

# SymWeb: Modelado de una aplicación de simulación web mediante OOHDM

Omar Sosa-Tzec <sup>1</sup>, Víctor Menéndez-Domínguez <sup>1</sup>, Luis Vargas-Villamil <sup>2</sup>

<sup>1</sup> Facultad de Matemáticas – UADY.  
Calle 8 x 21 Col. Ma. Luisa. CP 97177. Mérida, Yucatán.  
tzec@cimat.mx, mdoming@tunku.uady.mx  
www.uady.mx/sitios/matemati

<sup>2</sup> Facultad de Veterinaria y Zootecnia – UADY.  
Carretera Mérida-Xmatkuil Km. 15.5. CP 97100. Mérida, Yucatán.  
luis@avanzavet.com  
www.uady.mx/sitios/veterina

**Resumen.** Una problemática que existe actualmente es como diseñar aplicaciones web. El desarrollo de estas aplicaciones debería ajustarse a un modelo ingenieril como de cualquier software. Entre las opciones que se proponen actualmente, se encuentra el *Modelo de Diseño Orientado a Objetos de Hipermedia* (OOHDM). En este trabajo se presenta el modelado con OOHDM de una aplicación de simulación web relacionada con nutrición animal, denominada SymWeb. Se describe brevemente cada una de las fases del modelo. Luego se presenta la metodología empleada en el desarrollo; se muestra una síntesis del análisis de requerimientos y cómo se conformaron los diagramas conceptuales y de navegación, así como la implementación. Se discuten los resultados obtenidos de la aplicación del modelo en el caso de SymWeb. Se concluyó que la principal aportación del OOHDM es facilitar la comunicación en un grupo interdisciplinario enfocados al desarrollo de una aplicación web.

## 1 Introducción

En la actualidad surge el interés en desarrollar aplicaciones que corran en la Internet, en particular en la Web. Estas aplicaciones se conocen con el nombre de *aplicaciones web* o de *hipermedia* [19]; entre ellas se encuentran las que realizan simulaciones.

La historia de la simulación web comienza como un área de trabajo académico el Congreso de Simulación de Invierno (WSC por sus siglas en inglés) [11]. *simulación web* es un término que introdujo Paul A. Fishwick [3] para representar conexión entre la Web y el campo de la simulación. Otros autores añaden a la definición que el propósito de la simulación web es ejecutar más simulaciones eficientes [21].

En el caso de la simulación web, la Web puede servir como sistema operativo como el canal de distribución para sus aplicaciones, tomando las características de la Web a su favor, [8]. Sin embargo, este nuevo medio requiere explorar nuevas técnicas para el desarrollo de aplicaciones web, como por ejemplo el *Modelo de Diseño Orientado a Objetos de Hipermedia* (OOHDM).

## 2 Motivación

El Ciclo de Vida de Desarrollo del Sistema (CVDS) para una aplicación web es similar al de cualquier aplicación de software. Para desarrollarlo, se selecciona un *modelo* a seguir. El modelo se refiere al conjunto de productos, formalismos y mecanismos que se usan durante todo el CVDS. Los nombres de las etapas entre los diferentes modelos pueden cambiar, pero prácticamente son una variante del esquema básico del CVDS y que se arraigan a un tipo de conceptualización en particular (o metodología previa) dando su propia versión de este [19]. Al proceso de desarrollar una aplicación web bajo un enfoque ingenieril siguiendo un modelo establecido se le denomina como Ingeniería de Hipermedia.

En el caso de *SymWeb*, se consideró desarrollar, bajo un enfoque de Ingeniería de Hipermedia, una aplicación web para realizar la simulación de modelos dinámicos de nutrición animal mediante métodos numéricos. Estos modelos son producidos a través de un proceso de preprocesamiento de archivos generados por un software comercial de simulación.

El modelado en la producción animal permite reproducir los diferentes procesos digestivos y metabólicos que se llevan a cabo dentro del animal para así estudiarlos con una base cuantitativa y dinámica a más bajo costo que en la investigación pecuaria convencional” [7]. De esta idea surgió el proyecto de desarrollar un simulador de modelos matemáticos veterinarios abierto al público interesado dentro del proyecto 33722-B de CONACyT intitulado “Desarrollo de un Modelo Dinámico para la Simulación de la Producción Bovina de Carne en el Trópico”. El proyecto recibió el nombre de *SymWeb*, por Simulación y Modelado vía Web.

La problemática se enfocó en la manera de cómo se llevaría a cabo el desarrollo. Después de un análisis comparativo de la literatura, derivó en el uso del OOHDM desarrollado por Daniel Schwabe y Gustavo Rossi [18].

## 3 Descripción del OOHDM

A grandes rasgos OOHDM es un modelo de construcción de grandes aplicaciones hipermedia; enfocado en dos aspectos críticos en el desarrollo de aplicaciones web, las cuales son la interfaz del sistema y el esquema de navegación, para lo cual se utiliza una colección de objetos.

Este modelo está constituido de las siguientes fases: Análisis de requerimientos, modelo conceptual, diseño navegacional, diseño abstracto de interfaz e implementa-

ción. Los procesos de desarrollo se modelan usando notación del Lenguaje Modelado Unificado (UML) [20].

### 3.1 Análisis de requerimientos

En esta primera fase se identifican los actores y las tareas que realizan, definiendo así un escenario. Los escenarios son agrupados para formar un caso de uso el cual representado utilizando diagramas de interacción de usuarios (UID)[15]. Estos diagramas proporcionan una representación gráfica de la interacción del usuario con sistema durante la ejecución de una tarea [18].

Los escenarios son descripciones narrativas de cómo la aplicación puede ser usada. A las descripciones narrativas o textuales de las acciones de los actores según sus roles se les conoce como Especificaciones de Escenario (EE) [15]. Siempre recomendable que los usuarios proporcionen tales descripciones. Si lo anterior no posible, los diseñadores deberán encargarse de redactarlas. Para especificaciones los escenarios muy generalizadas, se aclara el “cómo” elaborando Casos de Uso (CU). En sí, éstos son agrupaciones de escenarios que describen la misma tarea [15].

Como las representaciones gráficas pueden ser más fáciles de seguir, entonces hace uso de los UID. Éstos explican de manera gráfica los casos de uso. Un UID está conformado por elipses y flechas que representan la información a intercambiar y el procesamiento de la información, respectivamente.

### 3.2 Diseño conceptual

En esta fase se concibe la aplicación en términos de objetos y las relaciones existentes entre ellos [18], utilizando la Tecnología de Modelado de Objetos (OMT) [12]. Esto es, definir diagramas de clases para representar abstractamente la aplicación.

El diseño conceptual tiene como producto el esquema conceptual, el cual explica la manera en la que está conformada la aplicación: Las entidades que contienen información y como se relacionan. En este punto no se requiere qué información verá el usuario o cómo va ha acceder a la información, mucho menos la forma en que se implementará la aplicación.

### 3.3 Diseño navegacional

En este punto, se describe la estructura navegacional de una aplicación hipertexto en términos de contextos navegacionales que son generados de clases navegacionales [13]. El diseño navegacional es expresado en dos esquemas: Esquema de Clases Navegacionales y Esquema de Contexto Navegacional. En el primero se realiza especificación de nodos, los cuales son proyecciones de una clase y representan elementos que tienen la información. El esquema de contexto navegacional presenta las diferentes maneras en la que un nodo puede ser accedido según el esquema de clases navegacionales. Estas maneras son los contextos [13].

Un contexto navegacional es un conjunto de nodos, enlaces y las denominadas clases contextuales. Puede ser definido en intención cuando presentan una propiedad particular presente en todos los nodos o enlaces; o bien, ser definidos en extensión, que es cuando se enumeran sus miembros [13].

Un diseño navegacional puede utilizarse para más de una aplicación. Los contextos tienen como función ayudar al usuario a realizar su tarea durante la navegación de la aplicación. Cuando se establece una plantilla de diseño navegacional que puede adaptarse a una aplicación según el contexto, se le llama esquema de contexto navegacional genérico, el cual está compuesto por representaciones genéricas de los contextos mencionados anteriormente. Existen diferentes notaciones gráficas para los contextos genéricos [13][14][16], pero básicamente son cinco: Contexto genérico, creación de instancia/modificación, estructura de acceso genérica o índice genérico, estructura de acceso jerárquica genérica y búsqueda genérica. Cada una de las notaciones para los contextos genéricos pueden estar acompañados de una tarjeta de especificación donde se definen las características como la relación definida por el enlace que los une (e.g. uno a muchos), tipos posibles de información que contienen (estáticos, dinámicos, jerárquicos), restricciones, etc.

### 3.4 Diseño abstracto de interfaz

El diseño de interfaz de usuario se auxilia del Diseño Abstracto de Interfaz (AID). Un AID se enfoca en saber cuales de los diferentes objetos navegacionales aparecerán, cuales activarán la navegación y cuales transformaciones de la interfaz se presentarán [15].

La conformación de un AID está dada por elementos llamados Vistas Abstractas de Datos (ADV). Estas ADV son propiamente la interfaz representada por los diferentes elementos con los que interactuará el usuario. Las ADV contienen los elementos claves que visualizará el usuario al acceder el nodo. Pueden ser cadenas de texto particulares, imágenes o información generada por el contexto. En el ADV se define primeramente los eventos relacionados con el acceso. Los demás elementos en la ADV que no están relacionados directamente con el nodo (i.e. elementos no proyectados en el nodo), son de auxilio en la navegación o poseen simplemente características decorativas. Se definen a éstos como decoradores de nodo.

### 3.5 Implementación

Por su naturaleza, OOHDM no expresa explícitamente como ha de ser la implementación de la aplicación. Sin embargo, presenta la noción de los objetos de navegación como vistas de las clases conceptuales, organiza el espacio de navegación con la introducción de contextos de navegación, separa los aspectos relacionados con la interfaz de usuario de aquellos navegacionales y explícitamente señala que existen decisiones de diseño que necesitan ser hechas solo al momento de la implementación [15]. Éstas son las características que pueden ayudar a establecer una arquitectura adecuada para implementación.

## 4 Resultados

Como *SymWeb* está enfocado a la simulación de modelos matemáticos veterinarios por parte de personas interesadas en el tema, se definieron los roles dependiendo las características particulares que puede poseer un usuario. El listado de roles (omitiendo tareas por cuestión de espacio) se presenta en el Cuadro 1.

Rol	Descripción
Usuario No Registrado	Usuario común; es decir, aquel que entra a navegar la aplicación con meros propósitos informativos, por curiosidad o por accidente.
Usuario Registrado	Usuario que ha pasado por el proceso de registro en <i>SymWeb</i>
Usuario Publicador	Usuario registrado que tiene la capacidad de publicar modelos matemáticos para otros usuarios registrados
Administrador	Usuario registrado único que tiene la capacidad de publicar modelos matemáticos en <i>SymWeb</i> pero también tiene la capacidad de administrar a los usuarios registrados

Cuadro 1. Nominación y descripción de roles de *SymWeb*.

El rol básico es el *Usuario No Registrado*. Este tipo de actor puede cambiar rol dependiendo del evento denominado *Registro de Usuario*, en donde este actor pasa ahora a ser parte de otro grupo (*Usuario Registrado*). Por otra parte, algunos estos usuarios registrados tienen la capacidad de poder ejecutar otro rol de manera simultánea: *Usuario Publicador*. Existe el actor de la aplicación que le indica a *Usuario Registrado* cuando puede desempeñar el rol alterno, éste es el *Administrador*, quien al fin y al cabo es un “actor multifacético”, ya que puede desempeñar todos los roles dentro de la aplicación.

Según el análisis realizado en los requerimientos, se obtuvieron 10 especificaciones de escenarios que se consideraron básicas para el modelado de la aplicación, cuales van desde *navegar SymWeb* hasta *salirse de SymWeb*, entre otros.

En consecuencia, se realizaron 12 CU, los cuales se enlistan a continuación: Registro de Usuario, Inicio de Sesión, Actualización de Datos del Usuario, Simulación de Modelos Matemáticos, Graficación de Resultados, Publicación de Modelos Matemáticos, Búsqueda Interna General, Búsqueda Interna Avanzada, Dar de baja a Usuario, Dar de baja a un Modelo, Darse de baja como Usuario y Salirse de *SymWeb*.

Para el desarrollo de *SymWeb* se enriqueció el significado de las elipses que componen los UID para señalar el lugar (físico o abstracto) que se ve afectado por procesamiento de información. También se adaptaron éstos con etiquetas añadidas las elipses para facilitar su lectura. En la Figura 1 se presenta el UID para el Registro de Usuario. Los UID en *SymWeb* se realizaron en el orden determinado por CU.

Para abstraer las clases involucradas con *SymWeb* hay que tener en cuenta que ta es una aplicación que consiste en simular modelos matemáticos que generen resultados gráficos por usuarios que probablemente tendrán la capacidad de publicar tales modelos bajo el control de un administrador, quién por defecto publica modelos. El enunciado anterior encierra los puntos señalados en los escenarios reali-

zados previamente. De esta forma se obtuvieron las clases preliminares: Modelos Matemáticos, Gráficas, Usuarios y Administrador.

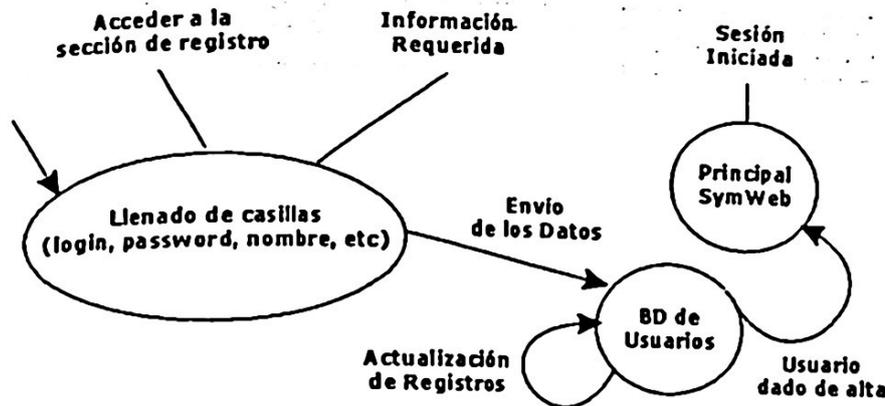


Fig. 1. Diagrama de Interacción de Usuario para Registro de Usuario

Como resultado de un diseño detallado, se abstraieron 17 clases (presentadas sintéticamente en la Figura 2) que definen 11 enunciados que describen *SymWeb* de manera conceptual.

La definición de los nodos en *SymWeb* se basó en realizar preguntas concretas sobre los eventos y acciones asociados a la clase conceptual. Como resultado se obtuvo 8 tipos de nodos diferentes. Como puede observarse en la Figura 3 correspondiente al nodo *Administrador*, el nodo *admin* es una instancia de la clase *Admin* que se reconoce dentro de la aplicación con la variable *admin*. Este nodo hereda los atributos (e.g. login) y métodos (e.g. *Darse\_de\_baja()*) de la clase *Usuario*. Diseñando de manera similar los demás nodos, se establecen 4 esquemas de clases navegacionales para *SymWeb*, donde cada una está relacionada con el usuario actual de la aplicación.

En el caso de *SymWeb* se concluyó el diseño navegacional con tan solo un esquema de contexto navegacional (Figura 4). Esto se debe a que modelamos la aplicación *SymWeb* de manera particular, por lo que no llegamos a un esquema conceptual adaptable a todas las aplicaciones de hipertexto que hagan simulación. Para poder realizar este esquema fue necesario instanciar los contextos genéricos en contextos particulares dentro de los nodos.

En el esquema de contexto navegacional se pueden observar ciertos detalles. Primeramente, se tiene un índice general denominado "Opciones de *SymWeb*" que procede según el tipo de usuario que navegue (i.e. registrado o no registrado). Los usuarios no registrados solo acceden a los nodos generales del modelo y de gráfica. Si un usuario es registrado, entonces accede al nodo de "login" donde dispara el método *inicio\_de\_sesion()* para después acceder al "Menú General" que está representado por un índice (i.e. una lista de enlaces a las opciones). Adjunto a este índice están el de "Menú Publicación" y "Menú Administración", los cuales están etiquetados con un ovalo negro a su izquierda; este ovalo indica que son índices de acceso restringido y que en nuestro caso solo están disponibles para un usuario con permiso de publicar y para un usuario que tenga además permiso de administrador, respecti-

vamente. Se observa también que todos los usuarios pueden acceder a un índice de "Búsqueda", el cual devuelve información tanto de modelos encontrados, así como de archivos relacionados a éstos, almacenados en el servidor.

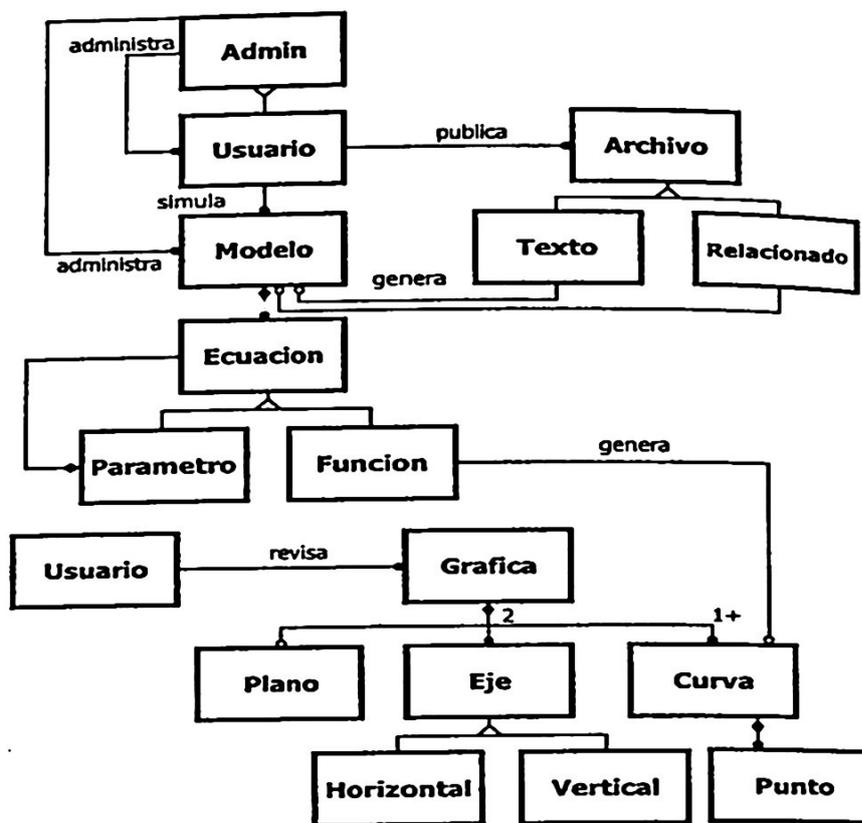


Fig. 2. Esquema conceptual de SymWeb (representación sintética)

<b>NODE</b> admin <b>FROM</b> Admin: admin <b>INHERITS FROM</b> Usuario
EsAdmin: booleano
ToUsuario: Anchor(administra)
ToModelo: Anchor(administra)
Admin_usr()
Admin_mod()

Fig. 3. Nodo Administrador

Durante el proceso de AID, se definieron varios ADV como el mostrado en la Figura 5, los cuales indicaban que componentes de información serían utilizados secciones particulares de la navegación.

Con respecto a la implementación, se definió que *SymWeb* fuera una aplicación basada en páginas web dinámicas que interactúan entre ellas. Estas páginas fueron desarrolladas con la tecnología de Páginas de Servidor Activas (ASP) de Microsoft, incluyéndose también javascript para analizar rutinas de validación y otras funciones en el lado del cliente. La decisión de utilizar ASP se basó en la tecnología disponible al momento de iniciar el proyecto. Sin embargo, se tiene contemplado reali-



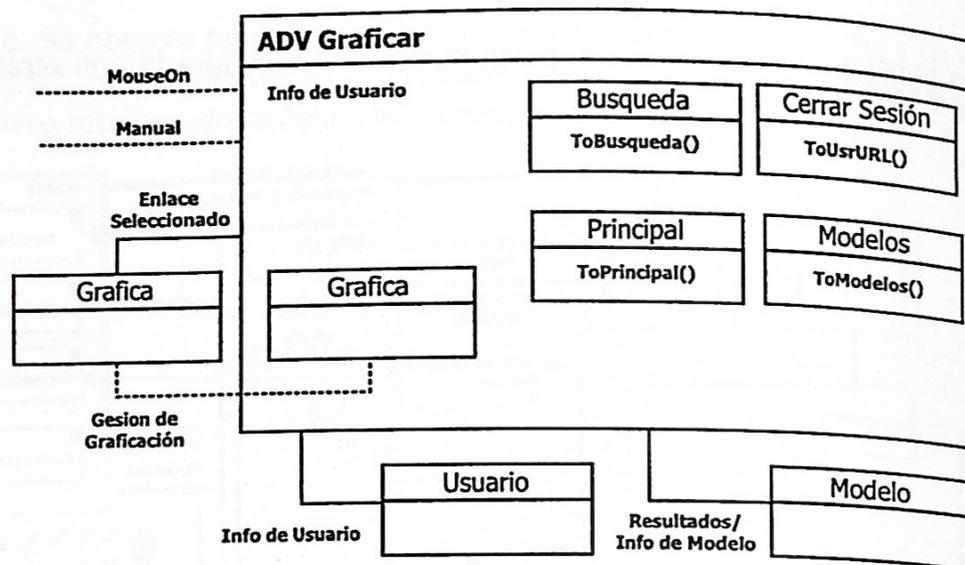


Fig. 5. Vista (sintetizada) de Datos Abstracto correspondiente a la graficación de resultados de un usuario registrado

El acceso por parte de los usuarios se realiza a través de sesiones. La entrada de toda la información se hace a través de elementos de formulario, la cual es procesada siempre del lado del servidor. El proceso de publicación de un modelo se realiza a través del preprocesamiento de un archivo de texto que contiene ecuaciones diferenciales generadas a partir de un software de simulación para PC [5]. Este proceso se auxilia con un componente para cargar archivos en el servidor [1]. Cuando realiza la simulación de un modelo, la tabulación de resultados se establece en un archivo XML que es pasado como parámetro a un componente de graficación [6]. Se cuida que ninguna gráfica se presentada hasta que algún modelo se simule. El manejo de los datos del lado del servidor se realiza utilizando tres tablas: Una para los usuarios, otra para los modelos y una tercera para archivos que contengan información sobre los modelos.

El sitio está compuesto por una raíz denominada *SymWeb* y que será el elemento final de la cadena correspondiente al URL. En esta carpeta se encuentran todas las páginas a las que accede el usuario. En adición, se encuentran otras carpetas de propósito específico (e.g. carpeta de modelos).

Se conceptualiza la GUI (figura 6) tomando en cuenta los cinco atributos de definición de la usabilidad según Nielsen [9][10]. Se utilizó un símil de carpetas. Cuando fue necesario, se presentó al usuario retroalimentación o mensajes de error procurando ser siempre claros. Se consideró una GUI en tonalidades verdosas debido al perfil del usuario potencial y se empleó tipografía fácil de leer en monitor con el fin de incrementar la satisfacción subjetiva que marca la usabilidad web [9]. Incluso se empleó en el diseño de interfaz los principios de la psicología Gestalt [2].

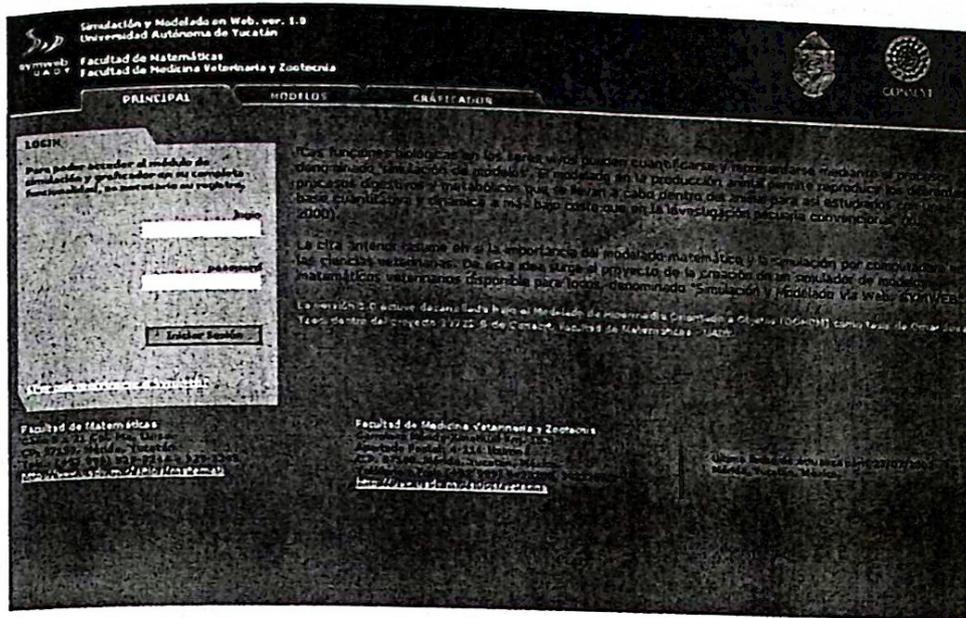


Fig. 6. Portal de acceso – Interfaz Gráfica de Usuario de *SymWeb*

## 5 Discusión

Para cualquier aplicación web, seleccionar una metodología de desarrollo con base en un modelo es una importante decisión. Hay que conocer las características del modelo a usar suponiendo, no garantizando, que modelará adecuadamente la aplicación. El OOHDM al ser un derivado de UML y OMT, tiene ventaja sobre otros modelos debido a que está en un contexto de desarrollo de software actual. Así, se modeló una aplicación simulación web para modelos veterinarios que es independiente de implementación; es decir, se podrá utilizar otra plataforma en un futuro utilizando este modelado. La ventaja de seguir este enfoque es que cualquier cambio a efectuarse en la aplicación derivará del modelado actual. En tal caso, se espera poder hacer un mapeo modelo a modelo de la aplicación.

Como desarrollador de software dentro de un grupo de simulación web, se tiene la ventaja de intervenir en el diseño para realizar modificaciones no consideradas por los investigadores de otras disciplinas dentro del proyecto. Para poder lograr una cooperación adecuada entre la gente de desarrollo y los demás integrantes del proyecto, se acudió al Diseño Centrado en el Usuario (UCD), el cual fue desarrollado a través de constantes sesiones. El UCD casi no es considerado cuando se desarrollan aplicaciones; muchas veces el enfoque está tan solo en la técnica. En el caso de aplicaciones web como en *SymWeb*, el UCD es vital en el desarrollo, ya que el simple hecho de que el programa correrá sobre la Internet, implica que el éxito radicará en el máximo cumplimiento de los requerimientos de la comunidad global de usuarios. De la combinación de seguir el proceso ingenieril con el UCD, se observó que beneficia al proceso de desarrollo, ya que la gente encargada de esto debe estar en constante contacto con los demás integrantes y adentrarse lo suficiente en el contexto del campo de la ciencia que está relacionada con el proyecto.

En sí, lo más recomendable al enfrentar el desarrollo de una aplicación para simulación web es seguir un proceso de ingeniería de hipermedia. Además, se recomienda realizar prototipos una vez comenzada la ingeniería. En particular, aplicando el OOHDM en mezcla con el UCD, se observó que la participación del usuario brinda muy buenos resultados durante las etapas de diseño conceptual. Con respecto al diseño navegacional, éste debe ser realizado por el grupo de desarrollo y luego efectuar pruebas con el usuario sobre la navegación ya sea con prototipos o interfaces en papel. Esto implica que hay que tener presente también un buen desarrollo de la interfaz desde el principio, sobre todo si los usuarios finales no están tan familiarizados con la Web, ya que además de aumentar el éxito de la aplicación, hace presentes nuevas especificaciones o se detectan limitantes relacionadas con el desarrollo, lo cual deriva en una redefinición de requerimientos o consideraciones para futuras versiones de la misma aplicación.

En la actualidad SymWeb pretende ser una herramienta que fomente el uso de modelos matemáticos entre la comunidad académica veterinaria, permitiendo optimizar los recursos de la investigación tradicional, debido a que permite la utilización inmediata de los modelos publicados en SynWeb: sin requerir el uso de una aplicación comercial para realizar la simulación.

## 6 Conclusiones

El desarrollo de aplicaciones de hipermedia puede ser considerado bajo un enfoque ingenieril, tal y como sucede en el desarrollo de software a la medida; así, se observó que la aplicación del modelo ayudó a establecer puntos de control y realizar retroalimentación en el desarrollo. Con base en los resultados obtenidos, se puede determinar que la simulación web mediante páginas dinámicas permite una adecuada implementación dadas las condiciones de la Internet actual; se observó que este enfoque presenta grandes ventajas: 1) Administrar a los usuarios mediante sesiones, 2) Permite la personalización de la interfaz, 3) Se ajusta a los criterios de usabilidad web, 4) Es posible integrar componentes, lo cual fomenta el reuso de software.

El UCD facilitó el desarrollo de la aplicación al establecer desde el principio un buen canal de comunicación con el usuario final y mantuvo en foco un diseño *hoc* a la naturaleza del proyecto (*i.e.* correr en la Web).

En la actualidad, a pesar de que la simulación web está en sus inicios, se observa que estar en un grupo de simulación web, requiere que las personas encargadas del desarrollo de la aplicación posean la habilidad de sumergirse en el contexto del proyecto, ya que este tipo de proyectos requieren de un enfoque multidisciplinario. También se afirma, con base en lo observado, que el grupo de desarrollo debe centrarse en el usuario y saber identificar sus necesidades.

## Referencias

1. aspSmart. aspSmartUpload. [www.aspsmart.com/aspSmartUpload/](http://www.aspsmart.com/aspSmartUpload/) (2002)
2. Dondis, D. A.: La sintaxis de la imagen. Introducción al alfabeto visual. GG Diseño. México. (1973)
3. Fishwick, P. A.: Web-Based Simulation: Some Personal Observations. Proceedings of the Winter Simulation Conference. (1996) 772-779
4. Fraternali, P.: Tools and Approaches for Developing Data-Intensive Web Applications: A Survey. ACM Computing Survey, Vol. 31, No. 3 (1999)
5. HPS, Inc. Stella IV. URL: [www.hps-inc.com](http://www.hps-inc.com) (2002)
6. Infsoft. Fxgraph 4.0. URL: [www.infsoftglobal.com/fxgraph](http://www.infsoftglobal.com/fxgraph) (2002)
7. Ku-Vera, J.: Desarrollo de un Modelo Dinámico para la Simulación de la Producción Bovina de Carne en el Trópico. Proyecto 33722-B. Conacyt. (2000)
8. Kuljis, J.: A Review of Web Based Simulation : Whither we wander? Proceedings of the Winter Simulation Conference. (2000) 1872-1881.
9. Nielsen, J.: Usability Engineering. Morgan Kaufmann. USA. (1993)
10. Nielsen, J.: Usabilidad. Diseño de sitios Web. Prentice Hall. España. (2000).
11. Page, E.: The Rise of Web-Based Simulation: Implications for the High Level Architecture. Proceedings of the 1998 Winter Simulation Conference. (1998) 1663-1668.
12. Rumbaugh, M.: Object Oriented Modelling and Design. Prentice Hall Inc. (1991)
13. Schwabe D.: An Object Oriented Approach to Web-Based Application Design. Theory and Practice of Objects Systems 4(4). Willey and Sons, New York. (1998) 1074-3224.
14. Schwabe, D.: 'Just Add Water' Applications: Hypermedia Application Frameworks. Extended abstract for the Hypertext '99 Workshop on "Hypermedia Development: Design Patterns in Hypermedia". In Press. (1999)
15. Schwabe, D.: Modeling interactions and Navigation in Web Applications. Proceeding of the World Wild Web and Conceptual Modeling '00 Workshop, ER '00 Conference. Springer. USA (2000)
16. Schwabe, D.: Web Design Frameworks: An Approach to Improve Reuse in Web Applications. Proceedings of the Second International Workshop on Web Engineering, WWW9 Conference. Springer. USA (2000)
17. Schwabe, D.: Engineering Web Applications for Reuse. IEEE Multimedia, Spring. (2001) 2-12.
18. Schwabe, D.: The Object-Oriented Hypermedia Design Model. [www.telemidia.puc-rio.br/oohdm/oohdm.html](http://www.telemidia.puc-rio.br/oohdm/oohdm.html) (2002)
19. Sosa-Tzec, O.: Desarrollo de un Programa Web para la Simulación de Modelos Dinámicos Ligados a Base de Datos. Tesis. Universidad Autónoma de Yucatán. México. (2002)
20. Unified Modeling Language. UML Home Page. [www.uml.org](http://www.uml.org)
21. Young, H.: Web-Based Simulation with the Remote Method Invocation. Hanyang University, South Korea. In Press. (1998)