

Diseño del Modelo Navegacional para Sistemas de Tratamiento de Bibliotecas Digitales

Jesús Torres, Manuel Mejías, M^aJosé Escalona, José A. Ortega, Juan M. Cordero

Dpto. Lenguajes y Sistemas Informáticos
Universidad de Sevilla
jtortes@lsi.us.es

Resumen. Los sistemas de tratamiento de bibliotecas digitales nacen como sistemas complejos en los que se mezclan importantes requisitos de almacenamiento de información, debido a los múltiples medios que se usan en ellos, con grandes necesidades funcionales. Pero además en el entorno de las bibliotecas digitales se añaden unas necesidades, que son propias de las aplicaciones multimedia y páginas web, como son el uso de los enlaces y el ofrecer una atractiva interfaz de usuario. Estas nuevas necesidades se hacen patentes principalmente en la fase de diseño de la aplicación, puesto que es en ella donde se le añadirán a la aplicación. En este trabajo, se propone cómo se puede incorporar el diseño de los aspectos de navegación a una aplicación.

1 Introducción

Una biblioteca digital es una biblioteca que ha sido extendida y mejorada mediante la aplicación de la tecnología digital. Es la unión de ordenadores, sistemas de almacenamiento y redes de comunicaciones con el contenido y el software necesario para reproducir, emular y extender los servicios proporcionados por las bibliotecas convencionales. Una biblioteca digital debe cumplir todas las tareas de una biblioteca convencional y explotar las ventajas de la tecnología digital en el almacenamiento, la búsqueda y las comunicaciones, además de la integración de nuevos tipos de medios (textos, imágenes, sonidos, vídeos, animaciones, etc). La biblioteca digital proporciona a una comunidad de usuarios un acceso coherente a repositorios de información grandes y organizados. Las bibliotecas digitales son construidas (recogiendo y organizando la información) por una comunidad de usuarios y sus funcionalidades son acordes a las necesidades de información de dicha comunidad. Las posibilidades de los usuarios para acceder, reorganizar y utilizar este repositorio están enriquecidas con las capacidades de la tecnología digital.

Analizando esta definición, se llega a la conclusión de que los sistemas de tratamiento de bibliotecas digitales van a ser sistemas complejos que deben hacer frente al almacenamiento eficiente y eficaz de información de diferente tipo (imagen, sonido, texto, animaciones, etc.), así como a importantes requisitos funcionales que permitan el manejo de esta diversidad de medios de una forma ágil y sencilla para el

usuario. A esto hay que unirle el hecho de que son sistemas en red y por tanto concurrentes y distribuidos. Este tipo de necesidades que denominaremos requisitos clásicos [4], podríamos diseñarlos mediante la propuesta realizada por el Proceso Unificado de UML, llegando así a un modelo de objetos de diseño y un diseño de la arquitectura. Sin embargo, hay aspectos que nacen con las bibliotecas digitales y que son necesarios diseñar. Uno de estos aspectos es lo que denominaremos requisitos de navegación o requisitos navegacionales [13]. Los sistemas para el tratamiento de bibliotecas digitales son sistemas que nacen orientados a internet. Algo característico de las páginas web son los enlaces que permiten al usuario ir de una página a otra. En el entorno de las bibliotecas digitales, estos enlaces van a ser lo que se ha denominado hipertexto [10], puesto que el origen del enlace no tiene por qué ser un texto, puede ser una imagen, un gráfico, etc. [7]

Otro de los aspectos que toman un especial interés en las bibliotecas digitales es lo que vamos a denominar Interfaz Abstracta de Usuario [12]. En la mayoría de las aplicaciones que se desarrollan en internet, aparecen diferentes tipos de usuario, cada uno de los cuales van a tener una visión del sistema. Incluso, cada uno de ellos va a tener diferentes posibilidades de navegación.

En resumen, los sistemas para el tratamiento de bibliotecas digitales recogen tanto aspectos propios de las aplicaciones de gestión (requisitos de almacenamiento, funcionales, etc.) como aspectos que hasta ahora eran propios de las aplicaciones multimedia y de la web (navegación, múltiples medios, etc.) En la figura 1 vemos recogida esta idea.

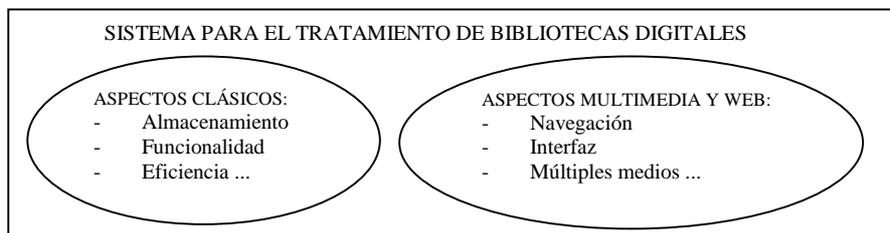


Figura 1: Aspectos que debe cubrir un sistema para el tratamiento de bibliotecas digitales

En este documento vamos a analizar cómo partiendo de un modelo de clases de diseño que llamaremos modelo básico, en el que solo se recojan los aspectos clásicos (requisitos de almacenamiento de información, requisitos funcionales, etc.) y que se obtendrá aplicando la propuesta del Proceso Unificado de UML, se puede llegar a un diseño completo del sistema de tratamiento de bibliotecas digitales en el que se recojan también los aspectos de navegación. Este documento se centra especialmente en los aspectos de navegación, aunque en el diseño de sistemas para el tratamiento de bibliotecas digitales sería necesario incluir también esos otros aspectos que hemos denominado de interfaz abstracta.

Para añadir estos nuevos aspectos, vamos a partir de la base propuesta por las metodologías de desarrollo de aplicaciones multimedia, debido a que la navegación y la importancia de la interfaz son aspectos que se hayan en las bibliotecas digitales al

igual que las aplicaciones multimedia. Dentro de las metodologías más conocidas como son HDM (Hypertext Design Model) y RMM (Relationship Management Methodology), OOHDM [11] (Object Oriented Hypermedia Design Model) y EORM [9] (Enhanced Object Relationship Model), sólo las dos últimas asumen el paradigma de la orientación a objetos como paradigma de diseño. Tomar el enfoque orientado a objetos como paradigma de diseño es muy útil debido al gran nivel de abstracción que ofrece y a sus mecanismos de composición que facilitan el modelado de la estructura hipermedial [9]. Por ello, parecen las más interesantes de estudiar como base para el desarrollo de sistemas basados en bibliotecas digitales. Si comparamos ahora EORM y OOHDM, para seleccionar una como marco de referencia, veremos que, a pesar de que en el diseño de los conceptos de navegación de la aplicación son bastantes similares, OOHDM hace más hincapié en el diseño de la interfaz de usuario, dedicando una fase de la metodología al diseño de la interfaz abstracta. Por ello, en este documento se va a tomar a OOHDM como base para el diseño de la navegación y la interfaz abstracta.

2 Diseño de bibliotecas digitales

La fase de diseño para sistemas de bibliotecas digitales va a dividirse en tres subfases fundamentalmente: subfase de diseño básico, subfase de diseño navegacional y subfase de diseño de interfaz abstracta. En la figura 2 vemos un esquema de estas subfases y de los productos que se van a obtener en cada una de ellas [5]:

Diseño	Diseño Básico	<ul style="list-style-type: none"> - La arquitectura abstracta del sistema - La división del sistema en subsistemas - El diseño de los casos de uso - El modelo de clases de diseño
	Diseño Navegacional	<ul style="list-style-type: none"> - Modelo de clases navegacionales
	Diseño de Interfaz Abstracta	<ul style="list-style-type: none"> - Modelo de clases navegacionales refinado con los aspectos de interfaz abstracta - Vistas abstractas de datos (ADV) - Diagramas de configuración - Modelo dinámico de los ADVs

Figura 2: Resumen de la fase de diseño y de los productos a obtener para los sistemas de tratamiento de bibliotecas digitales.

Vamos a ir analizando cada una de estas subfases, centrándonos especialmente en la segunda, puesto que es el diseño navegacional el objetivo de este trabajo. Para aplicar las ideas propuestas, vamos a plantearnos el desarrollo de una aplicación que trate con una biblioteca digital que muestre los datos referentes a los bienes muebles del patrimonio histórico de una comunidad autónoma. Este ejemplo solo contendrá una parte de las necesidades que serían necesarias para la gestión de esta biblioteca.

3 Planteamiento del problema

Vamos a plantearnos el desarrollo de una biblioteca digital que sirva para la gestión y la catalogación de los bienes muebles de una comunidad autónoma. Fundamentalmente se van a almacenar datos en cuatro módulos diferentes. Un módulo no es más que una agrupación de la información referente al bien. La información que se recoge en un determinado grupo tiene una cierta relación, bajo algún concepto que es especialmente interesante para el usuario. De esta forma, se presentará la información del bien estructurada según los criterios del propio usuario. Como decíamos, para nuestro ejemplo existen cuatro módulos: el módulo de datos de identificación, en los que se almacenan datos como la denominación del bien, el período histórico en el que se ubica, la dirección donde se encuentra el bien, etc., este módulo va a almacenar fundamentalmente campos de tipo cadena; el segundo módulo va a ser el módulo de documentación del bien, en él se mostrará documentación textual emitida por catalogadores sobre el bien (como libros, documentos web, etc.) y documentación gráfica, que serán imágenes digitales del bien; el tercer módulo almacenará datos de los estudios de conservación y restauración del bien, en él se recogerán los expedientes que recojan el estado del bien en una determinada fecha, así como la película de la intervención de restauración si ésta se ha realizado; y el último mostrará los datos museográficos del bien, las cuales serán datos referentes al museo en los que se almacena el bien, datos que indicarán si el bien es visitable o no, o incluso puede incluirse un vídeo que permita realizar una visita virtual por el museo. El ejemplo que se ha propuesto está basado en un ejemplo real, el Sistema de Información del Patrimonio Mueble del Instituto Andaluz de Patrimonio Histórico.

4 Diseño básico

La subfase de diseño básico consiste en realizar los pasos propios del diseño de una aplicación de gestión clásica, manteniendo al margen el hecho de que existan necesidades de navegación e interfaz abstracta. En el diseño básico hay que seleccionar la arquitectura que mejor soporte al sistema, así como los casos de uso que se hayan diseñado en fases anteriores del proceso de desarrollo. Además, habrá que obtener el modelo de clases de diseño. En nuestro caso el modelo sería el que se muestra en la figura 3.

En estas clases hay más atributos que no se han recogido para hacer más sencillo el modelo. En la clase BienMueble encontraríamos todos los atributos del bien que sólo puede tener un valor en cada bien. Por ejemplo, la denominación del bien, la dirección en la que se ubica o el nombre del museo en el que se encuentra. En el modelo encontramos también otras clases que representan las imágenes de un bien diferenciadas por un número de registro, los documentos (libros y documentos) que hablan del bien y los estudios de conservación que se realizan del mismo. Mientras que un bien sólo tiene una denominación, puede tener varias imágenes, varios estudios de conservación y varios documentos que lo refieran, de ahí la necesidad de crear estas clases para representar la multiplicidad.

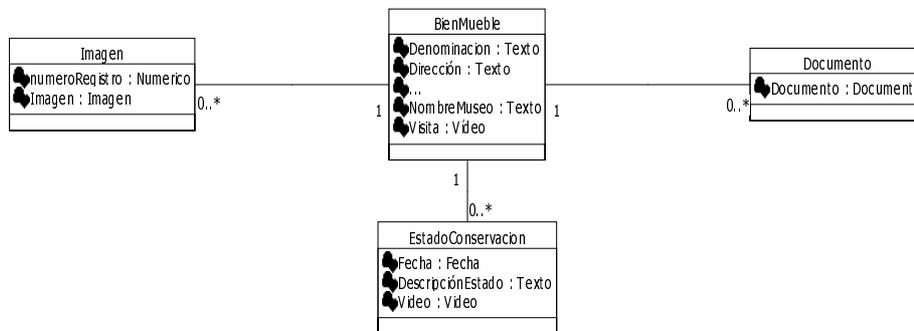


Figura 3: Modelo de clases de diseño básico

5 Diseño navegacional

Una vez realizado el diseño básico, hay que pasar a diseñar los aspectos navegacionales. A la hora de realizar un modelo navegacional, hay que tener en cuenta qué objetos del modelo básico van a ser navegables, qué tipo de relaciones y estructuras de composición hay entre estos objetos navegables y las conexiones entre los distintos objetos y las posibilidades de navegación de un nodo a otro. Proponemos el esquema de Clases Navegacionales [5] el cual se obtiene a partir del modelo de clases de diseño obtenido en el Diseño básico. El modelo básico se va a modificar de manera que se va a obtener un nuevo modelo que sólo va a incluir *nodos*, *enlaces* o *clases índice* y que nos va a permitir representar las posibilidades de navegación en el sistema.

Para un mismo modelo básico, se pueden diseñar diferentes sistemas de navegación, de ahí que un mismo modelo básico pueda dar varios modelos de navegación. El hecho de separar los conceptos de navegación de los conceptos de diseño básico facilita la reutilización del diseño, puesto que, si cambian los requisitos de navegación cambiará sólo el modelo navegacional, sin tener que modificarse el modelo de diseño básico.

Antes de continuar es necesario hacer una definición más concreta de lo que son clases nodo, clases enlace y clases índice. Una clase nodo es un tipo especial de clase que muestra una agrupación de información bajo un determinado criterio. Los atributos y operaciones de una clase nodo van a ser atributos y operaciones de clases del modelo básico. Una clase nodo puede contener atributos y operaciones de diferentes clases del modelo básico de la misma forma que una propiedad (atributo u operación) de una clase del modelo básico puede estar en diferentes nodos.

Por su parte, los enlaces son otras clases especiales que van a definir un tipo. Mientras que los nodos son propios de cada aplicación, la definición de la clase tipo enlace es genérica y podemos verla en la figura 4.

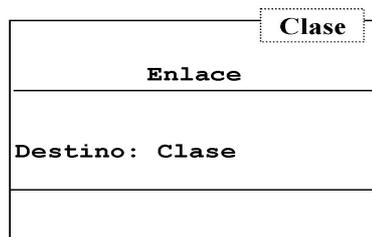


Figura 4: Definición de la clase enlace

El único atributo que tiene es el que indica la clase a la que se llega al ejecutar el enlace y que es del tipo parámetro en el que se instancia. Esta es una definición muy básica de la clase enlace pero pueden añadirse otros atributos u operaciones. La forma de instanciarla se ve en la figura 6.

Por último, las clases índice son unas clases que sirven para representar menús, diccionarios, etc. en la navegación. Las clases índice suelen estar compuestas por bastantes atributos de tipo enlace que llevan de un nodo a otro.

Para nuestro ejemplo, el modelo definido en el diseño básico podría estructurarse en cuatro nodos: el de datos de identificación, que englobarían a los datos básicos que identifican al bien; el de datos de documentación, que recogería tanto los datos de documentación textual (Documentos) como de información gráfica (imágenes); el de datos de conservación, que mostraría los datos sobre el estado de conservación del bien; y el de datos museográficos que contendrá los datos del bien que hacen referencia a los datos del museo en el que se encuentra ubicado. Además, vamos a definir una clase índice que nos permita navegar desde un nodo a otro. A su vez, cada nodo tendrá un enlace (AÍndice) que permitirá llegar hasta la clase índice *IndiceBienMueble*. Así obtenemos el modelo de clases que se muestra en la figura 5.

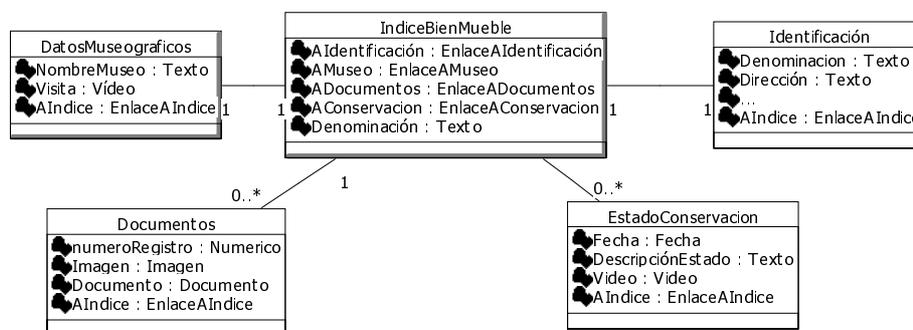


Figura 5: Modelo Navegacional para el ejemplo de los bienes muebles

En esta figura, vemos una serie de tipos no conocidos aún. Estos tipos son el tipo enlace concretizados a una clase. Para realizar esta tarea podemos utilizar el estereotipo *bind* de UML. Así, como vemos en la figura 6, el tipo *EnlaceAÍndice* es una concretización a la clase *IndiceBienMueble*. Por tanto, si encontramos el atributo *AÍndice* en la clase *Documentos*, por ejemplo, del tipo *EnlaceAÍndice*, indicamos con ello que este índice lleva a la clase *IndiceBienMueble*. Definida esta clase, podemos concretizarla y crear nuevas clases en las que la clase hacia la que se dirige el nodo sea una clase del sistema.

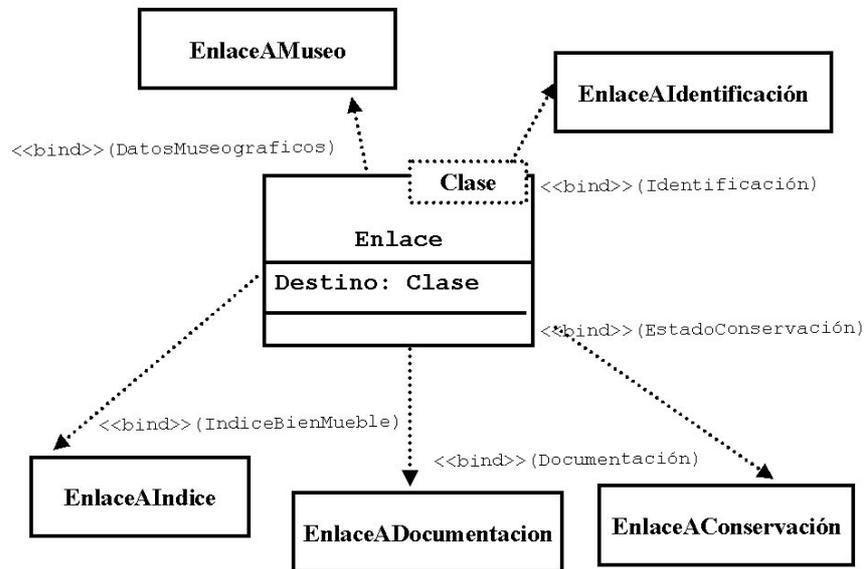


Figura 6: Instancias del tipo enlace en el modelo de bienes muebles

Nótese que cada uno de los nodos definidos en la figura 5 se corresponde con los módulos de información que definimos en el apartado 3. Como vemos, una instancia de la clase *IndiceBienMueble*, sólo se relaciona con un módulo de *DatosMuseográficos* y con uno de Identificación, mientras que puede tener varios módulos de documentos y varios de estudios de conservación. Si comparamos esta figura con el modelo de la figura 3, podemos observar que existe una relación entre los atributos y las multiplicidades que aparecen en ambos. Esta correspondencia hay que dejarla patente, por ello, una vez que se tiene diseñado este modelo, es necesario recoger la relación que estas clases tienen con las clases del diseño básico. Para ello, vamos a basarnos en la propuesta de [11]. En la figura 7 encontramos la definición del nodo datos museográficos.

```

NODO DatosMuseográficos [FROM BienMueble:BM]
NombreMuseo: BM.NombreMuseo
Visita: MB.Visita
  
```

Figura 7: Definición de la clase nodo datos museográficos en función del diseño básico

Aquí se indica que se define el nodo *DatosMuseográficos* que va a tomar datos de la clase *BienMueble* del modelo de diseño básico. Esto lo indicamos a través de la

cláusula FROM. Tras esta cláusula puede haber varias clases del modelo básico, si es que el nodo toma datos de varias clases. BM es una instancia de la clase *BienMueble* de diseño básico. Así, el atributo *NombreMuseo* del nodo, se corresponde con el atributo *NombreMuseo* de la clase de diseño básico *BienMueble*. Vemos ahora en la figura 8 la definición de nodo *Documentos*. Según vemos por la cláusula FROM, este toma datos de dos clases del diseño básico: Documento e Imagen.

```
NODO Documentos [FROM Imagen:I, Documento: D]
NumeroRegistro: I.numeroRegistro
Imagen: I.Imagen
```

Figura 8: Definición de la clase nodo Documentos en función del diseño básico

Como vemos, aquí se declaran dos instancias, la de Imagen, I, y la de Documento, D, que nos sirven para indicar de qué clases del diseño básico tomamos los datos.

Para declarar el otro tipo de clase especial, las clases índice, la nomenclatura es similar, la única diferencia se encuentra en que habrá que cambiar la palabra NODO por ÍNDICE. Así nuestra clase *ÍndiceBienMueble* quedaría como se muestra en la figura 9. Vemos aquí que las clases índices pueden tener atributos, aunque su filosofía es recoger enlaces.

```
INDICE IndiceBienMueble [FROM BienMueble:BM]
Denominación: BM.Denominación
```

Figura 9: Definición de la clase índice IndiceBienMueble en función del diseño básico

6 Camino a la implementación

Desarrollada la fase de diseño navegacional, sería necesario terminar con la última fase de diseño, el diseño de interfaz abstracta [12] que como ya se ha indicado no es objetivo de este documento pero que vamos a introducir. Ya hemos introducido en el apartado primero que dentro de los sistemas para el tratamiento de bibliotecas digitales, existen diferentes grupos de usuarios que tienen distintos permisos y posibilidades de navegación dentro de la aplicación. El diseño de la interfaz abstracta va a recoger si el usuario puede ver los datos de los nodos y las posibilidades de navegación que tiene. Para recoger esta idea, se seguiría la propuesta de [2]. En ella, se hace uso de las Vistas Abstractas de Datos (ADV's) [3] para representar la visión de los nodos que tiene cada usuario y las posibilidades de navegación. Además, se incluyen los llamados Diagramas de Configuración [11] que permite indicar la

relación entre estos ADVs y las clases navegacionales. Por último, el dinamismo de los ADVs se recogerá mediante máquinas de estados que indicarán cómo éstas se ven afectadas por los eventos que se producen en el sistema. Cada grupo de usuario, tendrá su visión del sistema, por lo que cada uno tendrá su propio modelo de interfaz abstracta compuesto por los ADVs, el diagrama de configuración y las máquinas de estado [8].

Terminado el diseño de interfaz abstracta y habiendo añadido el diseño navegacional tal y como se ha descrito en los apartados anteriores, el camino a la fase de la implementación resulta más sencillo.

Por un lado, al mantenerse la fase de diseño básico de forma similar a la propuesta por el Proceso Unificado de UML, las nuevas técnicas de diseño de software basado en métodos formales, tales como el diseño de persistencia [1] de los conceptos de concurrencia, así como el uso de los conocidos patrones de diseño, se pueden seguir aplicando. Manteniéndose así las ventajas que esto conlleva.

Pero además, con la aplicación de las nuevas subfases de diseño, podemos facilitar la tarea del implementador. Al diseñar el modelo básico, el implementador conocerá qué información debe almacenar en el sistema y qué operaciones se podrán realizar sobre él. Pero si a esto le añadimos el modelo navegacional, el implementador sabrá además cómo se agrupará esa información a la hora de presentarla y cómo se podrá pasar de un grupo de información a otro. Con el diseño de la interfaz abstracta, el programador podrá, por último, adecuar la presentación de la información a los grupos de usuarios que se requieran.

7 Conclusiones

Este trabajo ha presentado cómo se puede adecuar la propuesta de diseño de UML al diseño de bibliotecas digitales haciendo uso de los mecanismos usados en OOHDM. El diseño se ve como una fase compuesta por tres subfases en las que se estudian los aspectos más críticos de los sistemas para el tratamiento de bibliotecas digitales [2]. Con el diseño básico se estudian aspectos como las necesidades de almacenamiento y funcionalidad, la división en subsistemas y la arquitectura. Con el diseño navegacional los aspectos de navegación. Y, por último, con el diseño de interfaz abstracta se estudian la variabilidad de presentación de los datos dependiendo del grupo de usuarios.

El trabajo analiza en profundidad la subfase de diseño navegacional, que se basa directamente en las propuestas de las metodologías de desarrollo de aplicaciones multimedia. Se recogen las ventajas que supone separar la navegación de los aspectos básicos como de la interfaz para la reutilización de código y el paso a la implementación. Para ejemplificar estas ideas, se ha propuesto una versión reducida de ejemplo real de desarrollo de biblioteca digital que se está llevando a cabo en el Instituto Andaluz de Patrimonio Histórico, el desarrollo del Sistema de Información del Patrimonio Mueble de Andalucía [6].

Referencias

1. S.W. Ambler "Diseño de una capa de persistencia robusta para bases de datos relacionales" Copyright 1997-1999
2. M.J. Escalona, J.Torres, M.Mejías "Propuesta de metodología para el desarrollo de sistemas para el tratamiento de bibliotecas digitales", Report Interno, 2-2000 del Departamento de Lenguajes y Sistemas Informáticos. Universidad de Sevilla. Julio 2000.
3. D.D.Cowan, D.J.P.Lucena, "Abstract Data Views. An Interface Specification Concept to Enhance Design for Reuse", IEEE Transactions on Software Engineering. 18(1), 9-18, January 1995.
4. A. Durán. "Una Metodología para la Ingeniería de Requisitos". Tesis Doctoral. Depto. de Lenguajes y Sistemas Informáticos. Universidad de Sevilla, Septiembre 2000.
5. M.J. Escalona, M.Mejías, J.Torres. "Aproximación metodológica al desarrollo de sistemas para el tratamiento de bibliotecas digitales". Aceptado en las V Jornadas de Ingeniería del Software. Noviembre 2000.
6. M.J. Escalona, J.Torres, M.Mejías. "Aplicación de los sistemas de tratamiento de bibliotecas digitales al Sistema de Información del Patrimonio Histórico Andaluz". Boletín Trimestral del Instituto Andaluz de Patrimonio Histórico, Julio 2000.
7. D. Greco, M. Eskelinen, C. Funckhouser, M.Luesebrink, J.Rosenberg "Actual & Potential Hypertext & Hypermedia", ACM Digital Library. June 20-24, 1998.
8. M.Hoch, D. Schwabe. "Group interaction in a surround screen environment". Inst. for Visual Media ZKM, Karlsruhe, Germany. IEEE Computer Animation, 1999
9. D.B. Lange "An Object-Oriented Design Approach for Developing Hipermedia Information Systems", Research Report RT00112, IBM Research, Tokyo Research Laboratory, Japón, 1995.
10. J.M. Martínez, J.R.Hilera, J. Martínez, J.A. Gutiérrez, "Orientación a Objetos en la Documentación Hipermedia", Universidad de Alcalá de Henares, Departamento de Ciencias de la Computación.
11. G. Rossi. "An Object Oriented Method for Designing Hipermedia Applications". PHD Thesis, Departamento de Informática, PUC-Rio, Brazil, 1996.
12. G. Rossi, D. Schwabe, A. Garrido: "Design Reuse in Hipermedia Applications Development". Proceedings of ACM International Conference on Hypertext (Hypertext'97), Southampton, April 7-11, 1997. ACM Press.
13. D.Schwabe, G.Rossi, "The Object-Oriented Hipermedia Design Model", Communications of the ACM, 38(8), August, 1995, pp.45-46.