema en general, el trabajo de la especificación de requisitos comienza a estudiar el proceso de cómo se realizaba la tarea de dar información sobre el patrimonio antes del nuevo desarrollo. Para ello, se estudian un conjunto de formularios que los interesados debían enviar por fax o por e-mail al centro con sus peticiones.

Este proceso, ofrece una serie de carencias, como el tiempo que tiene que esperar el solicitante o la carga de trabajo que provoca. Pero el estudio de los formularios permite recuperar información y acercar al grupo de desarrollo al problema.

En este caso, la técnica usada es la de estudiar sistemas anteriores y realizar entrevistas al personal para detectar los objetivos que deben cubrirse en el sistema. En otros casos, se pueden aplicar técnicas diferentes para la elicitación de los objetivos, como el JAD o el Brainstorming, que en otros entornos pueden resultar más adecuadas. Tras la elicitación de objetivos, se hace una descripción de los

mismos usando el primer patrón propuesto por NDT. En este sistema se pueden detectar dos objetivos que son los que se muestran en las tablas 3.1 y 3.2.

Tabla 3.1- Patrón del objetivo OBJ-01

OBJ-01	Ofrecer itinerarios culturales a los visitantes
Descripción	El sistema deberá permitir a los usuarios describir los elementos de patrimonio que les interesa y generar un itinerario cultural en el que se resalten los elementos patrimonio histórico de Andalucía que más se ajusten a su petición. Dichos elementos se mostrarán situados sobre un mapa de la comunidad autónoma desde el que el usuario podrá, pinchando sobre un bien en concreto, acceder a la información concreta del bien.

Tabla 3.2- Patrón del objetivo OBJ-02

OBJ-02	Adecuar el sistema al perfil del usuario
Descripción	El sistema debe ser capaz de adaptarse al perfil del usuario concreto con el que se está trabajando en cada momento. En caso de que el usuario no se identifique será tratado como un turista anónimo, si se identifica, se tratará como un investigador que podrá ser un arqueólogo, un etnólogo o un artista o cualquier conjunción de ambos.

De estos patrones se han eliminado campos como las fuentes, la versión o los comentarios, porque no resultan relevantes para el objetivo de este anexo. Sin embargo, en la realidad de la empresa, en la definición de objetivos y requisitos es interesante controlar aspectos como las fuentes o las versiones puesto que en el proyecto deben participar personas de áreas muy diferentes: personal del centro de difusión del patrimonio, documentalistas, personal del área etnológica, arqueológica, etnológica, etc.

Identificar y definir los requisitos de almacenamiento de información

Para la definición de las necesidades de almacenamiento del sistema, fue fundamental el uso de los formularios que el centro había puesto en disposición del público para realizar sus peticiones de información y los documentos tipo de que disponía para la generación de los resultados. Con ellos, se hizo una primera propuesta de requisitos que hubo que validar en varias reuniones con los diferentes usuarios del sistema.

Con los formularios y documentos y en la realización de las entrevistas que se realizan, se detectó que los monumentos se clasifican en dos grandes grupos: el patrimonio mueble y el patrimonio inmueble y, aunque en muchos casos la información almacenada para cada uno puede coincidir, en otros es muy diferente. Por ello, se definieron dos requisitos de almacenamiento, uno para cada uno, que son los que se muestran en los patrones de las tablas 3.3 y 3.4.

Tabla 3.3. Patrón del requisito RA-01

RA-01	Bien mueble	
Objetivos	OBJ-01: Ofrecer itinerarios culturales a los visitantes	
Descripción	El sistema deberá almacenar la información correspondiente a los bienes muebles que se encuentran en el sistema. En concreto:	
Datos específicos	Nombre y descripción	Naturaleza
	Código bien: almacena información sobre el código que identifica de manera unívoca a cada bien	NA-01
	Ubicación: recoge una referencia al bien inmueble en el que se ubica la pieza	RA-02
	Nombre: almacena información sobre el nombre que identifica al bien	Cadena
	Descripción: recoge una descripción detallada del bien	Documento
	Cronología: establece los períodos sobre los que se ubica la creación del bien	Fecha Formato: {aaaa-aaaa}
	Imagen: almacena una galería de imágenes del centro gráfico donde aparece el bien	Imagen Cardinalidad: 0..n
	Autor: almacena una lista de los autores que o bien han fabricado el bien mueble o bien ha participado en su construcción	RA-03 Cardinalidad: 0..n
	Tipología: recoge la tipología del bien o el conjunto de tipologías en los que se enmarca el bien	RA-03 Cardinalidad: 0..n
	Estilo: recoge los estilos artísticos en los que se registra el bien	RA-03 Cardinalidad: 0..n
	Período Histórico: recoge la lista de períodos históricos en los que se cataloga el bien	RA-03 Cardinalidad: 0..n
	Escuela: recoge el grupo de escuelas cuyos estilos se encuentran en el bien	RA-03 Cardinalidad: 0..n
	Actividad: recoge el conjunto de actividades para los que se supone que era utilizado el bien.	RA-03 Cardinalidad: 0..n
	Historia de la pieza: recoge la historia conocida de la pieza	Documento

Tabla 3.4. Patrón del requisito RA-02

RA-02	Bien inmueble	
Objetivos asociados	OBJ-01: Ofrecer itinerarios culturales a los visitantes	
Descripción	El sistema deberá almacenar la información correspondiente a los bienes inmuebles que se encuentran en el sistema. En concreto:	
Datos específicos	Nombre y descripción	Naturaleza
	Código bien: almacena información sobre el código que identifica de manera unívoca a cada bien	NA-02
	Coordenadas utm: recoge las coordenadas x e y en las que se ubica el bien. Servirán para localizar los bienes en el mapa de la localidad	NA-03
	Nombre: almacena información sobre el nombre que identifica al bien	Cadena
	Descripción: recoge una descripción detallada del bien	Documento
	Bien que contiene: almacena una lista con los bienes muebles que se encuentran ubicados en él.	RA-01 Cardinalidad: 0..n
	Cronología: establece los períodos sobre los que se ubica la creación del bien	Fecha Formato: {aaaa-aaaa}
	Imagen: almacena una galería de imágenes del centro gráfico donde aparece el bien	Imagen Cardinalidad: 0..n
	Autor: almacena una lista de los autores que o bien han fabricado el bien mueble o bien ha participado en su construcción	RA-03 Cardinalidad: 0..n
	Tipología: recoge la tipología del bien o el conjunto de tipologías en los que se enmarca el bien	RA-03 Cardinalidad: 0..n
	Estilo: recoge los estilos artísticos en los que se registra el bien	RA-03 Cardinalidad: 0..n
	Período Histórico: recoge la lista de períodos históricos en los que se cataloga el bien	RA-03 Cardinalidad: 0..n
	Actividad: recoge el conjunto de actividades para los que se supone que era utilizado el bien.	RA-03 Cardinalidad: 0..n

Con respecto a los datos concretos cuya naturaleza es una naturaleza predefinida hay poco que decir. Lo que sí es necesario destacar son los otros dos casos. En el caso de los datos que tiene como naturaleza otro requisito de almacenamiento se dan dos situaciones diferentes. La primera aparece en el patrón RA-01. Durante las entrevistas con los usuarios, se detectó que cada bien mueble está ubicado en un

bien inmueble, por tanto es necesario almacenar esta relación en el sistema (ver Tabla 3.3).

El segundo caso, se da en todos los campos de los patrones RA-01 y RA-02 cuya naturaleza es RA-03. En las entrevistas se detecta que existen una serie de datos de los bienes cuya información está normalizada bajo un conjunto de términos cerrados que se encuentran catalogados dentro del Tesoro de Patrimonio Histórico. Por esta razón, se detecta la necesidad de definir un nuevo patrón de requisitos de almacenamiento que recoja la necesidad que tiene el sistema de tratar los términos del tesoro. Este patrón es el que se muestra en la tabla 3.5.

Tabla 3.5 Patrón del requisito RA-03

RA-03	Dato de tesoro	
Objetivos asociados	OBJ-01: Ofrecer itinerarios culturales a los visitantes	
Descripción	El sistema deberá almacenar la información correspondiente a los datos normalizados bajo la terminología del Tesoro de Patrimonio Histórico. En concreto:	
Datos específicos	Nombre y descripción	Naturaleza
	Término: nombre del término normalizado	Cadena
	Descripción: recoge una descripción del significado del término	Cadena
	Sinónimos: recoge una lista de términos sinónimos que también se encuentran en el tesoro	RA-03 Cardinalidad: 0..n
	Específicos: recoge una lista de los términos que describen el mismo concepto pero de una manera más concreta	RA-03 Cardinalidad: 0..n

Con este patrón se tienen definidos todos los requisitos de almacenamiento para el ejemplo. Sin embargo, en los patrones RA-01 y RA-02 se detalla la necesidad de definir nuevas naturalezas. En concreto son tres las definidas, que se corresponden a los patrones de las tablas 3.6, 3.7, 3.8 respectivamente.

Tabla 3.6. Patrón de la naturaleza NA-01

NA-01	Código Bien Mueble	
Objetivos asociados	OBJ-01: Ofrecer itinerarios culturales a los visitantes	
Descripción	Esta naturaleza representa la estructura que describe el código unívoco identificativo de cada bien mueble otorgado por la Dirección General de Bienes Culturales.	
Datos específicos	Campo	Naturaleza
	Bien inmueble: recoge una referencia al bien inmueble en el que fue hallado el bien mueble	NA-02
	Mueble: es una cadena de tres dígitos que guarda el número particular que tiene ese mueble en el inmueble.	Cadena Tamaño: 3
Presentación	Los datos se presentan mediante un código numérico de 12 dígitos. Los 9 primeros se corresponden con el código del bien inmueble, luego se acompaña con un punto, seguido de los tres dígitos correspondientes al código del mueble: XXXXXXXXX.XXX	

Tabla 3.7. Patrón de la naturaleza NA-02

NA-02	Código Bien Inmueble	
Objetivos asociados	OBJ-01: Ofrecer itinerarios culturales a los visitantes	
Descripción	Esta naturaleza representa la estructura que describe el código unívoco identificativo de cada bien inmueble otorgado por la Dirección General de Bienes Culturales.	
Datos específicos	Campo	Naturaleza
	Provincia: almacena, mediante un código de dos números, la provincia en la que se encuentra el monumento.	Cadena Tamaño: 2
	Municipio: almacena, mediante un código de tres números, el municipio de la provincia en el que se ubica el monumento	Cadena Tamaño: 3
	Inmueble: es una cadena de cuatro dígitos que guarda el número particular que tiene ese inmueble en el municipio.	Cadena Tamaño: 4
Presentación	Los datos se presentan mediante un código numérico de 9 dígitos. Los 2 primeros se refieren a la provincia, los tres siguientes al código del municipio y los cuatro siguientes al código de inmueble. Todos aparecen como una cadena continua: XXXXXXXXX	

Tabla 3.8. Patrón de la naturaleza NA-03

NA-03	Coordenada UTM	
Objetivos asociados	OBJ-01: Ofrecer itinerarios culturales a los visitantes	
Descripción	Esta naturaleza representa la estructura que deben tener las coordenadas X e Y que delimitan cada monumento en el plano.	
Datos específicos	Campo	Naturaleza
	Coordenada X: representa el valor de la	Real
	Coordenada Y: representa el valor de la coordenada Y	Real

Con este ejemplo se puede entender mejor la definición de naturaleza dada durante el capítulo cuatro y separarla mejor del concepto de requisito de almacenamiento. El bien mueble, el inmueble o los términos del tesoro representan información con la que debe trabajar el sistema. Sin embargo, el código inmueble y mueble o las coordenadas UTM que describen un dominio de datos con un significado relevante para el grupo de usuarios.

Identificar y definir los actores

Ya con las primeras entrevistas se detectó una clasificación inicial de los actores básicos. Esta primera clasificación llevaba a detectar dos roles independientes e incompatibles: el turista y el investigador. Así que se comenzó definiendo dichos roles en los patrones adecuados y que se corresponden con los de la tabla 3.9 y 3.10.

Tabla 3.9. Patrón del actor AC-01

AC-01	Turista
Objetivos asociados	OBJ-02: Adecuar el sistema al perfil del usuario
Descripción	El sistema deberá prever el tratamiento de los usuarios que pertenecen al grupo descrito como turista y que se refiere a personas que se conectan al sistema sin ser expertos ni investigadores del patrimonio cultural con fines puramente informativos y turísticos.

Tabla 3.10. Patrón del actor AC-02

AC-02	Investigador
Objetivos asociados	OBJ-02: Adecuar el sistema al perfil del usuario
Descripción	El sistema deberá prever el tratamiento de los usuarios que pertenecen al grupo descrito como investigador y que se refiere a personas que se conectan al sistema siendo expertos en gestión patrimonial.

Sin embargo, y a medida que se entraba en más detalles en las entrevistas, se detectó que era necesario entrar en más detalle en la definición de los posibles investigadores. Esto se debía a que el sistema era muy diferente si el usuario conectado era un arqueólogo o un artista, así que se definieron tres roles de actores más que heredaban del investigador. Para su definición se diseñó el diagrama de la figura 3.1, así como los tres patrones de descripción de las tablas 3. 11, 3. 12 y 3.1 3.

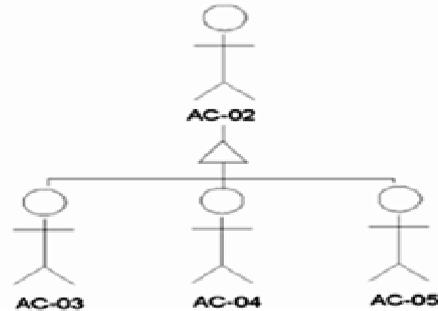


Figura 3.1-Generalización de actores

Tabla 3.11. Patrón del actor AC-03

AC-03	Arqueólogo
Objetivos	OBJ-02: Adecuar el sistema al perfil del usuario
Clasificación	Este es uno de los posibles roles dentro del sistema cuando se hace una clasificación de los actores en base a el tipo de investigador que se conecta en el sistema.
Descripción	El sistema deberá prever el tratamiento de los usuarios que pertenecen al grupo descrito como arqueólogo y que se refiere a personas que trabajan con el sistema y que son expertos en materia de arqueología.
Hereda de	AC-02: Investigador

Tabla 3.12. Patrón del actor AC-04

AC-04	Etnólogo
Objetivos	OBJ-02: Adecuar el sistema al perfil del usuario
Clasificación	Este es uno de los posibles roles dentro del sistema cuando se hace una clasificación de los actores en base a el tipo de investigador que se conecta en el sistema.
Descripción	El sistema deberá prever el tratamiento de los usuarios que pertenecen al grupo descrito como arqueólogo y que se refiere a personas que trabajan con el sistema y que son expertos en materia de etnología.
Hereda de	AC-02: Investigador

Tabla 3.13. Patrón del actor AC-05

AC-05	Artista
Objetivos	OBJ-02: Adecuar el sistema al perfil del usuario
Clasificación	Este es uno de los posibles roles dentro del sistema cuando se hace una clasificación de los actores en base a el tipo de investigador que se conecta en el sistema.
Descripción	El sistema deberá prever el tratamiento de los usuarios que pertenecen al grupo descrito como arqueólogo y que se refiere a personas que trabajan con el sistema y que son expertos en materia de historia del arte.
Hereda de	AC-02: Investigador

Tras estas definiciones llega el momento de definir la incompatibilidad entre actores. Por parte de los usuarios se define claramente que jamás un investigador puede hacer a la vez de turista o a la inversa. Esto provoca que ambos roles sean incompatibles y también lo sea el turista con el arqueólogo, el etnólogo y el artista. Con todo esto se define la matriz de la figura 3.14 para mostrar esta incompatibilidad.

Tabla 3.14. Matriz de incompatibilidad de actores

Actores	AC-01	AC-02	AC-03	AC-04	AC-05
AC-01	-	X	X	X	X
AC-02		-			
AC-03			-		
AC-04				-	
AC-05					-

Con el estudio de esta matriz, sorprende ver que mientras que, por ejemplo, un arqueólogo no puede ser tratado nunca como un turista, sí que podría jugar a la vez el papel de etnólogo, en cuyo caso, asumiría ambos papeles. Con esta inquietud, se comenta el problema a los usuarios para detectar si es necesario identificar actores derivados. Éstos indican que en los casos en los que un investigador asuma ambos roles, el sistema deberá comportarse como se comporta en el caso de cada uno. Es decir, que un usuario que asume el papel de arqueólogo y etnólogo a la vez no añada nueva funcionalidad o una visión diferente del sistema que no exista previamente para el arqueólogo o el etnólogo. Por esta razón, no tiene sentido definir actores derivados, puestos que los posibles no añaden nada nuevo al sistema.

Identificar y definir los requisitos funcionales

El sistema real sobre el que se basa este ejemplo, tiene una funcionalidad importante. Sin embargo, con el recorte realizado, la funcionalidad se puede recoger en un solo diagrama de casos de uso que es el que se muestra en la figura 3.2. Los casos de uso de dicho diagrama, se describen usando los patrones correspondientes en las tablas 3.15, 3.16 y 3.17.

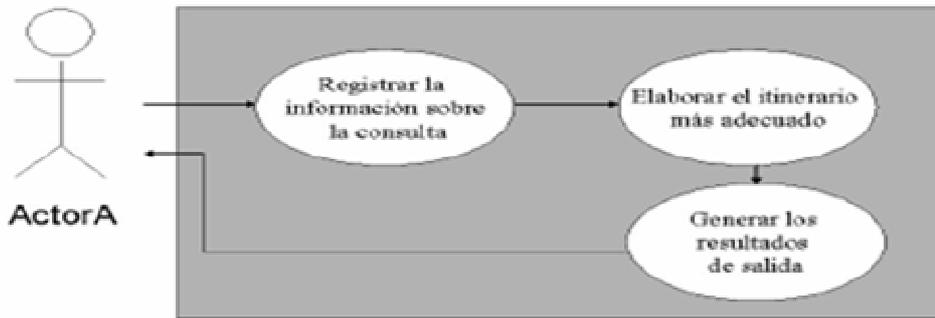


Figura 3.2- Diagrama de caso de uso del sistema

Tabla 3.15. Patrón del requisito RF-01

RF-01	Registrar la información sobre la consulta	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes 	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso y que representa la posibilidad que el sistema debe ofrecer al usuario de describir los criterios que requiere para su itinerario.	
Actores	Actor caso de uso	Actor del sistema
	Actor A	<ul style="list-style-type: none"> • AC-01: Turista • AC-02: Investigador
Secuencia normal	Paso	Acción
	1	El usuario solicita la realización de una consulta
	2	El sistema muestra la pantalla de consulta
	3	El usuario introduce todas las restricciones y sus intereses en la pantalla, indicando, además, si es experto en alguna materia: arqueología, etnología o arte.

Tabla 3.16. Patrón del requisito RF-02

F-02	Elaborar el itinerario más adecuado	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes 	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso y que representa la posibilidad que el sistema debe ofrecer al usuario de seleccionar, en base a los criterios establecidos por el mismo, los bienes que más se adecuan a los gustos del usuario.	
Actores	Actor caso de uso	Actor del sistema
	Actor A	<ul style="list-style-type: none"> • AC-01: Turista • AC-02: Investigador
Secuencia normal	Paso	Acción
	1	El sistema recibe los datos de la pantalla de consulta y el perfil, si existe, de investigación del usuario
	2	El sistema realiza la consulta adecuada al sistema gestor de base de datos para recuperar los bienes de interés para el usuario.

Tabla 3.17. Patrón del requisito RF-03

RF-03	Generar los resultados de salida	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes • OBJ-02: Adecuar el sistema al perfil del usuario 	
Descripción	El sistema deberá comportarse tal y como se describe en el siguiente caso de uso y que representa la posibilidad que el sistema debe ofrecer al usuario de, en base a la consulta resultante, generar el mapa de Andalucía en el que se localicen los bienes recuperados con la posibilidad de que el usuario acceda a la información concreta de cada uno de ellos.	
Actores	Actor caso de uso	Actor del sistema
	Actor A	<ul style="list-style-type: none"> • AC-01: Turista • AC-02: Investigador
Secuencia normal	Paso	Acción
	1	El sistema recibe los datos de la consulta y coloca los bienes en el mapa de la localidad utilizando las coordenadas utm en el caso de bienes inmuebles y, en el caso de los bienes muebles, las coordenadas utm de los inmuebles en los que está ubicado.
	2	El sistema muestra el resultado permitiendo posibilidad de acceder a la información concreta de cada bien.

Identificar y definir los requisitos de interacción

Los requisitos de interacción se dividen en dos grupos. El primero de ellos, las frases, adquieren un carácter crítico en este sistema puesto que es un sistema principalmente de consulta.

Para la definición de las mismas, se puso a disposición del usuario los patrones de requisitos de almacenamiento de información referente a los bienes muebles y a los bienes inmuebles (patrones RA-01 y RA-02 respectivamente) con idea de que pudieran determinar qué campos entrarían en la consulta.

Se define así una frase para cada tipo de bien. El primero, correspondiente a los bienes muebles, se describe mediante el patrón que se recoge en la tabla 3.18. El segundo, correspondiente a los bienes inmuebles, es el que se recoge en la tabla 3.19. Hay que destacar en ambas tablas varias ideas. La primera de ellas es la posibilidad que la generalización de actores que ofrece para simplificar la definición.

Así, cuando en un cuerpo se indica que puede ser interesante para el actor AC-02, indirectamente se indica que es interesante para todos los tipos de investigadores.

Otra idea importante es ver que en la definición de las frases, queda patente que la visión del turista es la más general de todas mientras que la especificación de perfil de investigador, selecciona los criterios de recuperación que le interesa. Así, las actividades sólo son interesantes para los etnólogos y los estilos para los artistas.

Tabla 3.18. Patrón de la frase FR-01

FR-01	Recuperación de bienes muebles	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes • OBJ-02: Adecuar el sistema al perfil del usuario 	
Cuerpo	Descripción	Actores
	El concepto RA-01.Código bien debe ser exactamente	•AC-02: Investigador
	El concepto RA-01.Nombre contener la siguiente cadena	•AC-01: Turista •AC-02: Investigador
	El concepto RA-01.Autor.Término debe contener la siguiente cadena	•AC-01: Turista •AC-05: Artista
	El concepto RA-01.Tipología.Término debe contener la siguiente cadena	•AC-01: Turista •AC-02: Investigador
	El concepto RA-01.Estilo.Término debe contener la siguiente cadena	•AC-01: Turista •AC-05: Artista
	El concepto RA-01.Período Histórico.Término debe contener la siguiente cadena	•AC-01: Turista •AC-02: Investigador
	El concepto RA-01.Escuela.Término debe contener la siguiente cadena	•AC-01: Turista •AC-05: Artista
	El concepto RA-01.Actividad.Término debe contener la siguiente cadena	•AC-01: Turista •AC-04: Etnólogo

Tabla 3.19. Patrón de la frase FR-02

FR-02	Recuperación de bienes inmuebles	
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes • OBJ-02: Adecuar el sistema al perfil del usuario 	
Cuerpo	Descripción	Actores
	El concepto RA-01.Código bien debe ser exactamente	<ul style="list-style-type: none"> • AC-02: Investigador
	El concepto RA-02.Nombre contener la siguiente cadena	<ul style="list-style-type: none"> • AC-01: Turista • AC-02: Investigador
	El concepto RA-02.Autor.Término debe contener la siguiente cadena	<ul style="list-style-type: none"> • AC-01: Turista • AC-05: Artista
	El concepto RA-02.Tipología.Término debe contener la siguiente cadena	<ul style="list-style-type: none"> • AC-01: Turista • AC-02: Investigador
	El concepto RA-02.Estilo.Término debe contener la siguiente cadena	<ul style="list-style-type: none"> • AC-01: Turista • AC-05: Artista
	El concepto RA-02.Período Histórico.Término debe contener la siguiente cadena	<ul style="list-style-type: none"> • AC-01: Turista • AC-02: Investigador
	El concepto RA-02.Actividad.Término debe contener la siguiente cadena	<ul style="list-style-type: none"> • AC-01: Turista • AC-04: Etnólogo

Tras la definición de las frases, llega el momento de definir los prototipos de visualización del sistema. En el sistema existen dos prototipos de visualización. El primero de ellos permite visualizar el contenido de un bien mueble para los turistas. Al igual que en las frases, la estructura del prototipo se va a condicionar al perfil del usuario que en cada momento esté conectado al sistema.

El prototipo para mostrar la información de un bien mueble cuando el que está interactuando con el sistema es un turista es el que se muestra en la tabla 3.20.

Tabla 3.20. Patrón para el prototipo PV-01

PV-01	Ficha de Bien Mueble para turistas
Objetivos	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes
Actor/es	<ul style="list-style-type: none"> • AC-01: Turista
Descripción	El sistema deberá permitir la visualización de los datos concretos
Frases	FR-01: Recuperación de bienes muebles
Información visualizada	RA-01.Nombre RA-01.Descripción RA-01.Ubicación RA-01.Cronología RA-01.Imagen RA-01.Historia de la pieza RA-01.Autor.Término RA-01.Tipología.Término RA-01.Estilo.Término RA-01.Período Histórico.Término RA-01.Escuela.Término RA-01.Actividad.Término
Prototipos de salida	PV-02 (DeVuelta)
Prototipos de entrada	PV-02

Desde este prototipo se puede viajar hacia el prototipo PV-02. Este prototipo, equivalente al anterior, es el que se muestra en la tabla 3.21 y que se corresponde al prototipo para visualizar los bienes inmuebles en el caso de los turistas. La idea de que desde el bien mueble se pueda navegar al inmueble se refiere a que se navega hacia el bien en el que se ubica.

Lo mismo ocurre en el caso de los bienes inmuebles descritos en el prototipo PV-02. Desde él se puede navegar al bien mueble, sin embargo, en este caso, la navegación es múltiple puesto que un bien inmueble se puede relacionar con múltiples bienes muebles, todos los que en él se ubiquen. Nótese que en este prototipo se resalta que el usuario turista sólo está interesado en un conjunto de datos de los bienes. Así se han obviado campos como las coordenadas utm o los códigos puesto que no aportan nada al usuario no experto. Además, el turista no está interesado en entrar en especificidades como los sinónimos en tesauro o sus datos específicos. De ahí, que para términos como los autores, sólo se muestre el término del tesauro y no el elemento completo. Esto es lo que interesa a los investigadores.

En los siguientes patrones se muestran los prototipos de visualización que son interesantes desde el punto de vistas de los investigadores. Éstos se adecuan al perfil de cada uno de ellos, por lo que van a resultar más complejos.

Tabla 3.21. Patrón para el prototipo PV-02

PV-02	Ficha de Bien Inmueble para turistas
Objetivos	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes
Actor/es	<ul style="list-style-type: none"> • AC-01: Turista
Descripción	El sistema deberá permitir la visualización de los 1 que se muestran a continuación y la navegación expresada y que representa la información que se muestra al turista sobre los bienes muebles
Frases	FR-02: Recuperación de bienes inmuebles
Información visualizada	RA-02.Nombre
	RA-02.Descripción
	RA-02.Cronología
	RA-02.Imagen
	RA-02.Autor.Término
	RA-02.Tipología.Término
	RA-02.Estilo.Término
	RA-02.Período Histórico.Término
RA-02.Actividad.Término	
Prototipos de salida	PV-01 (DeVuelta, Múltiple)
Prototipos de entrada	PV-01

En el caso de los turistas, la información sobre el bien se muestra en un único prototipo, en el caso de los investigadores existe un patrón con la información básica tanto para los bienes muebles como para los inmuebles que luego puede detallarse accediendo a los términos específicos del tesoro. Por tanto, aquí existen tres prototipos de visualización. El primero de ellos, es el que se muestra en la tabla 3.22 que se corresponde con la información básica para los bienes muebles.

El patrón es bastante similar al PV-01. De hecho, podría surgir la duda de si no sería posible crear un solo prototipo tanto para investigadores como turistas y utilizar la posibilidad de definir condiciones, salvando así las posibles diferencias que pueda haber. Realmente, esto podría hacerse, en cuyo caso se tendría un único prototipo para mostrar la información del bien mueble.

Tabla 3.22. Patrón para el prototipo PV-03

PV-03	Bien Mueble. Información básica
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes • OBJ-02: Adecuar el sistema al perfil del usuario
Actor/es	<ul style="list-style-type: none"> • AC-02. Investigador
Descripción	El sistema deberá permitir la visualización de los datos concretos que se muestran a continuación y la navegación expresada y que representa la información que se muestra a los investigadores sobre los
Frases	FR-01: Recuperación de bienes muebles
Información visualizada	RA-01.Código del bien
	RA-01.Nombre
	RA-01.Descripción
	RA-01.Ubicación
	RA-01.Cronología
	RA-01.Imagen
	RA-01.Historia de la pieza
	Si actor = AC-05 entonces RA-01.Autor.Término
	RA-01.Tipología.Término
	Si actor = AC-05 entonces RA-01.Estilo.Término
	RA-01.Período Histórico.Término
Si actor = AC-05 entonces RA-01.Escuela.Término	
Si actor = AC-04 entonces RA-01.Actividad.Término	
Prototipos de salida	PV-04 (DeVuelta) PV-05 (DeVuelta, Múltiple)
Prototipos de entrada	PV-04 PV-05

Es interesante observar también en la tabla 3.22 (o incluso en el ejemplo de la tabla 3.23) la potencia semántica que ofrecen las condiciones en el prototipo de visualización. En el caso del prototipo de la tabla 3.22, difiere muy poco si el que conecta es un arqueólogo, un etnólogo o un artista, por lo que no resulta complejo presentar un único patrón que visualice las necesidades de ambos roles. Sin embargo, también si el experto en requisitos lo estima oportuno, se podrían haber definido en lugar de uno, tres patrones, uno para cada tipo de investigador y así no tener que usar condicionales.

Además, el patrón de la tabla 3.22, muestra cómo se puede hacer uso de la generalización de actores definida en actividades anteriores. Nótese que el patrón se define para el actor genérico AC-02, usando las propiedades de la generalización, se indica que el prototipo sería accesible para cualquier investigador.

Este prototipo sería el que se muestra en la tabla 3.23. Realmente ambas soluciones, optar por tener un PV-01 y un PV-03, o tener un único PV-01, es igual para la aplicación de NDT. Sin embargo, la solución de optar por dos prototipos resulta más fácil de entender de manera general, y en concreto, para este sistema, para el grupo de usuarios y analistas.

Tabla 3.23. Patrón para el prototipo PV-01 fusionando PV-01 y PV-3

PV-01	Bien Mueble. Información básica
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes • OBJ-02: Adecuar el sistema al perfil del usuario
Actor/es	<ul style="list-style-type: none"> • AC-01. Turista • AC-02. Investigador
Descripción	El sistema deberá permitir la visualización de los datos concretos que se muestran a continuación y la navegación expresada y que representa la información que se
Frases	FR-01: Recuperación de bienes muebles
Información visualizada	Si actor = AC-02 entonces RA-01.Código bien RA-01.Nombre RA-01.Descripción RA-01.Ubicación RA-01.Cronología RA-01.Imagen RA-01.Historia de la pieza Si actor = AC-01 ó actor =AC-05 entonces RA-01.Autor.Término RA-01.Tipología.Término Si actor = AC-01 ó actor = AC-05 entonces RA-01.Estilo.Término RA-01.Período Histórico.Término Si actor = AC-01 ó actor = AC-05 entonces RA-01.Escuela.Término Si actor = AC-01 ó actor = AC-04 entonces RA-01.Actividad.Término
Prototipos de salida	Si actor = AC-01 entonces PV-02 (DeVuelta) sino PV-04 (DeVuelta) PV-05 (DeVuelta, Múltiple)
Prototipos de entrada	Si actor = AC-01 entonces PV-02 sino PV-04

Al separar los prototipos en dos, se hace una separación entre la visión del turista y del investigador que puede resultar muy adecuada a la hora de validar los requisitos por parte de los usuario. De todas formas, la elección depende de cada caso en concreto y debe ser el experto en ingeniería de requisitos el que decida qué solución tomar.

Luego, se puede concretar para especificar las necesidades de cada rol de investigador en concreto, como de hecho se hace en el prototipo con las condiciones. Para los bienes inmuebles, también se define un patrón específico para los investigadores, que, nuevamente, se concreta en algunos casos mediante condiciones, para recoger las necesidades de cada rol de investigador concreto. Este patrón es el que se muestra en la tabla 3.24.

Tabla 3.24. Patrón para el prototipo PV-04

PV-04	Bien Inmueble. Información básica
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes • OBJ-02: Adecuar el sistema al perfil del usuario
Actor/es	<ul style="list-style-type: none"> • AC-02. Investigador
Descripción	El sistema deberá permitir la visualización de los datos concretos que se muestran a continuación y la navegación expresada y que representa la información que se muestra a los investigadores sobre los bienes
Frases	FR-02: Recuperación de bienes inmuebles
Información visualizada	RA-02.Código del bien RA-02.Nombre RA-02.Descripción RA-02.Cronología RA-02.Imagen RA-02.Coordenada utm Si actor = AC-05 entonces RA-02.Autor.Término RA-02.Tipología.Término Si actor = AC-05 entonces RA-02.Estilo.Término RA-02.Período Histórico.Término Si actor = AC-04 entonces RA-02.Actividad.Término
Prototipos de salida	PV-03 (DeVuelta, Múltiple) PV-05 (DeVuelta, Múltiple)
Prototipos de entrada	PV-03 PV-05

Al igual que en el caso de los bienes muebles, las opciones de diseño son varias: se podría haber fusionado este prototipo con el de los bienes inmuebles para los

turistas o se podría haber separado en varios, uno por cada rol de investigador, para no necesitar hacer uso de las condiciones.

Para seguir el mismo criterio que en el caso anterior, se opta por un patrón para los turistas (el que se muestra en el prototipo PV-14) y uno para los investigadores en general, que es el mostrado en la 3.24, que se concreta para recoger las necesidades de cada investigador concreto.

El último prototipo también va a ser específico para los investigadores. Este prototipo va a permitir visualizar un término completo de tesoro. Es decir, todos los campos de cada término. Es el que se muestra en la tabla 3.25.

Tabla 3.25. Patrón para el prototipo PV-05

PV-05	Datos de los términos de tesoro
Objetivos asociados	<ul style="list-style-type: none"> • OBJ-01: Ofrecer itinerarios culturales a los visitantes • OBJ-02: Adecuar el sistema al perfil del usuario
Actor/es	<ul style="list-style-type: none"> • AC-02. Investigador
Descripción	El sistema deberá permitir la visualización de los datos concretos que se muestran a continuación y la navegación expresada y que representa la información que se muestra a los investigadores sobre cada término de tesoro.
Información visualizada	RA-03
Prototipos de salida	PV-03 (DeVuelta, Múltiple) PV-04 (DeVuelta, Múltiple) PV-05 (DeVuelta, Múltiple)
Prototipos de entrada	PV-03 PV-04 PV-05

Desde un término de tesoro, se puede navegar a muchos otros términos de tesoro. De ahí que en los prototipos de salida aparezca la posibilidad de navegar hacia el prototipo PV-05.

Además, también se puede ver que, en este caso, no se han especificado los datos específicos del requisito RA-03 que se visualizan. Esto se debe a que en el prototipo van a aparecer todos los datos específicos de RA-03 y en el mismo orden que fueron definidos en el prototipo.

Identificar y definir los requisitos no funcionales

Los requisitos no funcionales aparecen en NDT como una herramienta para contemplar requisitos no cubiertos en ninguno de los anteriores. Aunque en el

ejemplo real se especifican algunos requisitos de este tipo que cubren aspectos como la seguridad y la integridad de la información, no son relevantes para el objetivo de este anexo, por lo que no van a ser recogidos. Se supone, por tanto, que no se definen requisitos no funcionales.

Validar los requisitos

La siguiente tarea consiste en realizar una validación de los requisitos. Los patrones y modelos definidos en tareas anteriores, son la última versión de una serie de iteraciones en las que, a través de entrevistas y otras técnicas de elicitación, se hacía una definición de requisitos que eran revisadas por clientes y que eran modificadas para solventar los posibles errores e incongruencias. Como ya se ha comentado, el ejemplo mostrado es sólo una visión del ejemplo real, por lo que no tiene sentido aplicar técnicas más complejas de validación como las auditorías o la elaboración de tesauros. Sí que resulta interesante en este punto desarrollar la matriz de rastreabilidad para comprobar que todos los requisitos cubren los objetivos marcados. La matriz de rastreabilidad o trazabilidad es la que se muestra en la tabla 3.26.

Tabla 3.26. Matriz de rastreabilidad

	OBJ-01	OBJ-02
RA-01	*	
RA-02	*	
RA-03	*	
NA-01	*	
NA-02	*	
NA-03	*	
AC-01		*
AC-02		*
AC-03		*
AC-04		*
AC-05		*
RF-01	*	*
RF-02	*	*
RF-03	*	*
FR-01	*	*
FR-02	*	*
PV-01	*	*
PV-02	*	*
PV-03	*	*
PV-04	*	*
PV-05	*	*

Esta matriz permite ver que todos los objetivos que se han marcado en el sistema, en este ejemplo sólo dos, son alcanzados por al menos uno de los requisitos

planteados y que, además, todos los requisitos cubren al menos un objetivo, lo que garantiza la necesidad de definirlo.

En sistemas pequeños, como el de este ejemplo, la matriz de rastreabilidad no resulta tan importante como en grandes sistemas. Cuando el número de objetivos y requisitos definidos es elevado, en muchos casos cuesta controlar si realmente todos los objetivos han sido cubiertos. Para solventar esto, la matriz de rastreabilidad ofrece una visión gráfica de si realmente todos los objetivos han sido alcanzados.

Generar el documento de requisitos del sistema

Llegados a este punto, en el que ya los objetivos del sistema han sido marcados y los requisitos elicitados, definidos y validados por el grupo de usuarios, generar el documento de requisitos es una tarea sencilla. Consiste simplemente en recoger todos los patrones y diagramas en un documento que es el resultado de la ingeniería de requisitos que se toma para la siguiente fase.

2. Análisis

Partiendo de los resultados de la ingeniería de requisitos, llega el momento de realizar el análisis de los mismos. Al igual que en la fase anterior, se presenta el procedimiento actividad por actividad.

Realizar el modelo conceptual

El proceso de realización del modelo conceptual se compone, como ya se ha comentado, de dos tareas. En la primera se genera el modelo conceptual básico, de una manera sistemática, y en la segunda se estudia este modelo para encontrar posibles mejoras y conseguir así el modelo conceptual final.

Comenzando por la generación del modelo conceptual básico, el primer paso a realizar es la generación del diagrama de clases a partir de los requisitos de almacenamiento. Cada requisito de almacenamiento genera una clase en el modelo conceptual, con su mismo nombre y descripción.

La clase generada por un requisito concreto tiene por atributos los datos concretos de dicho requisito cuyas naturalezas no sean otro requisito de almacenamiento. De esta forma, a partir de los requisitos RA-01, RA-02 y RA-03 se generan las clases CL-01, CL-02 y CL-03 respectivamente. Los patrones de estas clases son los que se muestran en las tablas 3.27, 3.28 y 3.29 respectivamente.

Tabla 3.27. Patrón para la definición de la clase CL-01

CL-01	Bien mueble		
Requisito	RA-01.Bien mueble		
Descripción	El sistema deberá almacenar la información correspondiente a los bienes muebles que se encuentran en el sistema. En concreto:		
Atributos	Descripción	Significado	Dato
	Código bien: CLn-01	Almacena información sobre el código que identifica de manera unívoca a cada bien	RA-01.Código bien
	Nombre: Cadena	Almacena información sobre el nombre que identifica al bien	RA-01.Nombre
	Descripción: Documento	Recoge una descripción detallada	RA-01.Descrip
	Cronología: Fecha {formato: aaaa-aaaa}	Establece los períodos sobre los que se ubica la creación del bien	RA-01.Cronolo gía
	Imagen[0..n]: Imagen	Almacena una galería de imágenes del centro gráfico donde aparece el bien	RA-01.Imagen
	Historia de la pieza: Documento	Recoge la historia conocida de la pieza	RA-01.Historia
Estado	Básico		

Tabla 3.28. Patrón para la definición de la clase CL-02

CL-02	Bien inmueble		
Requisito	RA-02.Bien inmueble		
Descripción	El sistema deberá almacenar la información correspondiente a los bienes inmuebles		
Atributos	Descripción	Significado	Dato
	Código bien: CLn-02	Almacena información sobre el código que identifica de manera unívoca a cada bien	RA-02.Código bien
	Coordenadas utm: CIn-03	Recoge las coordenadas X e Y en las que se ubica el bien. Servirán para localizar los bienes en el mapa de la localidad	RA-02.Coordenadas utm
	Nombre: Cadena	Almacena información sobre el nombre que identifica al bien	RA-02.Nombre
	Descripción: Documento	Recoge una descripción detallada del bien	RA-02.Descripción
	Cronología: Fecha {formato: aaaa-aaaa}	Establece los períodos sobre los que se ubica la creación del bien	RA-02.Cronología
	Imagen [0..n]: Imagen	Almacena una galería de imágenes del centro gráfico donde aparece el bien	RA-02.Imagen
Estado	Básico		

Tabla 3.29. Patrón para la definición de la clase CL-03

CL-03	Dato de tesoro		
Requisito	RA-03.Dato de tesoro		
Descripción	El sistema deberá almacenar la información correspondiente a los datos normalizados bajo la terminología del Tesoro de Patrimonio Histórico. En concreto:		
Atributos	Descripción	Significado	Dato específico
	Término: Cadena	Nombre del término normalizado	RA-03.Término
	Descripción: Cadena	Recoge una descripción del significado del término	RA-03.Descripción
Estado	Básico		

Con respecto a las asociaciones hay que fijarse en los datos específicos de los patrones cuyas naturalezas sean a su vez un requisito de almacenamiento de información. Hay varios casos:

A. En los bienes muebles:

- i. Ubicación, de naturaleza RA-02 con cardinalidad 1.
- ii. Autor, Tipología, Estilo, Período Histórico, Escuela, Actividad de naturaleza RA-03 y con cardinalidad todos ellos de 0..n.

B. En los bienes inmuebles:

- i. Bien que contiene, de naturaleza RA-01 con cardinalidad 0..n.
- ii. Autor, Tipología, Estilo, Período Histórico, Actividad de naturaleza RA-03 y con cardinalidad todos ellos de 0..n.

C. En los datos del tesoro:

- i. Sinónimos y Específicos, de naturaleza RA-03 y cardinalidad 0..n todos ellos.

Con todo esto, se generan las siguientes asociaciones:

1. AS-01, será bidireccional y recoge la relación entre el requisito RA-01 y RA-02, su patrón es el que se describe en la tabla 3.30. En él puede verse que el rol de la asociación en la parte de RA-01 es *Bien que contiene* y la cardinalidad 0..n las que tiene el dato específico de naturaleza RA-01 en RA-02. En la parte de RA-02, el rol es ubicación y la cardinalidad 1, como en el patrón de RA-01. Nótese que el nombre de la asociación queda sin definir en el modelo básico.

Tabla 3.30- Patrón para la definición de la asociación AS-01

AS-01			
Requisitos	<ul style="list-style-type: none"> • RA-01: Bien mueble • RA-02: Bien inmueble 		
Descripción	Las clases CL-01 y CL-02 se relacionan mediante esta asociación que representa una referencia al bien inmueble en el que se ubica la pieza y la lista de bienes muebles que se encuentran ubicados en el bien inmueble.		
Tipo	Bidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-01	Bien que contiene	0..n
	CL-02	Ubicación	1..1
Datos específicos	<ul style="list-style-type: none"> • RA-01.Ubicación • RA-02.Bien que contiene 		
Estado	básico		

2. AS-02, AS-03, AS-04, AS-05, AS-06 y AS-07, que son unidireccionales y recogen las relaciones de Autor, Tipología, Estilo, Período Histórico, Escuela y Actividad respectivamente de RA-01. Son unidireccionales porque en RA-03 no aparecen

datos específicos con naturaleza RA-01 que garanticen la bidireccionalidad. Sus patrones de definición aparecen respectivamente en las tablas 3.31, 3.32, 3.33, 3.34, 3.35 y 3.36. En estas asociaciones, la cardinalidad de la clase CL-03 y el rol en cualquiera de los patrones es el heredado del atributo correspondiente en RA-01. Mientras que el rol y la cardinalidad de la clase CL-01 en cualquiera de ellos es la de por defecto: Bien mueble y 0..n.

Tabla 3.31. Patrón para la definición de la asociación AS-02

AS-02			
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-01 y CL-03 se relacionan mediante esta asociación que representa una lista de autores que o bien han fabricado el bien mueble o bien ha participado en su construcción.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-01	Autor	0..n
	CL-03	Bien mueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Autor 		
Estado	básico		

Tabla 3.32. Patrón para la definición de la asociación AS-03

AS-03			
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-01 y CL-03 se relacionan mediante esta asociación que representa la tipología del bien o el conjunto de tipologías en los que se enmarca el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-01	Tipología	0..n
	CL-03	Bien mueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Tipología 		
Estado	básico		

Tabla 3.33. Patrón para la definición de la asociación AS-04

AS-04			
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-01 y CL-03 se relacionan mediante esta asociación que representa los estilos artísticos en los que se registra el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-01	Estilo	0..n
	CL-03	Bien mueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Estilo 		
Estado	básico		

Tabla 3.34. Patrón para la definición de la asociación AS-05

AS-05			
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-01 y CL-03 se relacionan mediante esta asociación que representa la lista de períodos históricos en los que se cataloga el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-01	Período Histórico	0..n
	CL-03	Bien mueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Período Histórico 		
Estado	básico		

Tabla 3.35. Patrón para la definición de la asociación AS-06

AS-06			
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-01 y CL-03 se relacionan mediante esta asociación que representa el grupo de escuelas cuyos estilos se encuentra el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-01	Escuela	0..n
	CL-03	Bien mueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Escuela 		
Estado	básico		

Tabla 3.36. Patrón para la definición de la asociación AS-07

AS-07			
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-01 y CL-03 se relacionan mediante esta asociación que representa el conjunto de actividades para los que se supone que era utilizado el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-01	Actividad	0..n
	CL-03	Bien mueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Actividad 		
Estado	básico		

3. Con los datos específicos de RA-02 de Autor, Tipología, Estilo, Período Histórico y Actividad, ocurre algo muy similar a los anteriores. Igualmente surge una asociación por cada uno de ellos, AS-08, AS-09, AS-10, AS-11 y AS-12, respectivamente que se muestran en las tablas 3.37, 3.38, 3.39, 3.40 y 3.41. Ocurre igual que en el caso anterior, las asociaciones son unidireccionales porque en RA-03 no hay datos específicos de tipo RA-02. El rol y la cardinalidad que la clase CL-03 asume por defecto es Bien Inmueble y la cardinalidad 0..n. El rol y la cardinalidad de CL-02 vienen dadas por los valores del dato concreto.

Tabla 3.37. Patrón para la definición de la asociación AS-08

AS-08			
Requisitos	<ul style="list-style-type: none"> RA-02: Bien inmueble 		
Descripción	Las clases CL-02 y CL-03 se relacionan mediante esta asociación que representa una lista de autores que o bien han fabricado el bien mueble o bien ha participado en su construcción.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-02	Autor	0..n
	CL-03	Bien inmueble	0..n
Datos	<ul style="list-style-type: none"> RA-02.Autor 		
Estado	básico		

Tabla 3.38. Patrón para la definición de la asociación AS-09

AS-09			
Requisitos	<ul style="list-style-type: none"> RA-02: Bien inmueble 		
Descripción	Las clases CL-02 y CL-03 se relacionan mediante esta asociación que representa la tipología del bien o el conjunto de tipologías en los que se enmarca el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-02	Tipología	0..n
	CL-03	Bien inmueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-02.Tipología 		
Estado	básico		

Tabla 3.39. Patrón para la definición de la asociación AS-10

AS-10			
Requisitos	<ul style="list-style-type: none"> RA-02: Bien inmueble 		
Descripción	Las clases CL-02 y CL-03 se relacionan mediante esta asociación que representa los estilos artísticos en los que se registra el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-02	Estilo	0..n
	CL-03	Bien inmueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-02.Estilo 		
Estado	básico		

Tabla 3.40. Patrón para la definición de la asociación AS-11

AS-11			
Requisitos	<ul style="list-style-type: none"> RA-02: Bien inmueble 		
Descripción	Las clases CL-02 y CL-03 se relacionan mediante esta asociación que representa la lista de períodos históricos en los que se cataloga el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-02	Período Histórico	0..n
	CL-03	Bien inmueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-02.Período Histórico 		
Estado	básico		

Tabla 3.41. Patrón para la definición de la asociación AS-12

AS-12			
Requisitos	<ul style="list-style-type: none"> RA-02: Bien inmueble 		
Descripción	Las clases CL-02 y CL-03 se relacionan mediante esta asociación que representa el conjunto de actividades para los que se supone que era utilizado el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-02	Actividad	0..n
	CL-03	Bien inmueble	0..n
Datos específicos	<ul style="list-style-type: none"> RA-02.Actividad 		
Estado	básico		

4. Las dos últimas asociaciones, correspondientes a la relación que se establece entre los datos de tesauo, sus sinónimos y sus antónimos. Surgen de cada uno de ellas dos asociaciones unidireccionales porque no existe información suficiente en RA-03 para determinarlas como bidireccionales. Las asociaciones correspondientes son AS-13 y AS-14 y se muestran en las tablas 4.42 y 4.43 respectivamente.

Tabla 3.42. Patrón para la definición de la asociación AS-13

AS-13			
Requisitos	<ul style="list-style-type: none"> RA-03: Dato de tesauo 		
Descripción	Las clases CL-03 y CL-03 se relacionan mediante esta asociación que representa la lista de términos sinónimos que también se encuentran en el tesauo.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-03	Dato de tesauo	0..n
	CL-03	Sinónimos	0..n
Datos específicos	<ul style="list-style-type: none"> RA-03.Sinónimos 		
Estado	básico		

Tabla 3.43. Patrón para la definición de la asociación AS-14

AS-14			
Requisitos	<ul style="list-style-type: none"> RA-03.Dato de tesoro 		
Descripción	Las clases CL-03 y CL-03 se relacionan mediante esta asociación que representa la lista de los términos que describen el mismo concepto pero de una manera más concreta.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-03	Dato de tesoro	0..n
	CL-03	Específicos	0..n
Datos específicos	<ul style="list-style-type: none"> RA-03.Específicos 		
Estado	básico		

Todas éstas se corresponden con el modelo conceptual básico derivado de los requisitos de almacenamiento de información, pero hay que recordar que en el ejemplo también existen naturalezas. Cada una de las tres naturalezas genera una clase en el modelo conceptual básico, que son las que se representan en las tablas 3.44, correspondiente a la naturaleza NA-01, 3.45 correspondiente a la naturaleza NA-02, y 3.46 correspondiente a la naturaleza NA-03.

Tabla 3.44. Patrón para la definición de la clase CLn-01

CLn-01	Código Bien Mueble		
Naturaleza	NA-01.Código Bien Mueble		
Descripción	El sistema deberá almacenar la información correspondiente a la estructura que describe el código unívoco identificativo de cada bien mueble otorgado por la Dirección General de Bienes. En concreto:		
Atributos	Descripción	Significado	Dato
	Bien inmueble: CLn-02	Recoge una referencia al bien inmueble en el que fue hallado el bien mueble.	NA-01.Bien inmueble
	Mueble: Cadena {Tamaño: 3}	Es una cadena de tres dígitos que guarda el número particular que tiene ese mueble en el inmueble.	NA-01.Mueble
Estado	Básico		
Comentarios	Los datos se presentan mediante un código numérico de 12 dígitos. Los 9 primeros se corresponden con el código del bien inmueble, luego se acompaña con un punto, seguido de los tres dígitos correspondientes al código del mueble: XXXXXXXXXX.XXX		

Tabla 3.45. Patrón para la definición de la clase CLn-02

CLn-02	Código Bien Inmueble		
Naturaleza	NA-02.Código Bien Inmueble		
Descripción	El sistema deberá almacenar la información correspondiente a la estructura que describe el código unívoco identificativo de cada bien inmueble otorgado por la Dirección General de Bienes culturales. En concreto:		
Atributos	Descripción	Significado	Dato específico
	Provincia: Cadena {Tamaño:2}	Almacena, mediante un código de dos números, la provincia en la que se encuentra el monumento.	NA-02.Provincia
	Municipio: Cadena {Tamaño:3}	Almacena, mediante un código de tres números, el municipio de la provincia en el que se ubica el monumento	NA-02.Municipio
	Inmueble: Cadena {Tamaño: 4}	Es una cadena de cuatro dígitos que guarda el número particular que tiene ese inmueble en el municipio.	NA-02.Inmueble
Estado	Básico		
Comentarios	Los datos se presentan mediante un código numérico de 9 dígitos. Los 2 primeros se refieren a la provincia, los tres siguiente al código del municipio y los cuatro siguientes al código de inmueble. Todos aparecen como una cadena continua: XXXXXXXXXX.		

Tabla 3.46. Patrón para la definición de la clase CLn-03

CLn-03	Coordenada UTM		
Naturaleza	NA-03.Coordenada UTM		
Descripción	El sistema deberá almacenar la información correspondiente a la estructura que deben tener las coordenadas X e Y que delimitan cada monumento en el plano.		
Atributos	Descripción	Significado	Dato específico
	Coordenada X: Real	representa el valor de la coordenada X	NA-03. Coordenada X
	Coordenada Y: Real	representa el valor de la coordenada Y	NA-03. Coordenada Y
Estado	Básico		

Con estas definiciones, el diagrama de clases conceptuales queda tal y como se muestra en la figura 3.3. Se ha optado por dividir el diagrama en dos paquetes: el

correspondiente a los requisitos de almacenamiento, denominado *clases del sistema*, y el correspondiente a las naturalezas, denominado *nuevas naturalezas*. Como los atributos de cada clase se han especificado en los patrones anteriores, se han omitido del modelo para simplificarlo. Nótese también que los nombres en el modelo conceptual básico no siguen la nomenclatura propia de la orientación a objetos. Esto se debe a que vienen heredados directamente de los requisitos. Además las asociaciones aparecen sin nombres. Corregir todos estos errores sintácticos de la derivación es tarea, como se ve en el siguiente apartado, de la generación del modelo conceptual final.

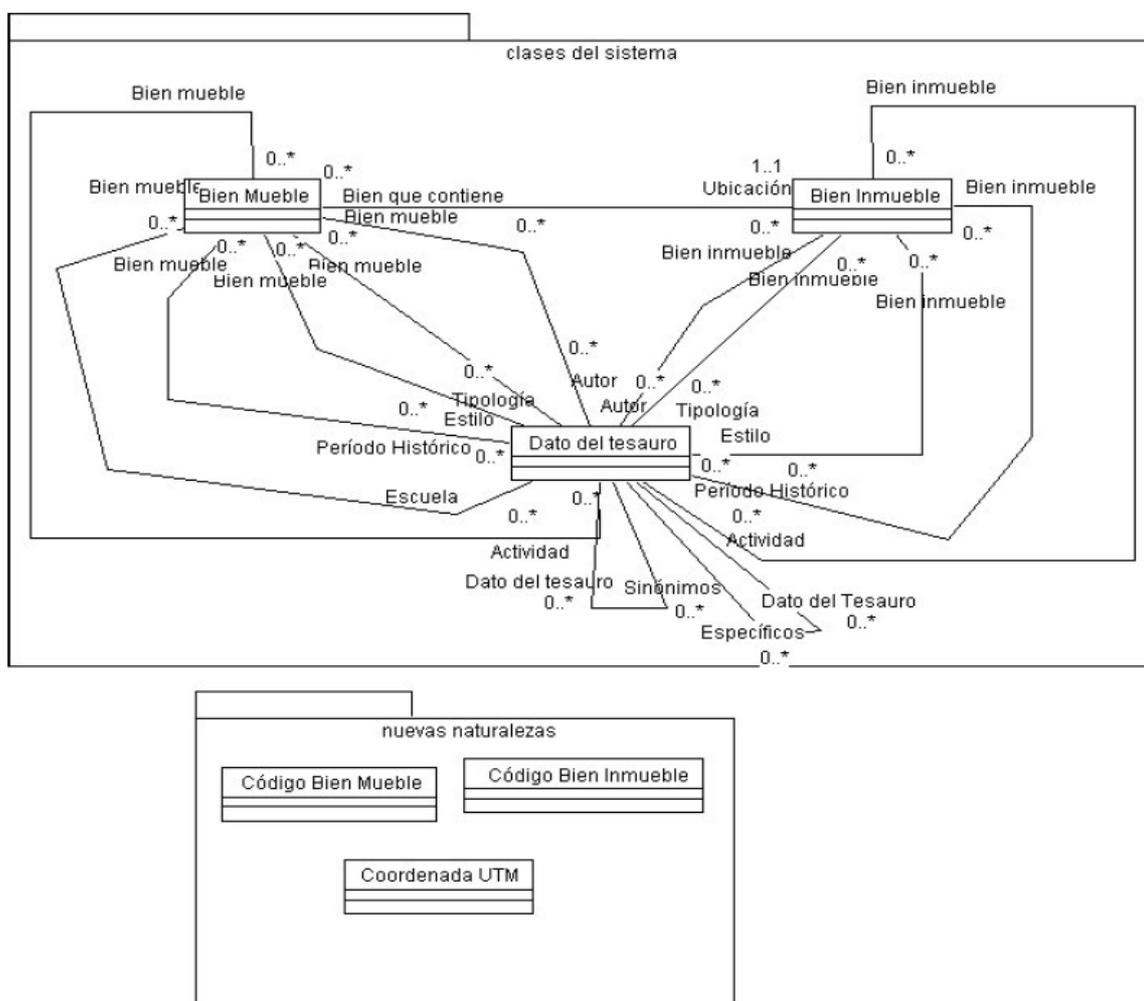


Figura 3.3. Modelo conceptual básico

Habría, por tanto que definir dos patrones más para los paquetes definidos. Éstos son los que se muestran en las tablas 3.47 y 3.48.

Tabla 3.47. Patrón para la definición del paquete PQ-01

PQ-01	clases del sistema
Descripción	Recoge las clases derivadas de requisitos de almacenamiento de información.

Tabla 3.48. Patrón para la definición del paquete PQ-02

PQ-02	Nuevas naturalezas
Descripción	Recoge las clase derivadas de las naturalezas del sistema

Con todo esto, ahora lo que corresponde es comenzar a revisar lo que se ha obtenido de manera sistemática y conseguir así el modelo conceptual final. Para ello, las revisiones y comprobaciones a realizar se enumeran en los siguientes pasos:

Paso 1-Revisar las clases

Paso 2-Revisar los atributos

Paso 3-Revisar las asociaciones

Paso 4- Detectar la herencia

Paso 5- Detectar clases asociación y las asociaciones calificadas

Paso 6- Revisar nombres y descripciones en los patrones

Paso 7- Revisar paquetes

La revisión de clases, atributos y asociaciones no produce ningún cambio en el modelo conceptual básico. Los cambios que habría que realizar en este respecto se corresponderían con la revisión de nombres y descripciones que hay que hacer en el paso 6. Sin embargo, en el paso 4 sí que se detecta un cambio. Si se observa el diagrama de clases básico, se puede ver que las clases *Bien mueble* y *Bien inmueble*, comparten mucha información y además, participan en muchas asociaciones similares, se podría pues establecer una clase padre, que se podría llamar *Bien*, para agrupar la información común. Esto requeriría además que se reestructurara el diagrama aprovechando la estructura jerárquica, tal y como se muestra en la figura 3.4.

En el paso 6 también hay que retocar el modelo conceptual básico, puesto que, además de nombrar a los elementos siguiendo la nomenclatura de la notación

orientada a objetos (por ejemplo la clase Bien mueble pasa a llamarse BienMueble, los roles se nombran con minúsculas y los atributos también), hay que poner nombres a las asociaciones que se han derivado.

Con respecto al último paso, no hay que hacer nada. Terminada la revisión, el diagrama conceptual final queda tal y como se muestra en la figura 3.4. Sólo se ha puesto en la figura el paquete de *clases del sistema* puesto que el de *nuevas naturalezas* no ha sufrido cambio.

El cambio de introducir la estructura jerárquica también conlleva cambios en los patrones, como era de esperar. Por lo pronto, hay que añadir una clase más *Bien*, cuyo código es CL-04 y se encuentra en la tabla 3.49. Y además hay que retocar los patrones correspondientes a las clases CL-01 y CL-02. De éstas se eliminan los atributos comunes que pasan a la clase padre. Además, se le añade el campo *Hereda de* para recoger la descripción de la estructura jerárquica. Se corresponden respectivamente con los patrones de las tablas 3.50 y la tabla 3.51.

Los patrones de las otras clases, ya sean del paquete *clases del sistema* o del paquete *nuevas naturalezas*, no se ven afectados por el cambio. Únicamente cambiaría el nombre para seguir la nomenclatura orientada a objeto (por ejemplo, *Dato del tesouro* pasa a ser *DatoDelTesouro*), por lo que no se va a repetir el patrón.

Tabla 3.49. Patrón para la definición de la clase CL-04

CL-04	Bien		
Descripción	El sistema deberá almacenar la información correspondiente a los bienes que se encuentran en el sistema. En concreto:		
Atributos	Descripción	Significado	Dato
	Nombre: Cadena	Almacena información sobre el nombre que identifica al bien	RA-01.Nombre RA-02.Nombre
	Descripción: Documento	Recoge una descripción detallada del bien	RA-01.Descripción RA-02.Descripción
	Cronología: Fecha {formato: aaaa-aaaa}	Establece los períodos sobre los que se ubica la creación del bien	RA-01.Cronología RA-02.Cronología
	Imagen [0..n]: Imagen	Almacena una galería de imágenes del centro gráfico donde aparece el bien	RA-01.Imagen RA-02.iMAGEN
Estado	Final		

Tabla 3.51. Patrón para la definición de la clase CL-02

CL-02	Bien inmueble		
Requisito	RA-02.Bien inmueble		
Hereda de	CL-04.Bien		
Descripción	El sistema deberá almacenar la información correspondiente a los bienes inmuebles que se encuentran en el sistema. En concreto:		
Atributos	Descripción	Significado	Dato
	Código bien: CLn-02	Almacena información sobre el código que identifica de manera unívoca a cada bien	RA-2.Código
	Coordenadas utm: Cln-03	Recoge las coordenadas X e Y en las que se ubica el bien. Servirán para localizar los bienes en el mapa de la localidad.	RA-02.Coordenadas
Estado	Final		

Con respecto a las asociaciones los cambios que existen se refieren a los nombres, a la nomenclatura de los nombres de roles o a los que vienen derivados de la herencia. Se van detallando una a una.

1. La asociación AS-01 asume el nombre *seUbica*, en el resto, su patrón no cambia.
2. La asociación AS-02 y AS-08 se funden en una única con el nombre *esFabricadoPor* que asume el código AS-02 y que se presenta en el patrón de la tabla 3.52. Además, se han modificado sus nombres de roles.
3. La asociación AS-03 y AS-09 se funden también en una que recibe el nombre de *es*. Queda recogida en el patrón de la tabla 3.53. También se han modificado los roles y la descripción.

Tabla 3.52. Patrón para la definición de la asociación AS-02

AS-02	esFabricadoPor		
Requisitos	<ul style="list-style-type: none"> RA-02: Bien mueble 		
Descripción	Las clases CL-04 y CL-03 se relacionan mediante esta asociación que representa una lista de autores que o bien han fabricado el bien o ha participado en su construcción.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-02	autor	0..n
	CL-03		0..n
Datos	<ul style="list-style-type: none"> RA-01.Autor RA-02.Autor 		
Estado	final		

Tabla 3.53. Patrón para la definición de la asociación AS-03

AS-03	es		
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-04 y CL-03 se relacionan mediante esta asociación que representa la tipología del bien o el conjunto de tipologías en los que se enmarca el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-04	tipología	0..n
	CL-03		0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Tipología RA-02.Tipología 		
Estado	final		

4. Lo mismo ocurre con las asociaciones AS-04 y AS-10 que asumen ahora el identificador AS-04 y cuyo patrón se muestra en la tabla 3.54.

Tabla 3.54. Patrón para la definición de la asociación AS-04

AS-04	tiene		
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble RA-02: Bien inmueble 		
Descripción	Las clases CL-04 y CL-03 se relacionan mediante esta asociación que representa los estilos artísticos en los que se registra el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-04	estilo	0..n
	CL-03		0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Estilo 	<ul style="list-style-type: none"> RA-02.Estilo 	
Estado	final		

5. Con las asociaciones AS-05 y AS-11 ocurre igual. En este caso se funden en la asociación AS-05 y toma el nombre *seSitúa*. Su patrón se muestra en la tabla 3.55.

Tabla 3.55. Patrón para la definición de la asociación AS-05

AS-05	seSitúa		
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble RA-02: Bien inmueble 		
Descripción	Las clases CL-04 y CL-03 se relacionan mediante esta asociación que representa la lista de períodos históricos en los que se cataloga el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de	Cardinalidad
	CL-04	períodoHistóri	0..n
	CL-03		0..n
Datos	<ul style="list-style-type: none"> RA-01.Período Histórico 		
Estado	final		

6. La asociación AS-06 correspondiente a la Escuela queda igual. Aunque se le asignaría un nombre llamándose *pertenece* y se cambiaría la nomenclatura en los nombres de roles. No merece la pena volver a indicar cómo queda el patrón para este cambio.

7. Las asociaciones AS-07 y AS-12 se fusionan como en casos anteriores, asumiendo el nombre *seUsaba*. Su patrón se encuentra en la tabla 3.56. También aquí se han retocado los nombres de roles y la nomenclatura.

8- Con las asociaciones AS-05 y AS-11 ocurre igual. En este caso se funden en la asociación AS-05 y toma el nombre *seSitúa*. Su patrón se muestra en la tabla 3.55.

Tabla 3.56. Patrón para la definición de la asociación AS-07

AS-07	seUsaba		
Requisitos	<ul style="list-style-type: none"> RA-01: Bien mueble 		
Descripción	Las clases CL-04 y CL-03 se relacionan mediante esta asociación que representa el conjunto de actividades para los que se supone que era utilizado el bien.		
Tipo	Unidireccional		
Clases	Nombre de la clase	Nombre de Rol	Cardinalidad
	CL-04	actividad	0..n
	CL-03		0..n
Datos específicos	<ul style="list-style-type: none"> RA-01.Actividad • RA-02.Actividad 		
Estado	final		

9. Las asociaciones AS-13 y AS-14 no cambian su patrón más que en la notación de sus roles aunque, para facilitar la codificación podría ser interesante identificarlas como AS-08 y AS-09.

Con todo esto, queda terminado el modelo conceptual final. Los patrones del modelo y el diagrama deben ser incorporados al documento de análisis del sistema siguiendo el índice que marca NDT y que se recuerda en siguientes actividades.

Realizar el modelo de navegación

La siguiente actividad de NDT consiste en generar el modelo de navegación. Para ello, lo primero a realizar es la definición de los actores en estudio del sistema. Teniendo en cuenta el algoritmo que propone NDT, se tiene:

1. Se define AE como el conjunto vacío
2. Cada actor del sistema que no herede de ningún otro actor, se incluye en el conjunto AE como un actor en estudio. Con esto, AC-01 pasa al grupo de actores AE-01 y AC-02 al de AE-02.
3. 3- De los actores en estudio que sí heredan de otro actor, éstos son en el ejemplo AC-03, AC-04 y AC-05:
 - a) Si existe un prototipo que sea accesible por el actor hijo y no por el actor padre, se añade el actor hijo al conjunto de actores en estudio como un actor independiente. No es el caso de ninguno de ellos.

- b) Si no, el actor hijo pasa a ser representado por el actor en estudio que representa a su padre. Es el caso de los tres, por lo que todos pasan a ser parte de AE-02.

En resumen, la definición de actores en estudio es la que se muestra en la tabla 3.57.

Tabla 3.57. Matriz de descripción de actores en estudio

ACTOR	AC-01	AC-02	AC-03	AC-04	AC-05
AE-01	^				
AE-02		^	^	^	^

Con esta definición se concluye que hay que desarrollar dos modelos de navegación: uno para el actor en estudio AE-01 o, lo que es lo mismo, los turistas, y otro para el actor en estudio AE-02, o lo que es lo mismo, cualquier tipo de investigador.

Realmente, el proceso para generar el modelo de navegación de los investigadores no aporta nada al estudiar el proceso de generación del modelo de navegación de los turistas. Por ello, sólo se va a estudiar este modelo. Este anexo no tiene por objetivo mostrar los resultados, como se ha comentado, la idea es ofrecer una visión práctica de NDT, en la que se muestre el proceso de aplicación de NDT y sus técnicas. Así que desarrollar aquí el modelo de navegación de los investigadores, que no aporta nada nuevo, no tiene sentido.

Centrando el trabajo pues en el modelo de navegación de los turistas, se puede obtener desde el resultado de la ingeniería de requisitos que tiene asociados dos prototipos de visualización: PV-01 y PV-02. Cada uno de ellos va a generar un nodo en el modelo de navegación básico: NO-01 y NO-02. El patrón correspondiente al primero de ellos se muestra en la tabla 3.58.

Puede verse que los campos se rellenan a partir de los datos recogidos en el patrón del prototipo PV-01. Así, la descripción, las operaciones o los atributos se asumen directamente desde el patrón PV-01.

El nodo NO-02, que se genera a partir del prototipo PV-02, se muestra en la tabla 3.59.

Tabla 3.58. Patrón para la definición del nodo NO-01

NO-01	Ficha del Bien Mueble para turistas
Prototipo	<ul style="list-style-type: none"> PV-01: Ficha del Bien Mueble para turistas
Actores en estudio	<ul style="list-style-type: none"> AE-01
Descripción	El sistema deberá mostrar la información correspondiente a los datos concretos que se muestran y a la navegación expresada que representa la información mostrada al turista sobre los bienes muebles. En concreto:
Atributos	Descripción
	RA-01.Nombre
	RA-01.Descripción
	RA-01.Ubicación
	RA-01.Cronología
	RA-01.Imagen
	RA-01.Historia de la pieza
	RA-01.Autor.Término
	RA-01.Tipología.Término
	RA-01.Estilo.Término
	RA-01.Período Histórico.Término
RA-01.Escuela.Término	
RA-01.Actividad.Término	
Estado	básico

Tabla 3.59. Patrón para la definición del nodo NO-02

NO-02	Ficha del Bien Inmueble para turistas
Prototipo	<ul style="list-style-type: none"> PV-02: Ficha del Bien Inmueble para turistas
Actores en estudio	<ul style="list-style-type: none"> AE-01
Descripción	El sistema deberá mostrar la información correspondiente a los datos concretos que se muestran y a la navegación expresada que representa la información mostrada al turista sobre los bienes inmuebles. En concreto:
Atributos	Descripción
	RA-02.Nombre
	RA-02.Descripción
	RA-02.Cronología
	RA-02.Imagen
	RA-02.Autor.Término
	RA-02.Tipología.Término
	RA-02.Estilo.Término
	RA-02.Período Histórico.Término
RA-02.Actividad.Término	
Estado	básico

El siguiente paso consiste en generar los enlaces directos. Para ello, se estudian los prototipos PV-01 y PV-02. Desde PV-01 hay un prototipo de salida a PV-02 que no es múltiple, por lo que se genera un enlace, EN-01, que va de PV-01 a PV-02. En PV-02 no hay prototipos de salida no múltiples. El patrón para describir el enlace

EN-01 se muestra en la tabla 3.60. Nótese que el nombre del enlace no se especifica desde el modelo básico.

Tabla 3.60. Patrón para describir el enlace EN-01

EN-01	
Origen	NO-01: Ficha del Bien Mueble para turistas
Destino	NO-02: Ficha del Bien Inmueble para turistas
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos N0-01 y N0-02
Tipo	bidireccional
Estado	Básico

Tras esto deben generarse los índices. Como hay un prototipo de salida múltiple desde PV-02 a PV-01 hay que colocar un índice entre ambos que permita al usuario seleccionar la opción. Para ello, se genera el índice IN-01 cuyo patrón se muestra en la tabla 3.61. Al igual que en el enlace, no se especifica el nombre del índice en el modelo básico. Tampoco la descripción se completa en la derivación del modelo básico.

Tabla 3.61. Patrón para describir el índice IN-01

IN-01	
Destino	NO-X: Ficha del Bien Mueble para turistas
Actores en estudio	• AE-01
Descripción	
Tipo	índice
Estado	básico

El siguiente paso, consiste en estudiar las queries necesarias. Como tanto el prototipo PV-01 como el PV-02 tienen frases asociadas hay que crear dos queries, QU-01 y QU-02, cuyos patrones respectivamente son los mostrados en las tablas 3.62 y 3.63.

Esta primera query es la que da entrada al nodo NO-01. Nótese que el cuerpo viene derivado directamente del cuerpo de la frase FR-01. De la misma forma, la query de la tabla 3.63 es la que da entrada al nodo NO-02 y su cuerpo coincide con el de la frase FR-02. En ninguna de las dos queries se ha podido definir el nombre de manera sistemática. Esta tarea, como se contempla más adelante se realiza durante la elaboración del modelo final.

Tabla 3.62. Patrón para describir la query QU-01

QU-01												
Destino	NO-01.Ficha del Bien Mueble para turistas											
Descripción	El sistema deberá permitir obtener información desde el usuario a partir de las frases que se muestran a continuación:											
	<table border="1"> <thead> <tr> <th>Cuerpo</th> <th>Actores</th> </tr> </thead> <tbody> <tr> <td>El concepto RA-02.Código bien debe ser exactamente</td> <td rowspan="8">• AE-01</td> </tr> <tr> <td>El concepto RA-01.Nombre contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-01.Autor.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-01.Tipología.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-01.Estilo.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-01.Período Histórico.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-01.Escuela.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-01.Actividad.Término debe contener la siguiente cadena</td> </tr> </tbody> </table>	Cuerpo	Actores	El concepto RA-02.Código bien debe ser exactamente	• AE-01	El concepto RA-01.Nombre contener la siguiente cadena	El concepto RA-01.Autor.Término debe contener la siguiente cadena	El concepto RA-01.Tipología.Término debe contener la siguiente cadena	El concepto RA-01.Estilo.Término debe contener la siguiente cadena	El concepto RA-01.Período Histórico.Término debe contener la siguiente cadena	El concepto RA-01.Escuela.Término debe contener la siguiente cadena	El concepto RA-01.Actividad.Término debe contener la siguiente cadena
Cuerpo	Actores											
El concepto RA-02.Código bien debe ser exactamente	• AE-01											
El concepto RA-01.Nombre contener la siguiente cadena												
El concepto RA-01.Autor.Término debe contener la siguiente cadena												
El concepto RA-01.Tipología.Término debe contener la siguiente cadena												
El concepto RA-01.Estilo.Término debe contener la siguiente cadena												
El concepto RA-01.Período Histórico.Término debe contener la siguiente cadena												
El concepto RA-01.Escuela.Término debe contener la siguiente cadena												
El concepto RA-01.Actividad.Término debe contener la siguiente cadena												
Estado	básico											

Tabla 3.63. Patrón para describir la query QU-02

QU-02											
Destino	NO-01.Ficha del Bien Inmueble para turistas										
Descripción	El sistema deberá permitir obtener información desde el usuario a partir de la las frases que se muestran a continuación:										
	<table border="1"> <thead> <tr> <th>Cuerpo</th> <th>Actores</th> </tr> </thead> <tbody> <tr> <td>El concepto RA-02.Código bien debe ser exactamente</td> <td rowspan="7">• AE-01</td> </tr> <tr> <td>El concepto RA-02.Nombre contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-02.Autor.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-02.Tipología.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-02.Estilo.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-02.Período Histórico.Término debe contener la siguiente cadena</td> </tr> <tr> <td>El concepto RA-02.Actividad.Término debe contener la siguiente cadena</td> </tr> </tbody> </table>	Cuerpo	Actores	El concepto RA-02.Código bien debe ser exactamente	• AE-01	El concepto RA-02.Nombre contener la siguiente cadena	El concepto RA-02.Autor.Término debe contener la siguiente cadena	El concepto RA-02.Tipología.Término debe contener la siguiente cadena	El concepto RA-02.Estilo.Término debe contener la siguiente cadena	El concepto RA-02.Período Histórico.Término debe contener la siguiente cadena	El concepto RA-02.Actividad.Término debe contener la siguiente cadena
Cuerpo	Actores										
El concepto RA-02.Código bien debe ser exactamente	• AE-01										
El concepto RA-02.Nombre contener la siguiente cadena											
El concepto RA-02.Autor.Término debe contener la siguiente cadena											
El concepto RA-02.Tipología.Término debe contener la siguiente cadena											
El concepto RA-02.Estilo.Término debe contener la siguiente cadena											
El concepto RA-02.Período Histórico.Término debe contener la siguiente cadena											
El concepto RA-02.Actividad.Término debe contener la siguiente cadena											
Estado	básico										

La creación de esta query y del índice anterior conlleva ciertos cambios en cuanto a los enlaces.

1- Hay que crear un enlace de QU-01 a NO-01, cuyo patrón se muestra en la tabla 3.64, y otro de QU-02 a NO-02, cuyo patrón se muestra en la tabla 3.64.

Tabla 3.64. Patrón para describir el enlace EN-02

EN-02	
Origen	QU-01
Destino	NO-01: Ficha del Bien Mueble para turistas
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos QU-01 y NO-
Tipo	bidireccional
Estado	básico

Tabla 3.65. Patrón para describir el enlace EN-03

EN-03	
Origen	QU-02
Destino	NO-02: Ficha del Bien Inmueble para turistas
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos QU-02 y
Tipo	bidireccional
Estado	básico

2- El enlace EN-01 hay que modificarlo puesto que antes de llegar al nodo NO-02, como éste tiene asociado una query, tiene que derivarse a la query. El enlace se modifica en su destino y su patrón queda modificado como se muestra en la tabla 3.66.

Tabla 3.66. Patrón para describir el enlace EN-01

EN-01	
Origen	NO-01: Ficha del Bien Mueble para turistas
Destino	QU-02
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos NO-01 y QU-02
Tipo	bidireccional
Estado	básico

3-Al definir el índice IN-01 hay que definir un enlace que va desde NO-02 al índice y otro que permita navegar desde el índice a NO-01. Ambos se describen mediante los patrones que se muestran respectivamente en las tablas 3.67 y 3.68.

Tabla 3.67. Patrón para describir el enlace EN-04

EN-04	
Origen	NO-02: Ficha del Bien Inmueble para turistas
Destino	IN-01
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos NO-02 y IN-01
Tipo	bidireccional
Estado	básico

Tabla 3.68. Patrón para describir el enlace EN-05

EN-05	
Origen	IN-01
Destino	NO-01: Ficha del Bien Mueble para turistas
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos IN-01 y NO-01
Tipo	bidireccional
Estado	básico

Con todo esto sólo habría que estudiar la necesidad de ver si son necesarios menús en el sistema. Para ello se define el grafo navegacional del sistema. Los nodos, índices y querys son los vértices del grafo y los enlaces las aristas. El grafo quedaría como se muestra en la figura 3.5.

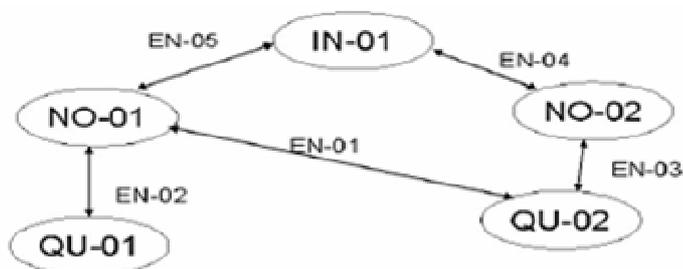


Figura 3.5- Grafo navegacional para los turistas

Como puede verse, el grafo es conexo por lo que no habría que definir, según las reglas de NDT para generar el modelo de navegación básico, ningún menú. Con todo

esto, se tendría generado el modelo de navegación básico. El diagrama se ve a continuación pues como se estudia no va a modificarse el modelo de navegación básico más que en rellenar aspectos como el nombre de las queries, el completar las descripciones, etc.

Según NDT, las revisiones a realizar para llegar al modelo de navegación final son las que se marcan en los siguientes pasos:

Paso 1- Revisar los nodos

Paso 2- Revisar los índices y rutas guiadas

Paso 3- Revisar las queries

Paso 4- Revisar los menús

Paso 5- Revisar los enlaces

Paso 6- Revisar nombres y descripciones en los patrones

Paso 7- Revisar la calidad del modelo

Tras la realización de los tres primeros pasos, no existen modificaciones a realizar. Quizás podría ser interesante definir el índice IN-01 como una ruta guiada para que los bienes aparezcan ordenados. En cuanto a la revisión de los menús, parece interesante, a pesar de que no es estrictamente necesario, definir un menú principal de entrada que permita ir bien a los datos del bien mueble bien a los datos del bien inmueble.

Si se opta por definirlo, quedaría como se muestra en el patrón de la tabla 3.69. Nótese que el menú lleva a las queries que dan entrada a los nodos que muestran los datos de los bienes muebles y de los inmuebles.

Tabla 3.69. Patrón para describir el menú principal

ME-01	Menú Principal
Destinos	<ul style="list-style-type: none"> • QU-01 • QU-02
Actores en estudio	<ul style="list-style-type: none"> • AE-01
Estado	final

La definición del menú obliga a definir dos nuevos enlaces que lleven desde el menú a cada una de las queries. Su definición se muestra en las tablas 3.70 y 3.71.

Tabla 3.70. Patrón para describir el enlace EN-06

EN-06	EnlaceDesdeMenúAQU01
Origen	ME-01: Menú Principal
Destino	QU-01
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos ME-01 y QU-01
Tipo	bidireccional
Estado	final

Tabla 3.71. Patrón para describir el enlace EN-05

EN-07	EnlaceDesdeMenúAQU02
Origen	ME-01: Menú Principal
Destino	QU-02
Actores en estudio	• AE-01
Descripción	El sistema debe permitir la navegación entre los elementos ME-01 y QU-02
Tipo	bidireccional
Estado	final

En el siguiente paso, en el que se revisan los nombres y descripciones es en el que más hay que trabajar. En concreto hay que generar los nombres y descripciones de todos los enlaces, índices y queries generadas con el modelo básico. Además hay que revisar la descripción de los nodos.

Con respecto al último paso, NDT propone estudiar:

- 1- No debe existir ningún vértice aislado.
- 2- No deben existir puntos de no retorno.
- 3- Todos los puntos de la navegación deben ser alcanzables desde cualquier otro punto.

En este ejemplo, las tres premisas se cumplen así que no hay que realizar nada.

Por tanto, con los cambios realizados en el modelo de navegación final el diagrama de clases para los turistas queda como se muestra en la figura 3.6.

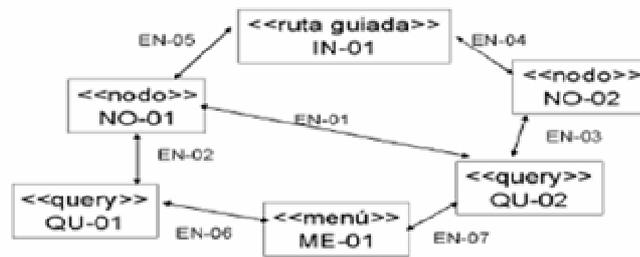


Figura 3.6. Modelo de navegación final para los turistas

Realmente en el modelo de navegación no deben aparecer los códigos de las clases de navegación sino los nombres. Sin embargo, para que resulte más sencillo de entender y debido a que el objetivo del anexo es dar a entender cómo se aplican las técnicas de NDT y no el resultado, se han representado los códigos que son con lo que se ha trabajado hasta ahora.

Realizar el conjunto de prototipos

A lo largo del capítulo, cuando se han presentado las técnicas de prototipado de NDT, se han ido presentando ejemplos concretos, así que volver a indicar lo mismo en este apartado no tiene sentido.

Sí que cabe decir que, en este caso, resulta interesante trabajar un poco los prototipos básicos que se generan mediante el proceso estudiado pues los usuarios y clientes finales no son expertos en informática y presentarles unos prototipos algo más avanzados ayuda en gran medida a facilitarles la validación.

Generar el documento de análisis del sistema.

Igual que en documento de requisitos, la generación del documento de análisis del sistema no tiene más problema que copiar los modelos y patrones resultantes de todo el proceso. No se presenta en este anexo pues no es su objetivo.

3.2. Ejemplo de MULTIMET

Introducción

El departamento de inglés del Instituto Superior Minero Metalúrgico de Moa no cuenta con un sistema de aprendizaje que ayude al alumno a ejercitar los

conocimientos teóricos adquiridos durante la asignatura de inglés. Se posee una gran cantidad de ejercicios de gramática y pronunciación, así como un sistema de evaluación que no está digitalizado. Es necesario crear una multimedia que ayude al estudiante a ejercitarse, y a autoevaluarse en el idioma inglés, además de que esta ofrecerá un ambiente que le permitirá a cualquier usuario interesarse por este idioma. El sistema deberá tener dos opciones, aprender inglés y un test de evaluación del nivel. Dentro de la opción aprender inglés tendremos otras opciones como son: diálogo, ejercicios de pronunciación, gramaticales, traducción de diálogo, evaluación final, opciones del sistema y ayuda. La multimedia permitirá la entrada de cualquier usuario, por lo que cualquier persona interesada que no sea del centro también puede ejercitarse en inglés con aplicación.

- **Definición del Producto.**

Necesidades del usuario y objetivos del producto.

El producto surge por la necesidad de un medio para la práctica del idioma inglés que le permita al cliente realizar ejercicios y evaluarse para saber su nivel de conocimiento en este idioma. La información que tenemos no está digitalizada

Problema: La inexistencia de un medio de enseñanza que te permita poner en práctica los conocimientos de cada cliente en el idioma inglés.

- **Tecnología necesaria para la ejecución del producto.**

Software: Sistema Operativo Windows XP Profesional.

Hardware: Pentium II con 128 MB de RAM y un microprocesador a 300 MHz, 6 Gb de disco duro.

Muchos programas de computadoras sólo incluyen como sonidos el silencio y los ruidos de la operación de la máquina. Puede ser una limitación del equipo de computación, si no tiene facilidad sonora. Para esto es necesario un equipo que brinde esta facilidad para poder crear la multimedia.

- **Definición de los objetivos de la aplicación.**

La multimedia es educativa, por lo que los objetivos de la aplicación son los siguientes:

- Insertar nombre de usuario para luego darle una evaluación final.
- La aplicación deberá mostrar diálogos para que el usuario los escuche y dependiendo de estos haga los ejercicios gramaticales y los de pronunciación.
- Permitirá deberá traducir los diálogos que tiene.
- EL sistema te mostrará un resumen de la evaluación que obtuviste durante la utilización del medio.
- Tendrá una opción que permitirá al usuario examinarse dependiendo de su nivel(graduado escolar, bachillerato y estudios superiores), que te mostrará los resultados dependiendo de las respuestas que hayas respondido correcta o incorrectamente.
- Mejorar el proceso de estudio al estudiante, brindándole la facilidad de acceso a un sistema que los ayude a aprender mejor practicando.
- Elevar los conocimientos en los estudiantes y otras personas interesadas en el tema.

- **Identificación de la audiencia**

Esta aplicación va dirigida principalmente a estudiantes del Instituto Superior Minero Metalúrgico de Moa, pero el sistema podrá ser usado por cualquier persona que posea conocimientos básicos de inglés y el manejo de la computadora.

La interfaz debe ser agradable y atractiva para conseguir el interés de los usuarios en la utilización de la aplicación. Se debe tener en cuenta algunos elementos de diseño como, utilizar colores sencillos y frescos en el diseño de la interfaz de la aplicación, una pantalla que te permita entrar tu nombre para la evaluación final de tus conocimientos. Utilización de imágenes que lleven texto y ayuden a los usuarios a familiarizarse con los temas dentro de la aplicación.

- **Especificación del contenido.**

Entre los temas que serán tratados en la aplicación están:

Temas:

1. Ejercitando el inglés

1. Presentación del Diálogo
2. Presentación Ejercicios de gramática

3. Presentación Ejercicios de pronunciación

2. Evaluación de su nivel

Test de evaluación

Definición de los medios y sus objetivos.

Tema	Medio	Objetivo	Disponible	Fuente
1-Ejercita inglés -Diálogo -Traducción -Ejercicios gramt -Ejercicios Pron	1-Texto, imagen	Enseñar	Si	Facilitadas por el departamento de inglés. Formatos <u>Imagen:</u> .JPEG <u>Sonido:</u> .Wma <u>Videos</u> .AVI <u>Texto</u> En la aplicación.
	-Video, sonido, texto	Enseñar	Si	
	- texto	Enseñar		
	- texto, imagen, sonido	Enseñar		
	- texto, imagen, sonido	Enseñar		
2- Evaluación de su nivel. -Test de evaluac.	2- Texto, imagen	Enseñar	Si	
	- Pagina HTML	Enseñar	Si	

▪ **Normas de diseño.**

En las aplicaciones multimedia se utilizan varios tipos de información como son: textos, imágenes, videos y sonidos para mantener una buena uniformidad estos medios tienen la siguiente forma:

Textos:

Tamaño de fuente: Arial 12

Porcentaje máximo de ocupación de pantallas: 15px.

Imágenes:

Fuentes utilizadas para títulos de selección de los temas y subtemas.

Tamaño máximo: 260px x 260px

Tamaño mínimo: 128px x 128px

Videos

Duración de cada video de diálogos: máximo (3min), mínimo (50s)

Sonido: Se utilizarán diferentes tipos de sonido como: música de fondo, locución y efectos.

Especificación del contenido de la aplicación:

Recopilación y preparación de los medios:

De acuerdo con las fuentes para obtener los medios definidas anteriormente, se procede a recopilar cada uno de ellos y luego a su preparación que en cada uno tendrá características especiales:

Textos: El texto en esta aplicación es almacenado con caracteres y como imagen.

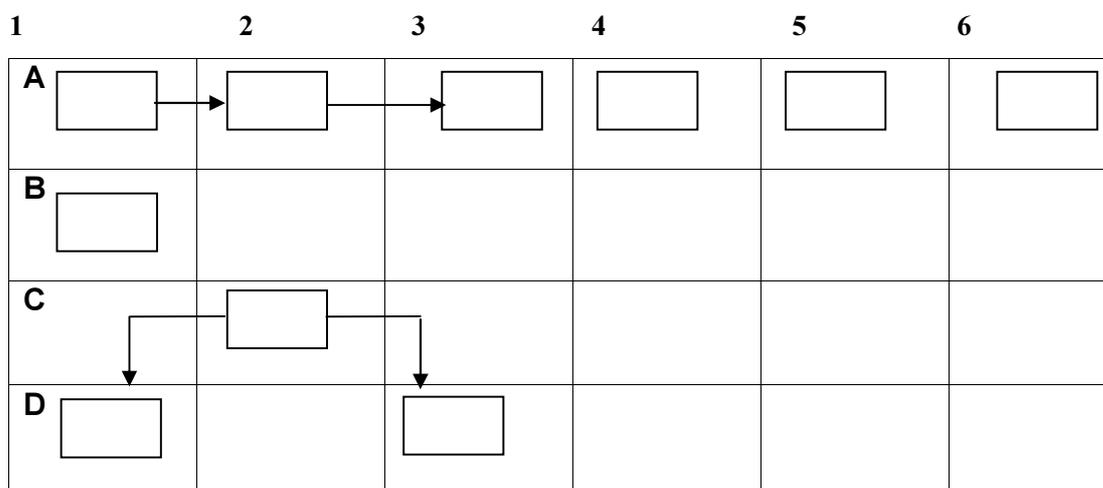
Imágenes: Las imágenes escogidas son imágenes pequeñas relacionadas con los diferentes temas.

Sonido: El sonido es almacenado en un medio externo a la computadora como cinta CD musical, para editarlo o en una carpeta creada por el especialista en el HDD, llamada sonido.

Videos / Animación: La fuente de los videos está digitalizada.

A continuación se muestra el diagrama de flujo para el desarrollo de la aplicación Multimedia.

Diagrama de flujo



Guión:

Tema	Posición Diag	Entradas	Alcance Inform	Acciones usuarios	Respuesta sistema
Presentación	1A	CD	animación	abrir	El (s) muestra la interfaz de presentación
Test(t)	1B	click	texto	Buscar idioma	Según el idioma escogido el (s) muestra un test y muestra los resultados obtenidos por el usuario.
Menú(m)	2A	click	Buscar d,e,g,p,o,v,a	Buscar d,e,g,p,o,v	El (s) presenta una interfaz que le permite al usuario acceder a al (d, e, g, p, o, v, e y a.)
Diálogo(d)	3A	click	Texto,video	Escuchar y traducir	Muestra de los diálogos (videos).
Ejercicios(e)	2C	click	Texto	Entrar en e	El (s) muestra los tipos de ejercicios que tendrá la aplicación
Ejerc.gram(g)	1D	click	Texto, imagen	Realizar g, después de haber escuchado el diálogo	Muestra los textos con las preguntas gramaticales
Ejerc.pronun(p)	3D	click	Texto,sonido	Realizar ejercicios y escuchar los sonidos de palabras y oraciones	Muestra los sonidos de palabras y frases.
Opciones(o)	4A	click	Imagen, texto, sonido	Subir y bajar volumen, etc	Muestra una pantalla con las distintas opciones de la aplic.
Evaluación(v)	5A	click	Texto, imagen	Dar resultados de la evaluación	El (s) muestra el resultado de la evaluación final del usuario que interactuó con la Multimedia.
Ayuda(a)	6A	click	Texto, imagen	Ofrece ayuda inicial al usuario	El (s) muestra la ayuda al usuario para indicarle cómo trabaja el sistema

CONCLUSIONES

En el presente trabajo de diploma se logró proponer una guía con la documentación necesaria de las metodologías para el desarrollo de Hipermedia y Web, y aplicaciones multimedia, permitiendo elevar el conocimiento de los desarrolladores de software al enfrentarse a estos tipos de aplicaciones.

Se analizaron las distintas metodologías de desarrollo existentes para Hipermedia y Web, y multimedia, y la documentación que proponen las mismas, seleccionando y detallando las características fundamentales y las fases de cada una de ellas.

A partir de todo el trabajo realizado, se desarrolló una propuesta de las estudiadas para Hipermedia y Web, y aplicaciones multimedia, la cual recoge los elementos necesarios para la construcción del software.

Finalmente se realizó un ejemplo de cada metodología propuestas que ayude al desarrollador a ponerla en práctica al enfrentarse a estos tipos de aplicaciones.

La guía metodológica afirma la necesidad de documentar la construcción del software.

RECOMENDACIONES

En este trabajo de diploma se alcanzaron de manera general los objetivos planteados, pero se considera que para solucionar los problemas con la documentación de las aplicaciones de escritorio, se necesita de un proceso investigativo que recoja a las metodologías ágiles, debido a el porcentaje de utilización obtenido en las encuestas realizadas de estas metodologías para este tipo de aplicación.

Para validar la guía se recomienda:

1. Hacer talleres en las Universidades donde se deje clara la importancia de la documentación durante el desarrollo de Hipermedia y Web, y aplicaciones multimedia.
2. Generalizar la guía a otros centros de trabajo y estudio, con el fin de elevar el conocimiento de estas metodologías de desarrollo.
3. Que la guía metodológica sea tomada en cuenta por parte de las instancias superiores y que se use en los proyectos que desarrollan software actualmente en la universidad.

BIBLIOGRAFÍA

- Craig, L. (2004). UML y patrones. Introducción al análisis y diseño orientado a objetos. Tomo I. Editorial felix Varela. La habana.
- Colectivo de autores (2000) Introducción a la informática educativa. Universidad de Pinar del Río. Hermanos Saínez e ISPJAE. Versión digital.
- Escalona Cuaresma, M.J. (2004). Modelos y técnicas para la especificación y el análisis de la navegación en sistemas software. Tesis presentada en opción al grado de Doctor, Escuela Técnica Superior de Ingeniería Informática Universidad de Sevilla, octubre 2004
- García Rubio, F.O. y Bravo Santos, C. Ingeniería del Software. Ingeniería del Software de Gestión. En: http://alarcos.inf-cr.uclm.es/per/fgarcia/isoftware/doc/tema3_1xh.pdf
- Kirschner, J. (2003). Aspect-oriented modeling of Multimedia. Applications (OMMMA). En: <http://www.Jens.kirschner.com/pub/ommma.pdf>
- Santos Fabelo, M. del C. Prevención de las disgrafías escolares: una necesidad de la escuela actual, para la atención a la diversidad. En: <http://www.monografias.com/trabajos55/disgrafia-escolar/disgrafia-colar2.shtml/> 16 de junio. 2008.
- Sonlezal Fernández, G y Díaz Catalá, S. Multimedia Auto-Aprende. Tesis presentada en opción al grado de ingeniero, ISPJAE, 2006.
- Wikipedia, la enciclopedia libre <http://es.wikipedia.org/wiki/Hipermedia>, 2008. En: <http://claseinscoop.wordpress.com/2008/02/09/que-son-aplicaciones-multimedia/> 17 de junio 2008.

GLOSARIO DE TÉRMINOS

Actor: Alguien o algo, fuera del sistema o negocio que interactúa con él.

Asociación: Descripción de un conjunto relacionado de enlaces o vínculos entre objetos de dos tipos.

BNL: El lenguaje natural acotado, es una propuesta que permite diseñar interfaces de usuario. Se basa en expresar las consultas que el usuario necesita realizar en el sistema mediante el lenguaje natural, lo que permite que los resultados obtenidos sean fácilmente entendibles por el cliente.

Caso de uso: Descripción narrativa textual de la secuencia de eventos y acciones que ocurren cuando un usuario parte o divide en un diálogo con un sistema durante un proceso significativo.

Clase: Una descripción de un conjunto de objetos que comparten los mismos atributos, operaciones, relaciones y semántica.

Cliente: Persona, organización o grupo de personas que encarga la construcción de un sistema, ya sea empezando desde cero, o mediante el refinamiento de versiones sucesivas.

Desarrollador: Trabajador participante en un flujo de trabajo fundamental. Por ejemplo, un ingeniero de casos de uso, un ingeniero de componentes, etc.

Diagrama: La representación gráfica de un conjunto de elementos, usualmente representado como un grafo conectado de vértices (elementos) y arcos (relaciones).

Diagrama de caso de uso: Un diagrama que muestra un conjunto de casos de uso y de actores y sus relaciones; los diagramas de casos de uso muestran los casos de uso de un sistema desde un punto de vista estático.

Diagrama de clases: Un diagrama que muestra un conjunto de clases, interfaces y colaboraciones y las relaciones entre estos; los diagramas de clases muestran el diseño de un sistema desde el punto de vista estático; un diagrama que muestra una colección de elementos (estáticos) declarativos.

Estado: condición de un objeto entre eventos.

Fase: Periodo de tiempo entre dos hitos principales de un proceso de desarrollo.

Glosario de términos: Términos comunes que se utilizan para describir el sistema.

Herramientas CASE: Son diversas aplicaciones informáticas destinadas a aumentar la productividad en el desarrollo de software reduciendo el coste de las mismas en términos de tiempo y de dinero. Estas herramientas nos pueden ayudar en todos los aspectos del ciclo de vida de desarrollo del software en tareas como el proceso de realizar un diseño del proyecto, calculo de costes, implementación de parte del código automáticamente con el diseño dado, compilación automática, documentación o detección de errores entre otras.

Interfaz: Conjunto de representaciones de operaciones públicas.

Metodologías de desarrollo de software: Son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

Modelo de caso de uso: es un conjunto de casos, actores y sus relaciones.

Modelo conceptual: representación de conceptos en un dominio del problema

Modelo E-R: es un lenguaje gráfico para describir conceptos. Informalmente, son simples dibujos o gráficos que describen la información que trata un sistema de información y el software que lo automatiza.

Navegación: La navegación se define como un subconjunto de relaciones semánticas establecidas entre clases. Permite ir de un objeto a otro relacionado semánticamente. La navegación es la más importante característica de la hipermedia

Nodo: Un elemento físico que existe en tiempo de ejecución y que representa un recurso computacional, que en general tiene al menos alguna memoria y a menudo capacidad de procesamiento.

OMMMA-L: es el Lenguaje de Modelado Orientado a objetos de Aplicaciones Multimedia. Se lanza como una propuesta de extensión de UML para la integración de especificaciones de sistemas multimedia basados en el paradigma orientado a objetos.

Pantalla: es un grupo de elementos de medias visuales que están comprendidos en una vista determinada.

Paquetes: Son usados para organizar y manipular la complejidad de los modelos largos. Un grupo de paquetes modelan elementos y los diagramas semejantes como el uso de casos, clases, actividades, procesos, estados, etc., y sus diagramas

asociados; en tal camino que eso puede ser remitido como uno entero. Los paquetes pueden ser representados en un diagrama, remitido como Diagrama de Paquete.

Patrón: Es la descripción etiquetada de un problema, de la solución, de cuándo aplicar la solución de manera de hacerlo dentro de otros contextos.

Producto: Resultado de cada etapa.

Prototipo: Primera versión de un nuevo tipo de producto, en el que se han incorporado sólo algunas características del sistema final, o no se han realizado completamente.

Proyecto: Esfuerzo de desarrollo para llevar un sistema a lo largo de un ciclo de vida.

Requisitos: Son las funciones, servicios y restricciones operativas del sistema.

Rol: Papel que desempeña cada integrante dentro del ciclo de vida de un proyecto de software.

Software: Se refiere a los programas y datos almacenados en un ordenador.

Sistema: delimita el mundo sobre el cual se está construyendo el modelo.

Slices: Un *slice* es un subconjunto de atributos de una entidad que van a ser presentados de forma agrupada al usuario, es lo que se podría llamar una vista del sistema.

UML: es el Lenguaje de Modelación Unificado es un lenguaje gráfico para detallar, construir, visualizar y documentar las partes o artefactos (información que se utiliza o produce mediante un proceso de software).

Usuario: Persona que interactúa con un sistema.

Visión: la visión del usuario sobre producto que se va a desarrollar.

Anexo 1

ENCUESTA A DESARROLLADORES DE SOFTWARE
2007-2008

Institución _____ Fecha: _____

La información que le solicitamos a continuación, acerca de las metodologías más utilizadas para el desarrollo de software por usted, es anónima.

1. ¿Cuáles son las metodologías que más utiliza para el desarrollo de software?

2. ¿Cuál metodología es la más utilizada para desarrollar las siguientes tipos de aplicaciones?:

Tipo de aplicaciones

Metodología

- Multimedia

- Aplicaciones de escritorio

- Aplicaciones WEB

- Páginas WEB

3. ¿Cuál es la metodología que más utiliza para proyectos?

a. A corto plazo _____

b. A largo plazo _____

c. Ambos _____

4. RUP es la metodología más adecuada para crear:

- Multimedia
- Aplicaciones de escritorio
- Aplicaciones WEB
- Páginas WEB
- Ninguna

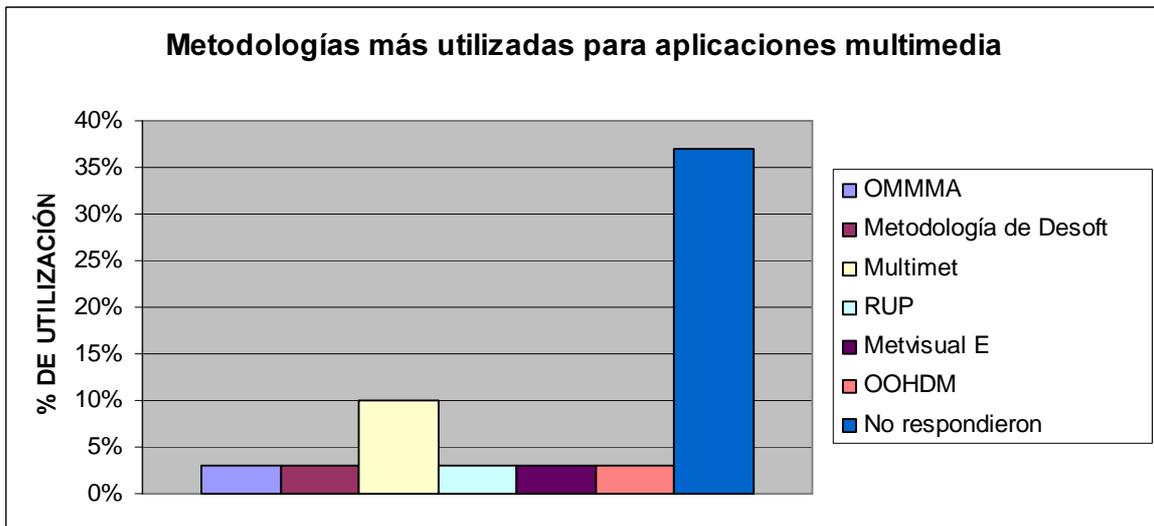
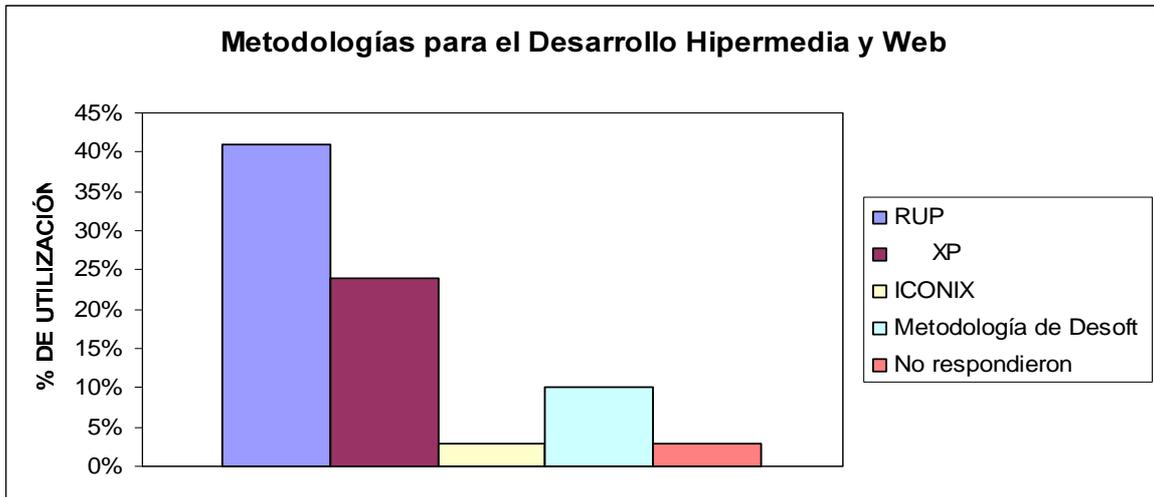
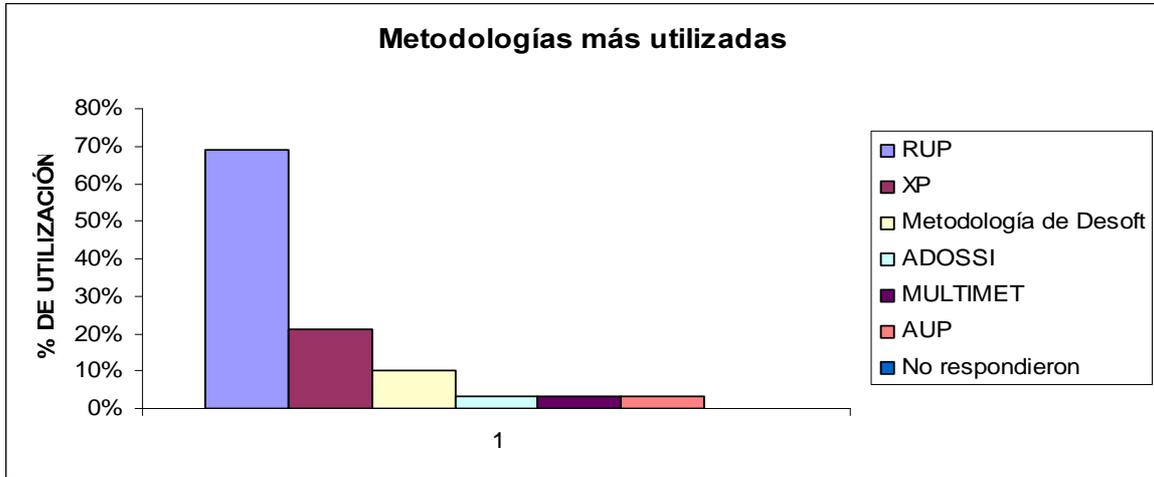
5. Sugerencias fundamentales de las metodologías propuestas por usted anteriormente para cada tipo de aplicación.

Multimedia _____

Aplicaciones de escritorio _____

Aplicaciones WEB _____

Anexo 2



Anexo 3. Sugerencias

Aplicaciones de Escritorios

1. XP suele funcionar bien solo que no resulta ser la mejor solución cuando se desea ser escalable.
2. Eliminar documentación. En ocasiones es redundante la documentación que propone RUP.
3. Si el proyecto es a corto plazo se podría utilizar la metodología RUP con alguna metodología ágil como XP. Metodologías ágiles que faciliten el trabajo productivo. XP con una de sus técnicas (programación en parejas).
4. Dependiendo de la magnitud de las aplicaciones de escritorio que se van a desarrollar se escogerá la metodología. Para aplicaciones de escritorio a largo plazo o grandes aplicaciones RUP es una magnífica metodología, pero requiere de mucha documentación aún para proyectos pequeños y medianos, cosa que no representa ningún beneficio para los desarrolladores.

Aplicaciones Web

1. Eliminar documentación. En ocasiones es redundante la documentación que propone RUP.
2. Metodologías ágiles que faciliten el trabajo productivo. XP con una de sus técnicas (programación en parejas).
3. La metodología de desoft es la más apropiada, además está basada en RUP más la experiencia, es iterativo e incremental, posee las mismas etapas de desarrollo, se realiza un profundo estudio del negocio, hay captura de requisitos y muestra el ciclo de vida del software.

Aplicaciones Multimedia

1. No he utilizado ninguna metodología para el desarrollo de aplicaciones de este tipo.
2. Metvisual E resulta muy eficiente por lo que no tengo sugerencias.
3. Metodologías ágiles que faciliten el trabajo productivo. XP con una de sus técnicas (programación en parejas).