

Comunicación e Informática para el Desarrollo de Software Multimediales Educativos.

Carolina Sandoval

Gonzalo Rojas

Resumen

El presente artículo presenta un enfoque comunicacional del proceso de desarrollo de software multimedial educativo, en respuesta a las falencias que los métodos tradicionales presentan para este tipo de aplicaciones. Como resultado de este enfoque, se presenta una propuesta de desarrollo para este tipo de aplicaciones, que se orienta tanto al equipo de trabajo como al proceso que éste lleva a cabo.

1. Introducción

We have forgotten that it's all about communication. (Watzman, 1998)

Las capacidades de presentación de información que permiten las tecnologías multimediales, han distraído la atención del usuario computacional del procesamiento interno de sus datos, centrándola en la forma en que les son comunicados los resultados.

Ante el desarrollo de un producto multimedial y, más aún, destinado a la educación, se puede observar claramente que el Ingeniero de Software no puede trabajar con un equipo integrado sólo por ingenieros y programadores. La comprensión del problema, la estrategia de solución y aun la implementación computacional de ésta requerirán la incorporación de profesionales tradicionalmente ajenos al desarrollo de software: los profesionales de la comunicación.

La inquietud por la integración de los campos de la comunicación y el diseño multimedial surge, desde el punto de vista comunicacional, ante la elaboración de un producto destinado a apoyar el aprendizaje de las personas y que, en muchos casos, constituye un proceso regido sólo por modelos de un alto nivel de abstracción, sin reparar en la compleja realidad en que el producto final será utilizado.

La importancia de los aspectos comunicacionales del software tradicionalmente ha sido soslayada por los especialistas en computación. La inversión de tiempo y recursos en tales aspectos se solía considerar como "un frívolo derroche", señala Nicholas Negroponte en su libro "Ser Digital", puesto que "los ciclos de computación eran muy valiosos y había que dedicarlos al problema, no a la persona." (Negroponte, 1995)

Llama la atención que el contacto con las personas se vea despojado de interés, en circunstancias que la construcción de un producto de ingeniería (en este caso, software multimediales) se centra en la exactitud para acercarse a la perfección, y su éxito real sólo puede ser ponderado en su relación con el usuario. Esta relación implica un proceso comunicativo, que no puede ser satisfactoriamente abordado sin manejar los conceptos y herramientas de la comunicación. Aminorar la importancia de este aspecto en el desarrollo de software educativos es una de las principales causas de la mala calidad de muchos de estos productos que, lejos de cumplir con sus objetivos pedagógicos, alejan al alumno de un medio de comunicación de alto potencial educativo. Como sostiene E.Barrett, "el computador ha textualizado el lugar de trabajo acercando las personas a él; ahora, el verdadero desafío es textualizar el computador y acercarlo a las personas" (Barrett, 1992)

El objetivo de este artículo es presentar los beneficios que reporta un enfoque de desarrollo de software educativo que, mediante la adopción de una visión comunicacional del proceso y del producto, permite

mejorar la calidad de ambos, obteniendo herramientas efectivas de apoyo al aprendizaje. En el capítulo 2 del artículo, se describirá y analizará la problemática asociada al adopción de un enfoque de desarrollo tradicional para la construcción de software educativos, que cuenta con una muy limitada participación de especialistas en comunicación, tradicionalmente alejados de este tipo de procesos. El enfoque comunicacional de desarrollo de software educativo se describe en el capítulo 3, a través de la presentación de propuestas para el producto y para el proceso de desarrollo, que consideran la integración con un rol protagónico de estos profesionales. Para finalizar, algunas conclusiones obtenidas de este trabajo se presentan en el capítulo 4.

2. Software multimedial educativo y desarrollo tradicional de software

2.1 Descripción del problema

Sobre el carácter multidisciplinario del desarrollo de software multimediales parece haber consenso. La incorporación de distintos medios audiovisuales como elementos relevantes de información en un producto computacional, obliga a considerar la participación de quienes poseen el conocimiento sobre la utilización de estos medios y sus efectos.

Sin embargo, la real integración de profesionales de la comunicación en este proceso a menudo sólo queda en las buenas intenciones. La gran mayoría de las tareas son asumidas por Ingenieros de Software, incluyendo, además de las tradicionales (especificación de requerimientos, diseño e implementación, pruebas y documentación), tareas como la generación de los contenidos audiovisuales, los aspectos de diseño de información y diseño gráfico. Especialistas en comunicación y arte sólo prestan una labor de asesoría, principalmente en aspectos gráficos, cumpliendo tareas complementarias al desarrollo del software (diseño de carátulas de CD's, redacción de manuales, etc.). En el caso particular de software educativos, se cuenta con la participación de un experto en las materias que se incluirán en el producto, generalmente un docente. Consideremos este escenario para analizar diversos aspectos del desarrollo de multimedios para la educación.

Los objetivos del proyecto y la descripción de usuarios son proporcionados por el docente, basándose en el paradigma de enseñanza tradicional profesor-alumnos, sin mayor consideración del nuevo contexto de aprendizaje que definen los aspectos computacionales y comunicacionales del software. La especificación de requerimientos, realizada en conjunto entre el docente y el Ingeniero de Software, obedece a similares criterios.

En lo referente a la organización del equipo de trabajo, el personal especialista en Comunicaciones y Arte se encuentra subordinado al Ingeniero de Software. La falta de un conocimiento cabal por parte de éste sobre la labor de aquellos profesionales desemboca en una asignación imprecisa de roles, tareas y ámbitos de acción, ocasionando una utilización deficiente de los recursos humanos disponibles y un desaprovechamiento de las capacidades profesionales individuales de los desarrolladores.

Para el Diseño de aplicaciones multimediales, existen herramientas que permiten modelar el dominio de información, como HDM (Garzotto, Paolini y Schwabe, 1993), RMM (Isakowitz, Stohr y Balasubramanian, 1995), DRM (Rojas, 1999), entre otros, pero carecen de amplia difusión. Ante el desconocimiento de algún modelo para multimedios o hipermedios (navegación entre medios), los Ingenieros de Software suelen optar por un simple boceto de navegación entre módulos de información; si conocen alguno, no es extraño que consideren suficiente el modelamiento realizado como para comenzar con la etapa de implementación.

En los dos casos anteriores, rara vez se considera un exhaustivo diseño de la información que se incluirá, olvidando el hecho que el estilo de presentación de la información en un software es (o debiera ser) muy distinto al estilo adoptado en libros, revistas u otros medios de comunicación tradicionales. Por ejemplo, los textos a incluir en la aplicación, al ser traspasados directamente desde libros u otras fuentes de información, carecen de una adecuación del estilo redaccional al contexto de enseñanza por

medio de un software. Para el usuario, el producto final resultará una extensa recopilación de información envasada.

La generación de contenidos (captura, digitalización y edición de información), dirigida por el Ingeniero de Software, se realiza según el criterio de obtener los datos requeridos, sin considerar la coherencia de cada unidad de información con el concepto original del software. El problema se acentúa cuando se trata de la generación de información audiovisual.

El Diseño de Interfaz se rige por los conocimientos que sobre la materia posea el Ingeniero de Software. Generalmente, esta etapa se limita a una definición de la metáfora general del software, no profundizándose en mayores niveles de detalle. Las decisiones más específicas se realizan durante la implementación del software, por parte del programador. Se cuenta, ahora sí en mayor medida, con la asesoría de los expertos en Comunicación y Arte, pero en una etapa tardía del desarrollo.

La implementación del software multimedial se basa en la construcción de prototipos. Al estar regida por requisitos imprecisos y un diseño deficiente, la productividad de las horas-programador es baja. Las decisiones de implementación de Interfaz Gráfica son las principales fuentes de retraso en esta etapa.

Los planes de pruebas y correcciones se limitan a verificar el correcto funcionamiento computacional del producto. Debido a la ausencia de una especificación rigurosa de requerimientos, las correcciones resultan drásticas y costosas, con gran cantidad de recursos invertidos en trabajo rehecho.

La documentación originada durante el desarrollo del producto recoge las limitaciones de todas las etapas anteriores, por lo que su calidad es deficiente.

2.2 Análisis del Problema

El escenario descrito revela un conflicto importante en el desarrollo de software multimediales educativos: si los integrantes del grupo de desarrollo no comprenden claramente su propio entorno de trabajo, su rol en el proceso y el modo de abordar las tareas que se les asignan, mal se puede esperar que el producto que obtengan cumpla el objetivo de constituirse en una herramienta eficaz de apoyo al aprendizaje.

El origen de esta problemática se puede sintetizar en dos puntos:

1. El problema que plantea el construir un software educativo como producto destinado a individuos.
2. El problema que plantea el desarrollo tradicional de software, empleado en la construcción del software educativo.

En el punto (1), se expone un dilema de concepto: cómo es concebido un software educativo. El que sea una herramienta tecnológica y se construya sobre la base de procedimientos complejos de programación, no debería justificar el modelamiento de su contenido al estilo de un software tradicional, de apoyo a procesos de negocio. El objetivo del software educativo es distinto: se crea como un instrumento de apoyo a la docencia, y la docencia es forzosamente un proceso interpersonal, de intercambio y dinámica.

Esta dinámica está dada por la transmisión de información, por el compartir saberes. En un software educativo, es importante que la forma en que se manifiesten estos saberes sea familiar para el grupo que los recibe. El considerar aspectos como el rango de edad, contexto de clases del grupo destino, el tipo y estilo de información a que están familiarizados (contenidos entregados en las aulas, medios de comunicación, otros software, etc.), permite equiparar el nivel de exigencia que impondrá el trabajo con el software con el asociado a la interacción con elementos que el usuario ya conoce.

El software multimedial educativo se concibe como un material de apoyo al proceso de enseñanza-aprendizaje, teniendo siempre en cuenta que tal proceso implica intercambios. El producto final deberá permitirlos, presentando contenidos en una forma fácilmente comprensible por el usuario.

Dentro de esta concepción, un objetivo fundamental a alcanzar por los desarrolladores es la flexibilidad del producto. Flexibilidad desde la perspectiva del Ingeniero de Software, impulsando un proceso de desarrollo que permita un modelamiento de datos no excesivamente restrictivo; flexibilidad desde el punto de vista del comunicador, permitiendo crear y exponer al usuario los contenidos a través de una serie de medios, en un ambiente integrado; y flexibilidad desde el punto de vista del usuario, creando un entorno de aprendizaje que le permita explorar los contenidos del software según sus propias reglas y prioridades, no limitado por la estructura navegacional establecida.

En síntesis, esta concepción apunta a obtener un software que permita al usuario construir el conocimiento, sin que el software mismo sea el límite de su aprendizaje. Un software que incentive su capacidad de aprender.

El punto (2) expone otro dilema de concepto: cómo se concibe un equipo desarrollador de software educativo.

Así como no es posible exponer una idea sin antes haber clasificado y clarificado los conceptos asociados en la mente, no es posible obtener un software educativo que refleje o comunique lo que se pretende sin tener claro quiénes lo construirán, qué tareas ejecutarán y los alcances de éstas en el proceso y el producto.

¿Qué sucedería si el desarrollador no sabe con claridad qué función cumple ni a quién debe acudir si necesita resolver algún problema no enfrentado anteriormente? ¿Y si, ante tales incertidumbres, en función de las dificultades que se encuentren en el camino, se deba replantear el sistema de trabajo durante el desarrollo de un producto, en una o varias oportunidades?

No se generan las condiciones necesarias para profundizar en las potencialidades individuales de los miembros del equipo desarrollador, tornando inestable todo el sistema. Finalmente, las tareas asignadas demoran mucho más de lo presupuestado. El proyecto se torna una carga para los desarrolladores y, además, deja de ser rentable. En el mejor de los casos, colapsa; en el peor, concluye con un producto de mala calidad; en ambos, no se logra cumplir los objetivos inicialmente propuestos.

3. Propuesta Comunicacional de Desarrollo de Software Multimedial Educativo

El sistema que aquí se plantea se basa en la adopción de un enfoque comunicacional del desarrollo de software educativo, que focaliza su atención en la coordinación y asignación de roles y tareas, contribuyendo a mejorar la calidad del producto final.

3.1 Propuesta para Equipo de Trabajo

"Parece claro que el resultado de cualquier proceso nunca será mejor que la gente que participa en él", afirman Logren y Stolterman (1999). En el desarrollo de software, esto implica que las capacidades y habilidades del desarrollador determinan la calidad del sistema final. De acuerdo a esto, los métodos de trabajo debieran ser vistos como herramientas para el desarrollo de las capacidades del desarrollador.

La propuesta que se presenta apunta a ese objetivo. Consiste principalmente en generar una micro-organización de trabajo, con un objetivo final (el producto) y una jerarquía horizontal claramente establecida, que promueva el respeto a la experiencia y a la asignación y cumplimiento de tareas.

Se requiere de un profesional encargado del área computacional y otro del área comunicacional, cada uno de ellos conduciendo y orientando a sus pares en el cumplimiento de su labor. Es decir, cada encargado deberá asignar funciones y derivar tareas específicas a los especialistas de su área.

Se promueve la conformación de grupos de trabajo, encargados de las etapas y/o conjuntos de tareas envueltas en la construcción del software. Cada integrante del equipo de desarrollo es asignado a uno o más grupos de trabajo, previa evaluación de su experiencia, habilidades y capacidades. El miembro más calificado asume la dirección de su grupo y la responsabilidad del cumplimiento de las tareas asignadas.

El encargado del área computacional organiza su equipo en grupos responsables de las etapas tradicionales del desarrollo de software (determinación de objetivos, especificación de requerimientos, diseño, implementación, pruebas y correcciones). El trabajo realizado en cada una de estas etapas se realizará en estrecha coordinación con los grupos de trabajo del área comunicacional.

El comunicador, por su parte, se encargará de conformar grupos de desarrollo de acuerdo a cada área de la comunicación relacionada con los tipos de información contenidos en el software, asignando, dirigiendo y controlando las tareas correspondientes a cada grupo:

- Grupo de redacción, encargado de captura y edición de textos, además de la determinación del estilo discursivo a adoptar.
- Grupo de captura y edición de medios audiovisuales, integrado por subgrupos encargados de cada tipo de medio en particular:
 - subgrupo de imágenes,
 - subgrupo de audio: voz, sonido, música, ambientación.
 - subgrupo de video y guiones
 - subgrupo de animaciones 2D y 3D,
- Grupo de diseño: Interfaz general e integración de los medios.

Los grupos computacionales y comunicacionales de trabajo deberán estar en permanente retroalimentación y coordinación. La mayoría de las tareas a realizar requerirá la participación de grupos pertenecientes a ambas áreas; otras, en cambio, serán realizadas sólo por especialistas de una de las áreas. En ambos casos, el informarse sobre lo que cada área realice es vital, para mantener la sincronía del trabajo en concordancia con los objetivos globales del software.

Además de las reuniones de trabajo realizadas por el equipo de desarrollo, se llevarán a cabo sesiones periódicas con el o los docentes participantes, a fin de verificar que la forma en que se exponen los contenidos y la forma en que se estima el usuario los percibirá, concuerdan con los fines pedagógicos propuestos.

3.2 Propuesta para Proceso de Desarrollo

Considerando el nuevo escenario planteado, se describirá a continuación el enfoque propuesto para cada etapa del desarrollo de un software multimedial educativo.

Para la determinación de objetivos del proyecto, se debe compatibilizar los objetivos académicos establecidos por el docente con los aspectos computacionales y comunicacionales que determinarán las posibilidades de concreción de tales objetivos. Estos aspectos serán indicados por los encargados de cada área. No se debe perder de vista que el problema principal a abordar en este tipo de proyectos y que orientará la definición de objetivos, no es la construcción de un software en sí, sino el proceso de enseñanza-aprendizaje al que tal construcción apoyará.

Se ha demostrado empíricamente que los software multimediales que obtienen mayor grado de aceptación por parte del usuario son aquéllos cuyo diseño considera los modelos mentales de la

mayoría de los usuarios, en comparación a los software diseñados sólo en base al pensamiento del cliente (en nuestro caso, el docente), que logran un éxito mucho menor (un ejemplo de estas experiencias se muestra en Nielsen, 1999b). De acuerdo a esto, se requiere que la descripción del usuario sea generada conjuntamente entre el docente, que aporta su visión del comportamiento del alumno-usuario en el contexto tradicional del aula, y los encargados del desarrollo, que aportan su visión del comportamiento del usuario frente a determinadas funcionalidades computacionales y aspectos comunicacionales de un producto multimedial..

Concretadas las etapas anteriores, el docente plantea sus requerimientos académicos. En conjunto con los desarrolladores, se establecen los requerimientos del producto, que se constituyen en directrices para su posterior diseño e implementación. El proyecto exigirá una continua validación de los requerimientos y el establecimiento de otros nuevos, para lo cual la participación del docente es fundamental. La documentación de los requerimientos se constituirá en un contrato informal, entre docente y desarrolladores, con el que se contrastarán los resultados de la implementación.

El Diseño del software se conformará de un Diseño Computacional y un Diseño Comunicacional, en estrecha coordinación. En el Diseño Computacional, se modelará el dominio de información, se determinarán los elementos de información requeridos, los aspectos relacionados con el procesamiento de datos y los aspectos navegacionales del producto. En el Diseño Comunicacional, se determinarán aspectos relacionados con la estructura de presentación de la información, como la cantidad final de unidades de contenido y los protocolos de conversión (reducción de textos a gráficos, caracterización de procesos de animación, entre otros).

Para el Diseño de Interfaz de Usuario, ambas áreas trabajan en conjunto. El efectivo cumplimiento de esta etapa libera al programador de las costosas decisiones de las cuales era injustamente responsable durante la implementación del software. Las tareas a realizar serán regidas por los tres principales criterios de diseño de interfaz de software, descritos por Wilding (1998):

1. Usabilidad: ¿puede el usuario aprender fácil y efectivamente interactuar con el sistema?
2. Funcionalidad: ¿qué funciones y controles están disponibles para permitir el uso óptimo del sistema?
3. Comunicación visual y estética: ¿cómo la apariencia visual y ubicación espacial de los elementos optimiza la funcionalidad?

En esta etapa, el establecimiento preciso de la metáfora del software es clave, puesto que los criterios de implementación y la generación de contenidos serán regidos por tal definición. La importancia del establecimiento de una buena metáfora, además, permitirá al usuario comprender, utilizar y recordar información más rápido, con mayor facilidad y satisfacción (Marcus, 1998).

Aspectos específicos a establecer en el Diseño de Interfaz de Usuario, referidos específicamente al ámbito comunicacional, son:

- Determinación del aspecto de la interfaz, de acuerdo a la caracterización de los usuarios finales.
- Determinación de las pautas gráficas para la construcción de ilustraciones y la edición de videos y animación.
- Determinación de ubicación de elementos de información y auxiliares en el espacio de trabajo (layout).

La generación de contenidos está a cargo de los grupos de comunicación. La captura y edición de datos se realizará de acuerdo al concepto original del software y a los requerimientos planteados, evitando generar información innecesaria, con el consiguiente ahorro de recursos. Las características internas y de presentación de los datos pasa de ser sólo un complemento a los objetivos del producto, a constituirse en factor determinante en la consecución de éstos. Se recalca la importancia del estilo de

redacción de texto a incluir en el software, el que debe ser acorde tanto a los objetivos pedagógicos como también al medio de enseñanza. Existen muchas diferencias entre un medio impreso (libro, revista) y un medio "digital", sobre todo por el carácter navegacional de este último. Esto hace necesario adoptar un estilo discursivo que utilice las fortalezas de cada uno y minimice sus debilidades (Nielsen, 1999b; 1999c).

Al ser la única tarea del programador y estar regida por una especificación de requerimientos precisa y por un diseño prolijo, la implementación del software consumirá menor tiempo. La necesidad de generar prototipos y un plan intensivo de pruebas con el usuario es aún más pronunciada en el desarrollo hipermedial que en el desarrollo de software tradicional, porque la tolerancia del usuario a eventuales errores en este tipo de aplicaciones es muy baja (Isakowitz, Stohr y Balasubramanian, 1995). Se sugiere, entonces, adoptar en esta etapa la modalidad de prototipos. Su carácter preventivo de errores y su continua validación de requerimientos, evitará costosas reparaciones. Para la implementación de la Interfaz de Usuario, se requerirá la permanente asesoría de miembros del grupo comunicacional involucrado en su diseño, además de otros miembros del área encargados de la generación de elementos gráficos que no hayan sido considerados con antelación.

El plan de pruebas se orientará al correcto funcionamiento de todos los módulos del software, de los elementos de interfaz y de navegación. Aparte de las pruebas realizadas por los propios miembros del equipo, se sugiere la participación de 2 a 3 personas ajenas al desarrollo el producto, idealmente usuarios con características similares a las establecidas en la descripción de usuario inicial.

Durante todo el desarrollo, cada grupo de trabajo gestionará su propia documentación. Los encargados de cada área recopilarán esta información, que conformará la documentación del proceso completo. Los manuales de usuario serán confeccionados por grupos de ambas áreas, encargando su redacción a un grupo de comunicaciones. La confección de material anexo ilustrativo (carátulas, sellos, envase del producto) también será labor de un grupo de esta área.

4. CONCLUSIONES

El desarrollo de multimedios para la educación es un proceso que involucra una alta inversión. Por ello, su concreción se justifica sólo si los beneficios de tal inversión en el proceso enseñanza aprendizaje superan significativamente los beneficios potenciales de la aplicación de herramientas educativas tradicionales, de costo inferior.

De acuerdo a lo anterior, cabe preguntarse cuál es la real ventaja comparativa de un software multimedial con respecto a tales herramientas ya existentes.

Al escribir, los autores convierten su conocimiento, existente como una compleja estructura de red, en una representación externa. Los medios tradicionales, como material impreso, cintas de audio o video, sólo permiten esta representación en una forma esencialmente lineal. Los hipermedios, en cambio, permiten una representación de la información más acorde a los procesos mentales de lectura y escritura, permitiendo crear estructuras de información no-lineales, mediante el uso de links. (Ginige, Lowe y Robertson, 1995)

Es esta característica la que debe ser explotada intensivamente para obtener un software multimedial que permite cumplir con su objetivo principal, el de herramienta de apoyo al aprendizaje. Los profesionales del área de Ingeniería de Software no han entendido completamente los aspectos cognitivos involucrados en el desarrollo de sistemas hipermediales. Las estrategias y metodologías de desarrollo tradicional, pudiéndose aplicar en este tipo de productos, fallan en un aspecto esencial: la satisfacción del usuario.

La importancia del enfoque comunicacional presentado en este artículo radica en los beneficios que presenta tanto para el proceso de desarrollo de multimedios para la educación, como para la calidad del

producto obtenido.

La definición precisa de roles, funciones, tareas y responsabilidades de cada uno de los participantes del proyecto, incluyendo docentes, desarrolladores y usuarios, favorece la comunicación entre ellos y la coordinación en el desempeño de tareas, puesto que cada uno sabe exactamente cuál es su función y cómo se relaciona con las restantes. La coordinación entre los distintos grupos se simplifica, lográndose coherencia en toma de decisiones y aceptación total de éstas. La productividad del equipo de desarrollo se ve aumentada y el tiempo de desarrollo se reduce notablemente.

El enfoque que se presenta en este artículo fue aplicado con éxito en el desarrollo de un software multimedial destinado a la enseñanza del idioma Inglés para estudiantes de carreras universitarias de idiomas extranjeros. Los objetivos de este producto son el desarrollar la comprensión oral del idioma Inglés y evaluar el grado de aprendizaje, por parte del alumno, de los contenidos revisados en el aula. El producto obtenido satisface ampliamente las expectativas del usuario. La interfaz es fácil de asimilar y se enfoca en la problemática de enseñanza más que en la presentación de contenidos. El alumno es participante activo de su propio aprendizaje y no sólo un receptor pasivo de información digitalizada. Todo esto, soportado por una ejecución computacional de alta calidad. En cuanto al proceso de desarrollo, los encargados de cada área, con sus tareas y roles claramente definidos, lograron establecer un ambiente de desarrollo de alta colaboración, obteniéndose un producto de alta calidad en un tiempo breve.

Finalmente, cabe mencionar que una exitosa implementación del enfoque presentado se basa en un equipo de trabajo conformado por profesionales de las áreas computacional y comunicacional que muestren una buena disposición al trabajo multidisciplinario y un alto espíritu de colaboración. Una "declaración de principios" nada novedosa, pero muchas veces ignorada.

5. REFERENCIAS BIBLIOGRÁFICAS

1. Barrett, E. (1992). *Sociomedia: Multimedia, Hipermedia, and the Social Construction of Knowledge*. Massachusetts Institute of Technology, MIT Press, Cambridge, MA, USA
2. Garzotto, F., Paolini, P., Schwabe, D. (1993). *HDM: A Model-Based Approach to Hypertext Application Design*. ACM Transactions on Office Information Systems, Vol.11 N°1, pp.1-26. >
3. Ginige, A., Lowe, D.B., Robertson, J. (1995). *Hypermedia Authoring*. IEEE Multimedia Magazine Vol.2, N°4, pp. 24-35.
4. Isakowitz, T., Stohr, E.A., Balasubramanian, P. (1995). *RMM: A Methodology for Structured Hypermedia Design*. Communications of the ACM, Vol.38, N°8, pp.34-44.
5. Löwgren, J., Stolterman, E. (1999). *Design Methodology and Design Practice*. ACM Interactions, Vol.6 N°1, pp.13-20.
6. Marcus, A. (1998). *Metaphor Design for User Interfaces*. Proceedings of the conference on CHI 98 summary: human factors in computing systems, pp 129-130.
7. Negroponte, N. (1995). *El Ser Digital*. Editorial Atlántida S.A., Buenos Aires, Argentina.
8. Nielsen, J. (1999a). *Differences Between Print Design and Web Design*. Jakob Nielsen's Alertbox, January 24, 1999. <http://www.useit.com/alertbox/990124.html>
9. Nielsen, J. (1999b). *Do Interface Standards Stifle Design Creativity?*. Jakob Nielsen's Alertbox, August 22, 1999. <http://www.useit.com/alertbox/990822.html>
10. Nielsen, J. (1999c). *Be Succinct! (Writing for the Web)*. Jakob Nielsen's Alertbox, March 15, 1997. <http://www.useit.com/alertbox/9703b.html>
11. Rojas, G. (1999). *Modelo DRM para Sistemas Hipermediales*. Revista Electrónica "Ingeniería Informática", DIICC - Universidad de Concepción. <http://www.inf.udec.cl/revista/edicion4-2/grojas/grojas.html>
12. Watzman, S. (1998). *The Art of the Interface: Visual Ideas, Principles and Inspiration for Interface Designers*. Proceedings of the conference on CHI 98 summary: human factors in computing systems, pp.151-152.

13. Wilding, C. (1998). *Practical GUI Screen Design: Making It Usable*. Proceedings of the conference on CHI 98 summary: human factors in computing systems, pp. 125-126.