

La utilización de la Ingeniería de Software en hipermedia

Resumen

En la actualidad el uso de la hipermedia se ha convertido en la vía de acceso a la información más utilizada a nivel mundial, y se ha visto como el progreso en las Tecnologías de la Información y la Comunicación (TIC's) son cada día más evidentes. El objetivo de esta investigación es demostrar cómo la Ingeniería de Software Orientada a Objetos puede contribuir al desarrollo más eficiente de software para hipermedia, sea en el ámbito educativo, informativo o comercial. Para el estudio se revisaron varias de las metodologías orientadas a objetos, obteniendo que existen muy pocas dirigidas al desarrollo específico de hipermedia.

Palabras clave: Software, hipermedia, ingeniería de software, orientación a objetos, desarrollo de software, software educativo, software de información, software comercial.

Abstract

At present the use of hypermedia has become the most used means of access to information worldwide and we have seen how the progress in ICT is increasingly evident. The objective of this research is to demonstrate how Object-Oriented Software Engineering can contribute to more efficient development of hypermedia software, be it in the educational, informational or commercial sphere. Several object-oriented methodologies were reviewed for this study, with the conclusion that there is very little that is specifically oriented to the development of hypermedia.

Keywords: Software, hypermedia, software engineering, object orientation, software development, educational software, information software, commercial software.

Recibido: Septiembre, 2011

Aceptado: Diciembre, 2011



Jesennia Cárdenas Cobo, Lsi.¹

Unidad Académica Ciencias de la Ingeniería
jcardenasc@unemi.edu.ec



INTRODUCCIÓN

La presente investigación surge a partir de la experiencia docente dirigiendo proyectos de grado en la carrera de Ingeniería en Sistemas de la Universidad Estatal de Milagro, donde un número considerable de ellos se han orientado al ambiente hipermedia, por la tendencia que existe hacia esta área de desarrollo [1]. Pero al revisar el tiempo de implementación y de analizar sus resultados, se puede determinar que la ingeniería de software tradicional que aplican los estudiantes no ayuda en la realización eficiente del producto, puesto que se han encontrado proyectos finales que carecen de una organización de su información, difíciles de navegar, diseños poco atractivos y en la mayoría de los casos lentos. Por este motivo, se ha visto la necesidad de crear una metodología específica para software hipermedia que contribuya al análisis, diseño, desarrollo e implementación de una forma eficiente y eficaz.

El trabajo se centra en una metodología basada en Ingeniería de Software Orientada a Objetos, enfocada al desarrollo del mismo ya sea en el ámbito educativo, informativo o comercial. Sin dejar de lado la posibilidad de decir en forma generalizada que serviría para cualquier tipo de software orientado a hipermedia, considerando que para desarrollarlo siempre van a estar presente los fundamentos básicos referentes al ciclo de vida: análisis, diseño, desarrollo, implementación y pruebas; pero a este ciclo es necesario agregar un sinnúmero de aspectos de análisis y diseño, y determinar los requerimientos especiales que tiene el construir software hipermedia. El estudiante al iniciar sus proyectos se encuentra con un determina-

do número de ideas, no conoce si empezar a diseñar o empezar ya con la programación y ¿qué pasó con las imágenes?; en muchos de los casos termina repitiendo la programación o los diseños, cuando lo único que debería cambiar es el contenido. ¿Y esta opción me debería mostrar qué cosa? O ¿Por qué opciones debería llegar al mismo lugar? Siendo ese el principio básico de la ingeniería de software orientada a objetos: la reutilización de código.

Además, con la aparición de las plantillas se está dejando de lado la innovación y exclusividad del software; “que nos facilita la vida”, totalmente de acuerdo; pero la adaptación de esas plantillas a los requerimientos exclusivos del software es lo que se pretende alcanzar. Se ve la necesidad que los estudiantes de Ingeniería en Sistemas puedan crear plataformas exclusivas, que bien pueden estar basadas en estándares, pero sin dejar de lado lo inédito de un sitio web, que servirá a la entidad para quien fue construido como el portal para comunicarse con los usuarios ligados a la aplicación web, en el lugar que ellos se encuentren.

El crecimiento exponencial de enlaces y de navegación a través de la información, complica la creación de sistemas hipermedia, pues no les permite ser estructuras lineales como los sistemas tradicionales de escritorio, es precisamente este crecimiento que da apertura a la ingeniería de software orientada a objetos que sugiere la reutilización de componentes para hacer más sencillo su diseño y posterior mantenimiento. Es posible que permitan hacer un software de calidad en corto tiempo; pero ¿qué es calidad dentro del concepto de desarrollo de software?; a criterio del autor, es que la

aplicación haga lo que tiene que hacer en el menor tiempo posible, que incremente la eficiencia de la organización o institución, y que brinde la confianza al usuario que lo utiliza. La calidad es medida dentro de la ingeniería de software desde dos puntos de vista: de acuerdo a la satisfacción del usuario final y según su tiempo de respuesta; convirtiendo al usuario final en el evaluador más estricto del proceso de desarrollo del software.

A pesar de que la Ingeniería de Software Orientada a Objetos ha resuelto ciertos problemas que se han presentado desde su aplicación, tales como la interacción binaria, la modularidad, entre otros, las soluciones que ofrecen algunas metodologías creadas ayudan a aplacar estos problemas y recompensan los sacrificios hechos al momento del diseño, pues se obtiene como resultado un producto más eficiente y eficaz.

La modularidad se ve afectada cuando se diseña bajo una orientación a objeto; sin embargo, el modelo de dominio aplicado en este caso mejora el ordenamiento de los componentes utilizados, ordenando las actividades de desarrollo, provocando que la reutilización de componentes sea más sencilla, y evitando que se empiece a desarrollar desde la nada.

Marco referencial

En la Universidad Estatal de Milagro dentro de la red curricular de la carrera de Ingeniería en Sistemas existe un módulo denominado Ingeniería de Software Orientada a Objetos, donde se aplica la metodología de **MERODE** (Model-driven, Existence-dependency Relation Object-oriented, Development) para el desarrollo de software, que fue desarrollada en la Uni-

versidad de Leuven, se basa en la creación de tres vistas de un mismo concepto: **EDG** (Existence Dependency Graph: Gráfico de la Dependencia de Existencia), **OET** (Object Event Table: Tabla de Objetos Eventos) y **FSM** (Finite State Machine: Máquina de Estado Finito), se utilizará el mismo concepto de orientación a objetos para la creación de una metodología que además sea exclusiva para desarrollo de hipermedia; y que permita a los estudiantes de sistemas diseñar y desarrollar software en esta área [2].

La metodología **MOOM** (Metodología Orientada a Objeto Multimedia), que en su estructura pretende involucrar estos aspectos tan importantes en el mundo actual de la informática, ha sido revisada en forma exhaustiva para proponer adaptaciones que permitan adecuar una nueva metodología que sirva para cualquier tipo de software hipermedia y no solo educativo o informativo.

OOHDM (Object-Oriented Hypermedia Design Model)

Ciertos autores lo describen como un lenguaje de modelado orientado a aplicaciones hipermedia; sin embargo, al analizar su metodología se

synetiza en un proceso de desarrollo, relacionado con **UML** que es un lenguaje de modelado independiente, OOHDM lo usa en su proceso, muy similar a la secuencia de **RUP** (Rational Unified Process), con la diferencia de que RUP está orientado para el desarrollo de cualquier aplicación, mientras que OOHDM hace énfasis en la organización de los objetos y los aspectos de la navegación para cumplir su propósito [3], [4].

El método de desarrollo de hipermedia OOHDM [5], [6], [7], [8] introduce el modelado orientado a objetos en el desarrollo de hipermedia. En OOHDM se modela la navegación a través del diagrama de clases navegacionales y del diagrama de contextos.

El diagrama de clases navegacionales es una vista del modelo estructural. Cada conjunto de usuarios tiene una en particular, y sus elementos son clases y asociaciones. Las clases reflejan la estructura de un documento de hipermedia, son definidas como una vista de las clases conceptuales y se construyen a partir de lenguajes de consulta como los especificados en [9] y [10]. Las asociaciones del diagrama de clases navegacionales pueden corresponder a en-

laces (anchor) o estructuras de acceso que son incluidas como atributos de las clases.

En la Figura 1 se muestran las etapas consideradas dentro del modelo OOHDM.

Los **ADV's** (Vista de Datos Abstracta) son diagramas basados en máquinas de estados finitas, usados por UML y OMT con algunas extensiones. Estos modelan el comportamiento de los objetos basados en eventos.

En OOHDM existe un conjunto de tipos predefinidos de clases navegacionales: nodos, enlaces y estructuras de acceso. La semántica de los nodos y los enlaces son las tradicionales de las aplicaciones hipermedia, y las estructuras de acceso, tales como índices o recorridos guiados, representan los posibles caminos de acceso a los nodos.

EORM (Enhanced Object Relationship Methodology)

Es una metodología definida por un proceso iterativo que se concentra en el modelado orientado a objetos, por la representación de relaciones entre los objetos (acoplamientos) como objetos, es por ello que fue una de las primeras propuestas para Web centrada en el paradigma de la orientación a objetos. Se basa

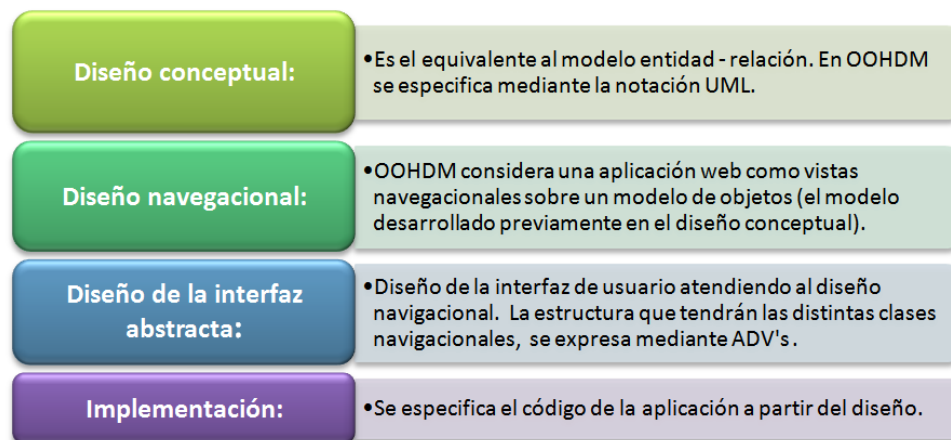


Figura 1. Etapas de OOHDM, para el desarrollo de software hipermedia [11].

en muchas de las ideas que se definen en HDM, pero las traslada a la orientación a objetos. Distingue dos tipos de relaciones: Relaciones de generalización y relaciones definidas por el usuario. Mientras que los primeros se concentran en la semántica asociada entre ellas, los segundos confían totalmente en la especificación del usuario.

Ventajas: Encajamiento de relaciones semánticas en construcciones extensibles, tener bibliotecas reutilizables.

La metodología consta tres fases según el siguiente diagrama de flujo:

Fase de Análisis: Se trata de orientar objetos al sistema sin considerar los aspectos hipermediales del mismo, obteniéndose para ello un Modelo de Objetos con la misma notación utilizada en OMT, que refleje la estructura de la información (mediante clases de objetos con atributos y

relaciones entre las clases) y el comportamiento del sistema (a través de los métodos asociados a las clases de objetos).

Fase de Diseño: se refleja la estructura de la información (modelo abstracto hipermedial compuesto de nodos y enlaces) y las posibilidades de navegación ofrecidas por el sistema; sobre dicha estructura, para lo cual existirá un repositorio o librería de clases de enlaces, donde se especifican las posibles operaciones asociadas a cada enlace de un hiperdocumento, que serán de tipo crear, eliminar, atravesar, siguiente, previo, entre otros; así como sus posibles atributos (fecha de creación del enlace, estilo de presentación en pantalla, restricciones de acceso, etc.).

Fase de Construcción: Aquí se transforman los esquemas en código y son guardados en una Base de

Datos Orientada a Objetos, se elaboran formularios de consulta de las clases con la ayuda de un editor gráfico de interfaces. Se genera el código fuente (por ejemplo en C#) correspondiente a cada clase y se prepara la Interface Gráfica de Usuario.

RNA (Relationship Navigational Analysis)

Es un método de Análisis de Navegación Relacional, que define una secuencia de pasos que se utilizarán para el desarrollo de la Web. Es especialmente útil para uso de Aplicaciones Web creadas en base de sistema de herencia. En este método aparecen cinco fases, las que se muestran en la Figura 2.

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

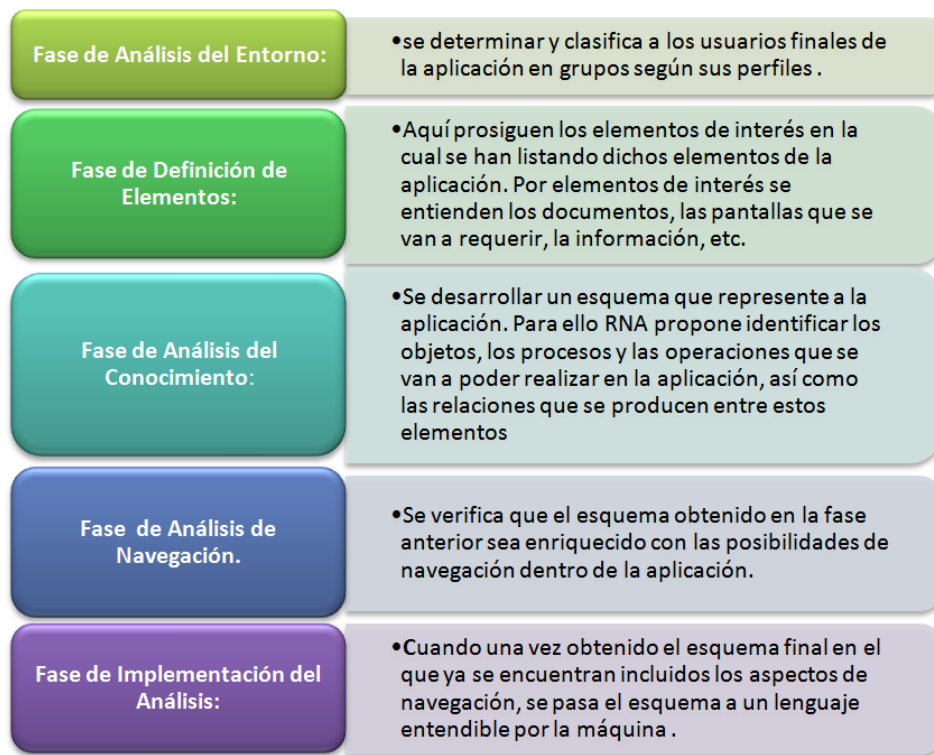


Figura 2. Etapas de RNA, para el desarrollo de software hipermedia

y de los elementos de interés.

RMM (Relationship Management Data Model, RMDM)

Es una metodología para el desarrollo de aplicaciones de hipermedia que tienen una estructura regular definida mediante entidades y relaciones entre éstas. Ejemplos típicos

de información estructurada a los que se puede aplicar RMM son catálogos y bases de datos. Los modelos construidos con las primitivas de RMM son llamados Modelo de Datos de Administración de Relaciones (Relationship Management Data Model, RMDM) [12].

La navegación es mo-

delada a través de enlaces unidireccionales, enlaces bidireccionales, y estructuras de acceso; éstas pueden ser índices, visitas guiadas, índice-visita-guiada y grupos (ver Figura 3). A continuación se describen brevemente cada una de ellas.

Enlaces: Conectan una

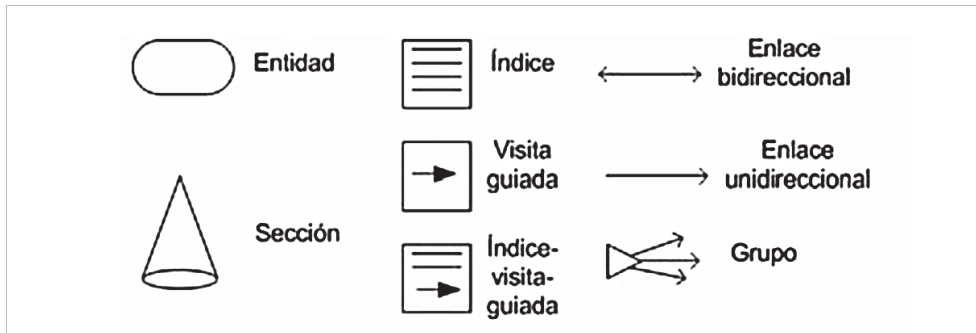


Figura 3. Notación de RMM.



Figura 4. Etapas de WSDM, para el desarrollo de software hipermedia.

entidad con sus secciones, o las secciones de una misma entidad entre sí. Las conexiones pueden ser bidireccionales o unidireccionales, y se denominan enlaces estructurales.

Grupo: Es una construcción similar a un menú; está formado por una lista de enlaces.

WSDM (Web Site Design Method)

Es un método de diseño para sitios web donde hay un acercamiento al usuario que define los objetos de información basado en sus requisitos de información para el uso de la Web. En este método se define una aplicación Web a partir de los diferentes grupos de

transformaciones XSLT [18]. WebML cuenta con una herramienta de desarrollo llamada WebRatio que permite construir sus modelos y generar el código de la aplicación hipermedia mediante transformaciones XSLT.

Las etapas de desarrollo de WebML se centran en la construcción de 4 modelos: estructural, de hipertexto, de presentación y de personalización.

El modelo estructural o de datos representa las entidades y atributos relevantes en el dominio del problema. La notación que sigue WebML es entidad - relación (E-R).

El modelo de hipertexto es llamado site view, que describe las vistas del modelo

ces entre páginas.

El modelo de presentación en WebML se efectúa a través de la aplicación de hojas de estilo XSL a documentos XML que representan una instancia del modelo navegacional, y el resultado de la transformación es un documento HTML [19].

El modelo de personalización está conformado por las entidades predefinidas usuario y grupo. Las características de estas entidades se usan para mostrar contenido individualizado.

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

Es un método que Desarrolla Diseño en Panoramas (es-

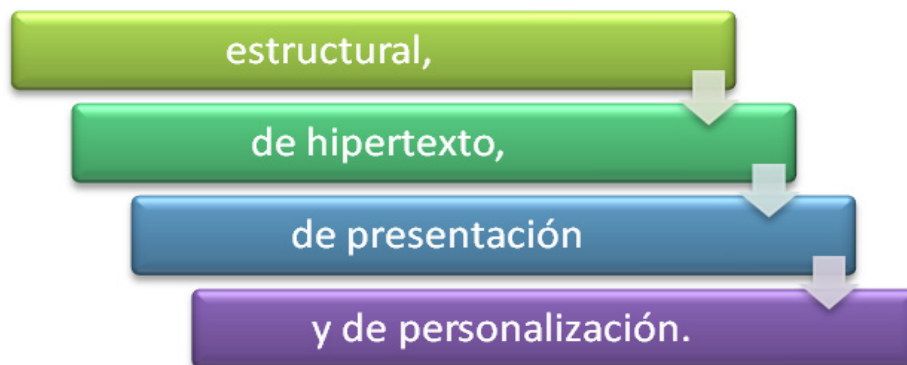


Figura 5. Las etapas de desarrollo de WebML.

usuarios que vaya a reconocer el sistema [13].

WebML (Web Modeling Language)

A diferencia de los otros métodos, WebML [14], [15], [16] no hace énfasis en ser un método de desarrollo de hipermedia, sino un método para diseñar sitios web. WebML utiliza XML [17] para representar los modelos generados en cada etapa del desarrollo, lo que permite que sean fácilmente consumidos por generadores de código fuente que usan

estructural que estarán publicadas en el sitio web. El site view se divide en dos modelos, el de navegación y el de composición.

El modelo de composición define que páginas estarán contenidas en el sitio web, y que contenido tiene cada página. En el modelo se construyen nodos de hipermedia con enlaces estructurales o internos, que no son convertidos en hiperenlaces. Los elementos son llamados unidades.

El modelo de navegación sirve para especificar los enla-

cenario) Orientado a Objetos en Hipermedia. Presenta la necesidad de disponer de un proceso que permita capturar las necesidades del sistema. Para ello, propone el uso de escenarios.

El proceso de definición de requisitos parte de la realización de un diagrama de contexto tal y como se propone en los diagramas de flujos de datos (DFD) de Yourdon (1989) [20]. En este diagrama de contexto se identifican las entidades externas que se comunican con el sistema, así como

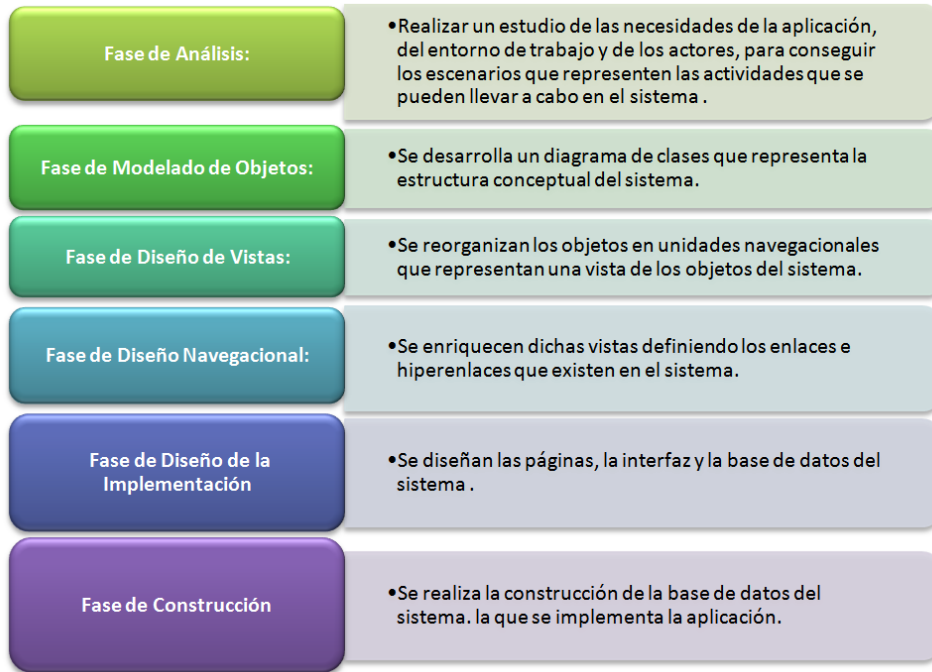


Figura 6. Etapas de SOHDM, para el desarrollo de software hipermedia

Metodología	Técnica de Modelado	Notación	Proceso completo del Software	Aseguramiento de Calidad del proceso del Software
HDM (Hypermedia Design Methodology)	Relacional	Entidad - Relación	No	No
RMM (Relationship Management Methodology)	Relacional	Entidad - Relación	No	No
EORM (Enhanced Object Relationship Methodology)	Orientado a Objetos	OMT	No	No
OOHDM (Object-Oriented Hypermedia Design Model)	Orientado a Objetos	OMT/UML, propietaria y AdvS	No	No
SOHDM (Scenario - based Object-oriented Hypermedia Design Methodology)	Escenarios y Visitas Orientadas a Objetos	Escenarios, diagramas de clases y visitas orientadas a Objetos	No	No
WSDM (Web Site Design Method)	Relacional y Orientadas a Objetos		No	No
RNA (Relationship Navigational Analysis)			No	No

Figura 7. Resumen de las metodologías revisadas.

los eventos que provocan esa comunicación. La lista de eventos es una tabla que indica en qué eventos puede par-

ticipar cada entidad. Por cada evento diferente, SOHDM propone elaborar un escenario. Estos son representados grá-

ficamente mediante los denominados SACs2 (Scenario Activity Chart).

Cada escenario descri-

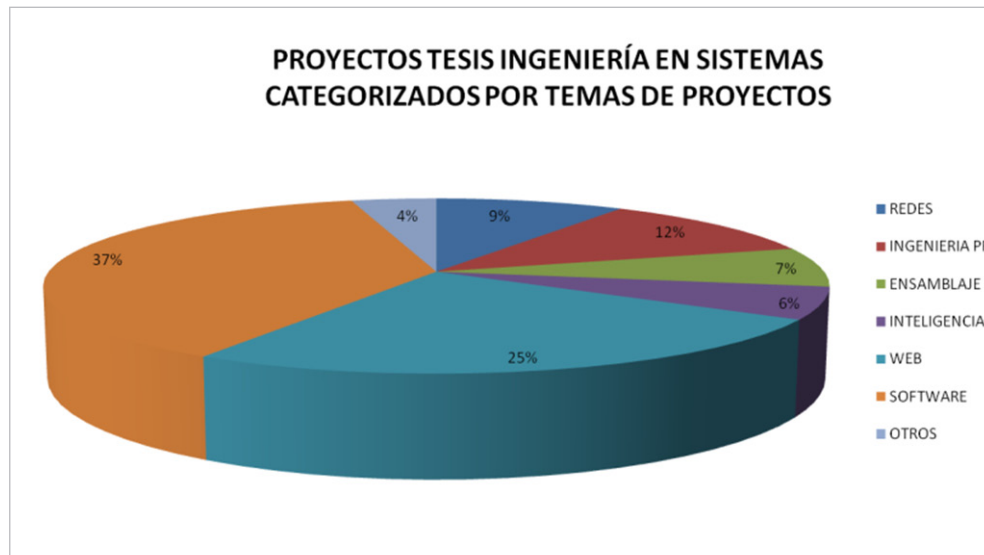


Gráfico 1. Proyectos de grado de la carrera Ingeniería en Sistemas.
Fuente: Unidad Académica Ciencias de la Ingeniería.

be 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.

Consiste en seis fases: análisis del dominio, modelado del objeto, diseño de la visión, diseño de la navegación, diseño de la puesta en práctica y construcción.

Esta metodología tiene semejanzas con, OOHDM y EORM [21] donde se diferencian en el uso de panoramas, que describen las actividades en los acontecimientos y primitivas de flujos de actividades. Los panoramas se definen en la fase de análisis y se utilizan para modelar los objetos. En la Figura 6 se detallan sus fases.

En conclusión, la metodología SOHDM es una propuesta nueva que cubre en mayor parte todas las fases del proceso de desarrollo, aunque no toma en cuenta la implantación y las pruebas, proponiendo un proceso cíclico de tal forma que al realizar una fase se puede regresar a alguna de las anteriores para refinarla y adaptarla mejor.

La propuesta es hasta ahora la única que tiene en cuenta aspectos como la especificación de requisitos haciendo uso de los escenarios. Otra ventaja, es un proceso sencillo de seguir, no obstante su nomenclatura es muy cerrada. Además, es una propuesta donde se hacen uso de técnicas de modelado orientado a objetos, algo muy significativo ya que es adecuado para el desarrollo de este tipo de aplicaciones.

La Tabla 1 presenta un resumen de las metodologías revisadas, en la cual se toma como parámetro de comparación la técnica de modelado, la notación, la complejidad de su desarrollo y la seguridad en el proceso de desarrollo.

Cada una de estas metodologías aporta elementos importantes al desarrollo de software de tipo hipermedial; sin embargo, se observa tres grandes debilidades: la aplicación de normas de calidad para las mejores prácticas en la producción de software, la ausencia de fases del proceso del software y la aplicación de modelos pedagógicos que sustentan el diseño educativo [22].

Los métodos centran su mayor interés en aspectos de navegación y presentación respecto a la semántica de los modelos para capturar el problema de desarrollo en entorno web, de allí la necesidad de seleccionar uno de ellos y darle el valor agregado sobre calidad de software, poniendo énfasis en la navegabilidad y la facilidad al usuario. Por este motivo, la metodología tomada como sustento para el presente estudio es OOHDM; pues es la más validada [23] y porque su estructura se fundamenta en varios paradigmas de la ingeniería actual.

Propuesta

Sobre la base del análisis de las diversas metodologías analizadas, se plantea el siguiente esquema de trabajo, para producir software hipermedia de tipo educativo, comunicativo o comercial, con un añadido de calidad que permita generar aplicaciones en el menor tiempo posible, pero cumpliendo las especificaciones y estándares de los usuarios.

Se proyecta disponer de una metodología para la implementación de software

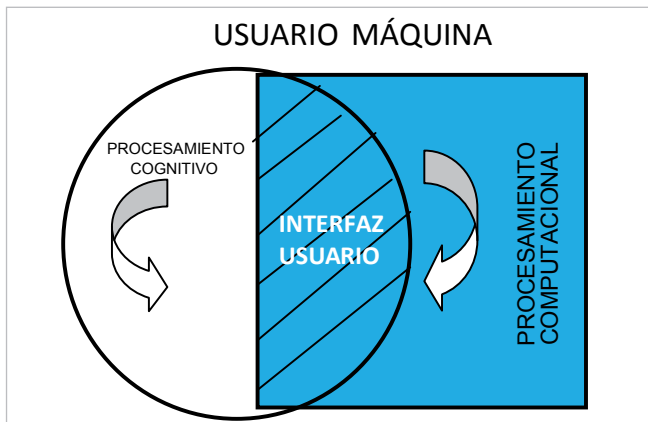


Gráfico 2. Modelo simplificado de la interacción usuario máquina, dentro del modelo DESHI

hipertexto que permita a los desarrolladores la facilidad de uso, así como disminuir el tiempo de respuesta, evitar la duplicidad de código, trabajar en función de texto, imágenes u otros recursos y no de la estructura que será una

una aplicación web es una tarea compleja, se debe especificar los objetos que serán implementados y cómo estos objetos interactúan entre sí. La dedicación de tiempo a esta etapa debe ser considerada como una inversión.

distintas, siempre recaerán en las etapas de análisis, diseño, desarrollo e implantación; pero al momento de desarrollar aplicaciones web, se debe poner mayor énfasis en alguna de ellas, o considerar la navegabilidad como factor decisivo.

Por eso, se presenta esta metodología sin hacer un cambio radical a la forma en cómo se ha venido trabajando en aplicaciones de escritorio, la propuesta es producto del análisis de los proyectos de grado desarrollados en la carrera de Ingeniería en Sistemas, donde se ha detectado que la metodología utilizada no es la ideal, pues en la fase de desarrollo, un 97% de ellos no culminan en los plazos especificados.

Es evidente que el desa-

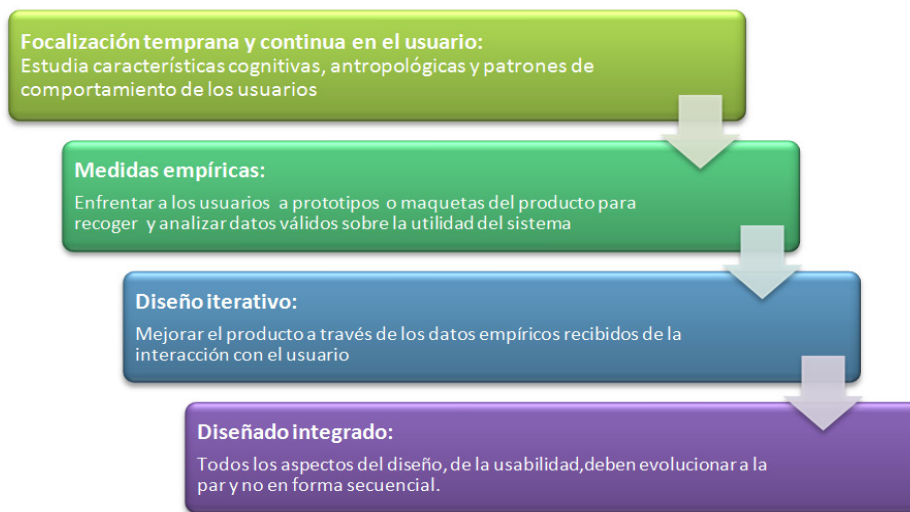


Figura 7. MDiseño centrado en el usuario

constante en el desarrollo. Sin embargo, como se encontró mayor dificultad en las etapas de análisis y diseño en la investigación realizada a los proyectos desarrollados como tesis de grado, se presenta un esquema que enfatiza sólo estas dos etapas, las cuales generan el retraso del 60% del desarrollo del proyecto.

Desarrollar la interfaz de

Esta inversión de tiempo está ampliamente justificada, no sólo porque simplifica el proceso de desarrollo, facilitando el trabajo del equipo encargado de cada capa de la aplicación, sino también durante su mantenimiento y eventual extensión [24].

El ciclo de vida de las aplicaciones de software, por más que le den denominaciones

rollo de software web ha ganado terreno en los tipos de proyecto que se desarrollan, debido a la disponibilidad de estos tipos de aplicaciones, tal como se lo muestra en el Gráfico 1. Donde el desarrollo web ocupa el segundo lugar con un 25% de proyectos desarrollados en esa área, y con una mínima diferencia de los proyectos de software tradi-

cionales (aunque la mayoría fueron realizados al inicio de la carrera).

Con todos estos antecedentes, la metodología propuesta se denomina: **DESHI (Desarrollo de software Hipermedia).**

Este método hace imprescindible la necesidad de incluir en su ciclo de vida la fase de evaluación, para que permita la participación con el usuario en la construcción de software hipermedia. Adicionalmente se incluyen ciertos aspectos que se tomarán en cuenta dentro del proceso:

- Desarrollo centrado en el usuario y en sus necesidades.
- Equipo de trabajo pluridisciplinar.

Desarrollo centrado en el usuario y en sus necesidades.

Las aplicaciones hipermedias tienen como objetivo conseguir que el usuario realice una actividad en forma más efectiva y eficiente, como por ejemplo comprar o vender algún producto, aprender algo, comunicarse con alguien o simplemente divertirse a través de un diálogo entre usuario y máquina el cual se materializa a través de la interfaz con el usuario.

El diseño centrado en el usuario, término acuñado por Gould y Lewis en 1985, tiene como objetivo maximizar la usabilidad del producto, lo cual asume 4 principios básicos [25]. Ver Figura 8.

De acuerdo a la norma ISO 13407: Proceso de Diseño Centrado en la Persona para Sistemas Interactivos (Human-centred desing process for interactive system), las actividades involucradas en el ciclo de vida de sistemas interactivos son: analizar y especificar los contextos de uso o escenarios, especificar los requisitos orga-



Figura 8. Actividades de diseño centrado en el humano y sus relaciones [26]

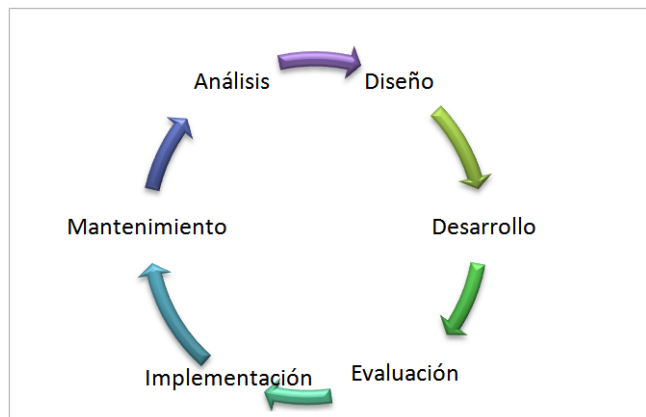


Figura 9. Ciclo de vida de desarrollo de software hipermedia

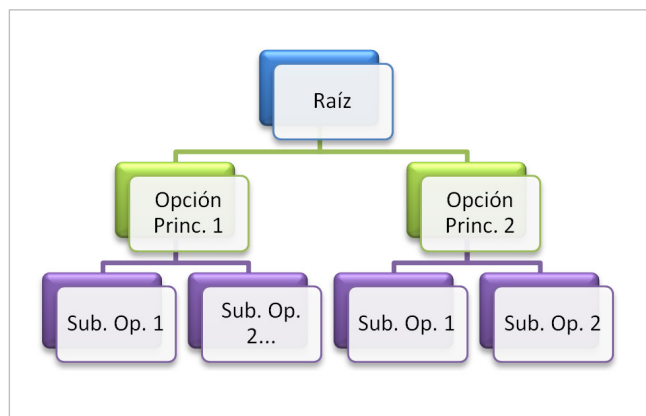


Figura 10. Mapa de Sitio.

nizativos así como los de los usuarios, producir soluciones de diseño y evaluar esas soluciones frente a los requisitos. Véase Figura 9.

Es en este contexto como el modelo propuesto permite

secuenciar tales actividades para contar con la participación del usuario durante todo el proceso de desarrollo, y conseguir un producto de mayor calidad. Es necesario definir desde el comienzo del

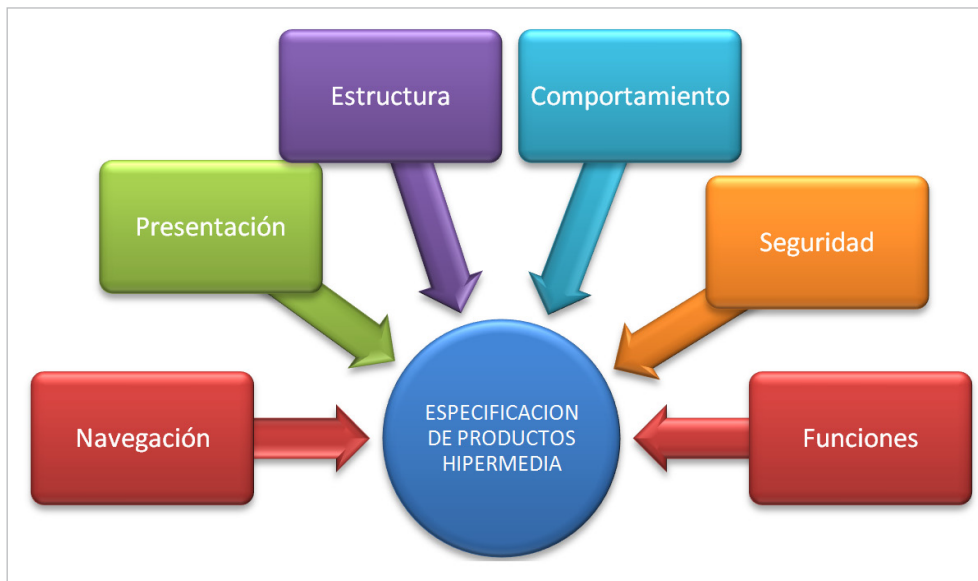


Figura 11. Perspectivas para el desarrollo de sistemas hipermmedia.

proyecto, cuándo y cómo van a participar los usuarios (expertos o usuarios finales), el entorno en el que el usuario realiza su trabajo, y la terminología y notación a utilizar.

Toda aplicación de software, nace a partir de una necesidad de usuario, por lo tanto se debe aplicar apropiadamente la ingeniería de requerimientos, pues son los usuarios iniciales quienes dan los primeros parámetros que servirán para crear el prototipo que puede ser subido a la web, y que un mayor número de usuarios empiece a dar sus aportes con la intención de mejorar la aplicación.

La participación del usuario durante todo el proceso de desarrollo de la aplicación va a depender del tipo de software que se esté produciendo; por ejemplo, si es una aplicación educativa, el registro del estudiante y su participación durante todo el módulo marcará la pauta para el ingreso de sus preferencias; si es una aplicación informativa, los usuarios serán diversos; sin embargo, las preferencias por una u otra opción dentro del software permitirá determinar las

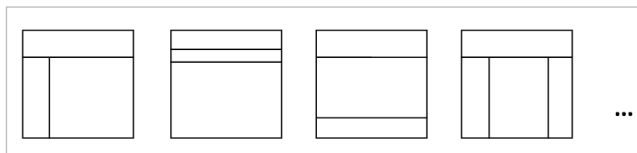


Figura 12. Opciones de diseño.

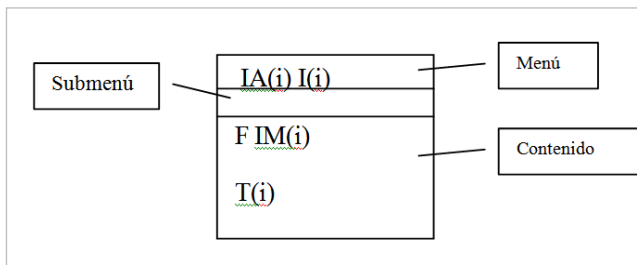


Figura 13. Ejemplo de esquematización de plantilla.

relevancias de la aplicación. Y así con los demás tipos.

Equipo de trabajo pluridisciplinar.

El sistema hipermmedia está formado por un sistema informático y por sus usuarios, para comprender las necesidades de los usuarios se pueden requerir conocimientos de psicología, sociología, antropología e incluso fisiología; para el desarrollo de la aplicación, personal fundamentalmente técnico; y para la interfaz de usuario la intervención

de especialistas de diseño, ingeniería, lingüística o arte [27].

Se considera que a pesar de que no siempre es factible involucrar a tanta variedad de profesionales, sí es recomendable que el equipo de trabajo sea mínimamente pluridisciplinar, contando al menos con:

- Técnicos capaces de realizar el análisis, el diseño y la implementación del software de aplicación.
- Especialistas en las necesi-

dades de los usuarios que puedan tomar decisiones en las opciones de diseño (por ejemplo si es un sistema de enseñanza, un pedagogo).

- Diseñadores gráficos y otros especialistas en el tratamiento de información multimedia.

Continuando con las características de la metodología propuesta, en la Figura 10 se plantean las etapas que componen su ciclo de vida.

1. Etapa de Análisis:

- Determinación de los requerimientos.
- Identificación de los objetos.
- Elaboración del mapa de navegación del sistema. Ver Figura 11.

2. Etapa de Diseño:

- Especificación de producto hipermedia
- Diseño de la plantilla o interfaz de usuario
- Estandarización de formatos
- Diseño de la base de datos (Al momento de crear las entidades se debe considerar el número de objetos obtenidos de la etapa anterior)
- Diseño del plan de prueba
- Diseño del plan de evaluación y mantenimiento

Especificación de producto hipermedia: Ver Figura 12.

Navegación.- Al utilizar hipertexto ofertamos una opción al usuario de navegar por un sinnúmero de enlaces o conexiones, por este motivo el modelo presenta una propuesta para crear una estructura hipertextual, a través de una red de nodos interconectados, recogiendo caminos asociativos y la semántica de la navegación.

Elemento	Nomenclatura
Texto	T(i) i = 1,...,n
Hipertexto	Ht(i) i = 1,...,n
Animación	A(i) i = 1,...,n
Vídeo	V(i) i = 1,...,n
Botón(es)	B(i), i=1,...,n
Iconos o Iconos Animados	I(i) o IA(i)
Imagen	IM(i) i = 1,...,n
Sonido	S(i) i = 1,...,n
Fondo	F(i) i = 1,...,n
Narración	N(i) i = 1,...,n
Morfismo	M(i) i = 1,...,n

Tabla 1. Nomenclaturas de Elementos Multimedia

DESCRIPCIÓN	
Páginas	pw_descripción.extensión
Imágenes	im_descripcion.extensión
Videos	vi_descripcion.extensión

Tabla 2. Descripción de archivos

- Estructura de enlaces: s.
- Virtuales.
- Multiorigen o multidestino(n-arios).
- Herramientas de navegación.
- Mapas visuales.
- Índices activos.
- Visitas guiadas.
- Marcas.
- Huellas.
- Vuelta atrás.

Presentación.- Debemos indicar en esta fase algunas características a nivel conceptual que fueren a utilizar algunos principios de diseño de la usabilidad de manera que se incite a los diseñadores en prestar total importancia a la naturaleza estética de este tipo de aplicaciones.

- Relaciones, restricciones temporales y espaciales (el uso de ubicaciones relativas con respecto a las absolutas posibilita una mayor independencia).
- Características de presentación (cada conteni-

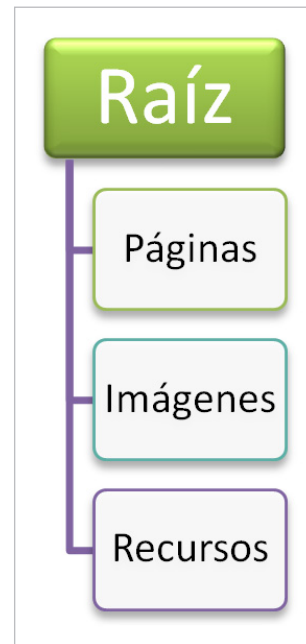


Figura 14. Estructura física del sitio.

do debe ir acompañado de especificaciones de presentación como tamaño de letra, color, tipografía...).

Estructura

- Separación estructura

Descripción	Ejemplo:
Base de datos	BD_DESCRIPCION
Tabla	tb_x_descripcion(donde x es el tipo de tabla, M:maestra, T:transaccional, H:histórica...)
Campo	xx_descripcion(donde xx son dos iniciales que hagan referencia al nombre de la tabla)
Procedimiento almacenado	pa_descripcion
Vistas	vt_descripcion
Trigger	tr_descripcion

Tabla 3. Descripción de Base de datos.

- del contenido.
 - Objetos compuestos.
 - Relaciones estructurales.
- Comportamiento
- Interacción.
 - Objetos virtuales.
 - Especificaciones procedimentales.
- Seguridad.- especificar políticas de alto nivel que regules quién puede hacer qué y en qué contexto, desde una perspectiva más conceptual que física.
- Confidencialidad.
 - Integridad.
 - Políticas de grano fino.
 - Personalización.
- Funciones
- Procesamiento.
 - Acceso a aplicaciones externas
- Diseño de la plantilla o interfaz de usuario.

Existen varias formas de establecer la distribución de información que se usarán en el sitio web, es necesario realizar un análisis de las ventajas y desventajas que cada una de ellas presenta. Lo primero que se debe resolver es la cantidad de texto, imágenes o datos que se va a presentar; en lo particular se sugiere la opción b, puesto que al tener el menú y el submenú en la parte superior se obtiene un mayor espacio para el contenido. La Figura 14 muestra un ejemplo de esquema según la opción b.

Estandarización de formatos.
Nomenclatura Elementos Multimedia.

En la etapa de diseño se requiere utilizar un sinnúmero de elementos dentro del sitio; para facilitarlos, se recomienda emplear la nomenclatura mostrada en la Tabla 2.

Descripción de las páginas: Ver Tabla 3.

Estructura de sitio: Ver Figura 15.

Es preferible tener organizado los archivos para que el mantenimiento sea más eficiente.

Diseño de la base de datos

Para iniciar el trabajo del modelo de base de datos es necesario determinar los objetos que fueron definidos en la etapa anterior, si optaron por un diagrama UML, o EDG de OMT para modelar los objetos, es criterio del programador -a partir de éstos- iniciar la creación de las tablas. Algunos tendrán que ser divididos porque al momento de relacionarse con otros pueden generar una cardinalidad de muchos, lo que obligará a la creación de un nuevo objeto y conste en una tabla del modelo.

Es conveniente estandarizar los nombres de las tablas, campos y procedimientos almacenados, puesto que, como se mencionó en párrafos anteriores, es más fácil



Figura 15. Tabla de métricas propuestas para el modelo.

brindar mantenimiento si está estandarizado. Ver Tabla 4.

El estándar de descripciones de los elementos de la Base de Datos no es una camisa de fuerza para el programador; sin embargo, ayudan mucho al momento de que el desarrollador los usa. Al revisar su código después de algún tiempo de haber sido programado, es más fácil identificar de dónde proviene algún campo. En ciertas ocasiones, realizando auditorías de modelos de Bases de Datos, se ha encontrado nombres de personas o letras únicas que no dan mayor información, que seguramente reflejaban el estado de ánimo del diseñador en ese momento, lo que dificulta la tarea de revisión.

3. Etapa de Desarrollo.

Intervienen los siguientes aspectos:

- Análisis y selección de la plataforma de trabajo.
- Establecer clase de conexión.
- Desarrollo de los procedimientos almacenados en la base de datos, es preferible iniciar con las tablas maestras, para luego seguir con las consultas y por último las transaccionales.
- Construir la arquitectura de navegación del sitio.
- Codificar los programas técnicos y la funcionalidad del sitio.
- Creación y diseño de la página principal disponible, la misma que servida de punto inicial para todos los enlaces del sitio.

4. Etapa de evaluación.

En esta etapa es valiosa la participación del usuario (experto y real), para evaluar la calidad del producto a través de métricas establecidas para medir la usabilidad acorde a las necesidades.

Las definiciones ISO 9126

[28], muestran que el objetivo es encontrar las necesidades del usuario, aunque la calidad de software desde su perspectiva es exhibir excelencia en las condiciones actuales de uso.

Por este motivo se incluye un nuevo modelo de calidad que distingue entre tres diferentes aproximaciones de calidad de producto en las normas ISO 14598 [29].

- Calidad Interna: mide las propiedades estáticas de código.
- Calidad Externa: mide las propiedades dinámicas del código
- Calidad en Uso: se mide por el grado en el que el software está diseñado en función de las necesidades del usuario.

La calidad de sistemas web básicamente se fundamenta bajo los mismos principios de calidad de software en general.

Dentro de la misma norma ISO 9126-4 (calidad de uso), se incluyen las métricas a aplicarse dentro del modelo, véase la Figura 16.

Las métricas deben ser fáciles de calcular y se aplican a través de test con usuarios, registros de utilización u observación.

5 Etapa de Implementación.

La etapa de implementación consiste en:

- Preparar el lanzamiento de la web considerando el número de usuarios que ingresarán al sitio.
- Aplicar el plan de pruebas, para asegurar la calidad del sitio.
- Hospedar el sitio para probar en línea las páginas.
- Evaluar todas las opciones del sitio conjuntamente con el usuario, para determinar si se

cumplió con todos los requisitos.

- Determinar la fecha de lanzamiento oficial del sitio y su puesta en producción.

6. Etapa de Mantenimiento.

Es la fase generalmente olvidada por algunas metodologías de desarrollo de software, sin embargo, es una de las etapas decisivas para poder mantener la aplicación en vigencia por un tiempo prolongado, además las aplicaciones web se caracterizan por su dinamismo y constante cambio, se adaptan al usuario; lo que provoca la imperiosa necesidad de darle mantenimiento permanente a su estructura y contenido.

La estructura es mantenida por el desarrollador y el contenido por el usuario final, quien gracias a la capacitación adecuada, podrá controlar el sitio web sin ningún inconveniente.

Para entrar en la etapa de mantenimiento se debe esperar que el sitio esté en producción mínimo un mes, dentro del cual el usuario se ha familiarizado con la aplicación, el volumen de datos se ha incrementado, así como el número de visitantes.

Aplicando la encuesta planificada en la etapa de diseño, se obtendrán los datos que la retroalimentación del mantenimiento necesita. Se filtran las novedades y se establece un listado de los nuevos requerimientos, empezando con el ciclo nuevamente.

Es necesario ir desarrollando punto por punto cada una de las etapas; sin embargo, será motivo de un nuevo ensayo. Para las personas que inician en el mundo del desarrollo web, las pautas sugeridas orientan el camino hacia nuevas y más eficientes metodologías que demanda el progreso constante de la

tecnología informática.

Conclusiones

- Los desarrolladores de software hipermmedia deben mejorar la productividad y calidad para satisfacer las necesidades del contexto, reducir costos y tiempos de entrega. No obstante, los métodos y técnicas estándar que se utilizan para desarrollo web, aún presentan falencias.
- Las herramientas actuales están enfocadas tan solo en la etapa de implementación mas no en la de análisis y diseño, emplea demasiado código que resulta complicado reutilizar, además de sus componentes, que son difíciles de personalizar y encajar juntos.
- De lo analizado se concluye que se puede utilizar diagramas UML u OMT en las etapas de modelado, esto va a de-

- pende del dominio de una de estas dos metodologías de diseño, para luego hacerlas efectivas en la etapa de desarrollo.
- Si se opta por el enfoque orientado a objetos, las ventajas reconocidas para esta técnica de modelado, tal como la flexibilidad y la reutilización, son quienes marcan la diferencia en los proyectos de desarrollo hipermmedia.
- Al momento de decidir la generación de aplicaciones web, se debe tener en cuenta que fueron creadas para realizar navegación a través del espacio de información, utilizando un simple modelo de datos de hipermmedia.
- La metodología sugerida coincide con la idea de invertir un mayor tiempo en las tareas iniciales que corresponden al análisis de los requerimien-

- tos y el diseño, porque a pesar de que involucra mayores costos, al final se evidencia cómo al reutilizar los componentes se reduce notablemente los tiempos de desarrollo y mantenimiento.
- Se debe poner énfasis en la participación de los usuarios, puesto que numerosos estudios [30], demuestran que el 80 por ciento de los costos de mantenimiento se deben a la interacción de usuario-sistema y no a problemas técnicos, dejando en evidencia que la aplicación apropiada de ingeniería de requerimientos es la clave para un mejor producto.
- En el presente artículo se ha dado mayor énfasis a la etapa de análisis y diseño, que a las otras, por la firme convicción de que son los pilares fundamentales para el éxito de un proyecto web.

Referencias Bibliográficas

- [1] Booch G., (1996). "Análisis y diseño orientado a objetos con aplicaciones", Editorial Addison Wesley, p. 42.
- [2] Sitio oficial de Merode, <http://merode.econ.kuleuven.ac.be/>, extraída 10 de enero de 2011.
- [3] Schwabe, D. & Rossi, G. "Building Hypermedia Applications as Navigational Views of Information Models".
- [4] Rossi, G. "An Object-Oriented Method for Designing Hypermedia Applications", Tesis de Doctor en Informática, Departamento de Informática, Pontificia Universidad de Rio de Janeiro, Brasil. En <http://www.lifia.info.unlp.edu.ar/~fer/oohdm/>.
- [5] Schwabe, D. & Rossi, G. (1995). Building hypermedia applications as navigational views of information models. In HICSS (3), pp. 231-240.
- [6] Schwabe, D., Rossi G. & Junqueira, S. (1996). Systematic hypermedia application design with OOHDM. In Hypertext, pp. 116-128.
- [7] Schwabe, D. & Rossi, G. (1995). The object-oriented hypermedia design model. Common. ACM, 38(8), pp. 45-46.
- [8] Schwabe, D. & Rossi, G. (1998). An object oriented approach to web-based applications design. TAPSO, 4(4), pp. 207-225.
- [9] Kim, W. (1995). Modern Database Systems: The Object Model, Interoperability, and Beyond. ACM Press and Addison-Wesley.
- [10] Blazewicz, Z., et al. (2003). Handbook of Data Management in Information Systems. Springer.
- [11] Rossi, G., Schmid, A., & Lyardet, F. (2003). Customizing business processes in web applications. In EC-Web, pp. 359-368.
- [12] Isakowitz, T., Stohr, E., & Balasubramanian, P. (1995). RMM: a methodology for structured hypermedia design. Common. ACM, 38(8), pp. 34-44.
- [13] De Troyer, O. & Leune, C. WSDM: a user centered design method for web sites. Computer Networks and ISDN System, 30(1-7), pp. 85-94.
- [14] Ceri, S., Fraternali, P., & Bongio, A. (2000). Web modeling language (WebML): a modeling language for designing web sites. Computer Networks (Amsterdam, Netherlands: 1999), 33(1-6), pp. 137-157.
- [15] Bongio, A., Ceri, S., Fraternali, P. & Maurino, A. (2000). Modeling data entry and operations in WebML. In WebDB (Informal Proceedings), pp. 87-92.
- [16] Brambilla, M., Ceri, S., Comai, S., Fraternali, P., & Manolescu, I. (2003). Specification and design of workflow-driven hypertexts. J. Web Eng., 1(2), pp. 163-182.
- [17] Sperberg-McQueen, M., Maler, E., Paoli, J., Yergeau, F., & Bray T. (2006). Extensible markup language (XML) 1.0 (fourth edition). W3C Recommendation <http://www.w3.org/TR/2006/REC-xml-20060816>, W3C.
- [18] Clark, J. (1999). XSL transformations (XSLT) version 1.0. W3C Recommendation <http://www.w3.org/TR/1999/REC-xslt-19991116>, W3C.
- [19] Horsa, L., Jacobs, I., & Raggett, D. (1999). HTML 4.01 specification. W3C Recommendation <http://www.w3.org/TR/1999/REC-html401-19991224>, W3C.
- [20] Yourdon, E. (1989). Modern Structured Analysis. Prentice-Hall.
- [21] Lange, D. B. (1994). An object-oriented design method for hypermedia information systems. In HICSS (3), pp. 366-375.
- [22] Cabero, J. "Evaluación de medios y materiales de enseñanza en soporte multimedia", Comunicación y Pedagogía, pp. 15-28
- [23] Koch, N. A comparative Study of Methods for Hypermedia Development, Ludwig-Maximilians-Universität Munich, Institute of Computer Science.
- [24] Metodologías de Aplicaciones Web, <http://www.areaordenadores.com/Metodologias-Web.html>, Creative Commons, extraído el 5 de octubre de 2011.
- [25] Gouldj, D., Boies, S. J. & Ukelson, J. P. (1997). "How to design usable system". En Handbook of Human Computer Interaction, pp 231-254.
- [26] Lowe, D & Hall, W. (1999). "Hypermedia and the web: an engineering approach". John Wiley & Sons.
- [27] Faulkner, C. (1998). "The essence of human-computer interaction". Prentice Hall.
- [28] ISO/IEC 9126 International Standard (1991). Information technology: software product evaluation, Quality characteristics and guidelines for their use.
- [29] ISO/IEC 14598-1 International Standard (1998). Information technology: Evaluation of software Products. Part I General Guide.
- [30] Launder, T. K. (1995). "The trouble with computers: Usefulness, Usability and Productivity". MIT Press.