

# **Sistema Web para el proceso Enseñanza-Aprendizaje**

M.C. ORDOÑEZ PACHECO LUCAS DANIEL  
Instituto Tecnológico de Cd. Madero  
Departamento de Sistemas y Computación

## **RESUMEN**

Actualmente la educación centrada en el estudiante tiene como objetivo que la secuenciación y autoría de contenido, los modelos pedagógicos y los procesos de evaluación realmente cumplan las metas de aprendizaje para los estudiantes. Para ello los repositorios de contenido y evaluaciones deben ser apropiados para los requerimientos particulares de los individuos, pero al mismo tiempo deben ser flexibles para ajustarse a una amplia comunidad de desarrolladores y estudiantes. Es necesario que el desarrollo de un sistema de educación basada en Web considere toda la diversidad de requerimientos tecnológicos y pedagógicos, proporcionando la funcionalidad necesaria a través de los recursos de Internet.

Por tal razón este trabajo muestra una nueva arquitectura para el desarrollo de Sistemas de Educación Basada en la Web. Esta arquitectura se basa en la especificación IEEE 1484 LTSA (Learning Technology System Architecture) (IEEE, 2001) y reúne a los modelos de desarrollo de software y diseño instruccional y podría presentar las bases para los nuevos desarrollos de este tipo de sistema.

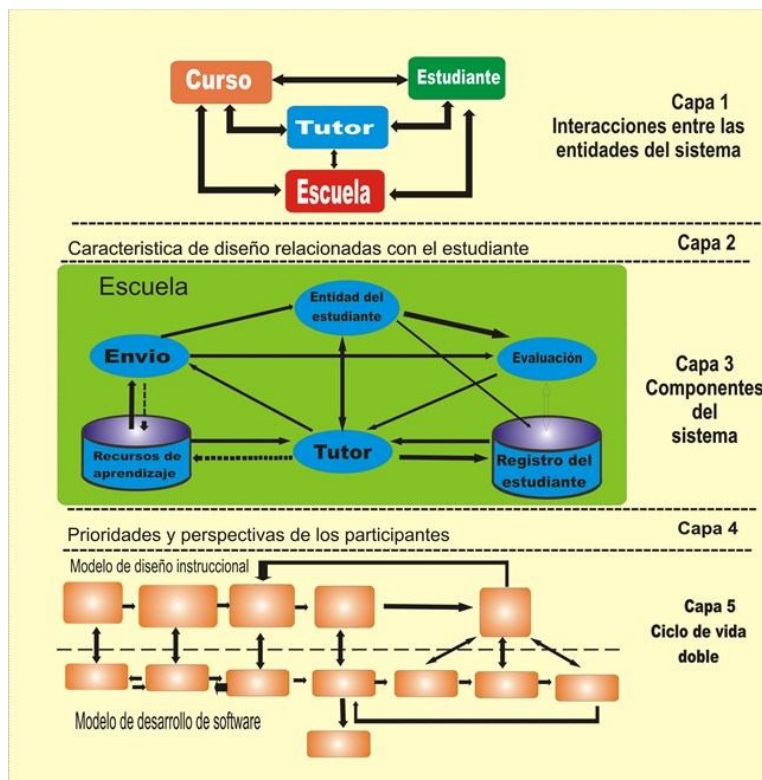
### **1. Introducción**

El diseño de la arquitectura de sistemas tecnológicos para la educación basada en la Web (EBW) es afectado directamente por las necesidades de los estudiantes y en particular, la naturaleza del aprendizaje humano. En este sentido se establece un modelo de diseño instruccional para transmitir, construir y generar conocimientos adecuados para esta herramienta tecnológica. Dicho modelo se basa en la teoría cognitiva de mapas conceptuales (MC), porque estos nos permiten generar un modelo esquemático sobre la forma en que los seres humanos estructuran una información y tienen la finalidad de ayudar a los estudiantes en la adquisición de conocimiento.

### **2. Componentes del sistema**

En la figura 1 se muestra la capa de componentes de esta arquitectura, en donde se identifican 5 procesos: Entidad del estudiante, evaluación, Tutor, proceso de envío y Escuela. Además dos bases de datos, una para recursos de aprendizaje y otra para registro del estudiante; así como catorce flujos de información entre estos componentes.

Primeramente se considera el proceso de la Escuela porque esta interviene de forma directa en todo el sistema educativo de acuerdo con las *Interacciones entre las entidades del sistema*, como se aprecia en la capa uno. Por otra parte el proceso del tutor se divide en dos: Tutor y Tutor virtual, porque este proceso tiene que adaptarse a las necesidades individuales de los estudiantes (capa dos). Por ejemplo durante el proceso enseñanza aprendizaje algunas decisiones tales como: Secuencia de temas, pruebas, actividades académicas, etc. se podrán elegir manualmente por el Tutor. En este sentido participa un tutor virtual que puede realizar modificaciones en el proceso pero de manera automática cada vez que el estudiante termine una lección o prueba y que guarde todos los movimientos realizados en el historial del estudiante.



**Figura 1. Arquitectura Tecnológica para la EBW**

El funcionamiento global de los componentes del sistema tiene la siguiente forma:

- 1) Los estilos de aprendizaje, estrategias, métodos, etc. Son valorados y acordados entre el estudiante y otro participante, y se comunican como preferencias del aprendizaje.
- 2) Se recolecta la información de las valoraciones del estudiante con base en su comportamiento y productividad, por ejemplo: Tiempo empleado en un tema, intentos en la resolución de un ejercicio, uso de los diferentes contenidos de aprendizaje, etc. Esta información se guarda directamente en los registros del estudiante, mientras el estudiante interactúa con el sistema. Estas son tareas nuevas que se incorporan al modelo original de IEEE1484 LTSA.
- 3) El estudiante es observado y evaluado en el contexto de interacciones multimedia.
- 4) La evaluación produce valoraciones formales del estudiante e información del estudiante.
- 5) La información del estudiante, es decir sus respuestas (escritas, de voz, de selección, etc.) se guarda en la base de datos del estudiante.
- 6) El tutor revisa la evaluación e información del estudiante, así como las preferencias, el historial, y posiblemente los objetivos de aprendizajes futuros.
- 7) El Tutor virtual revisa el comportamiento e información del estudiante y de forma automática e inteligente hace modificaciones dinámicas en la secuencia del curso de una materia de forma personalizada de acuerdo a las necesidades del estudiante, basándose en el diseño del proceso de aprendizaje.
- 8) El tutor virtual busca en los recursos de aprendizaje, mediante consulta e información del catálogo, un contenido de aprendizaje apropiado para el estudiante.
- 9) El tutor Virtual, extrae la ubicación de la información de los catálogos disponibles y pasa las ubicaciones (URLs) al proceso de envío o entrega, por ejemplo, un plan para abordar una lección o ligas a contenidos.
- 10) El proceso de envío extrae el contenido de aprendizaje desde la base de datos de los recursos de aprendizaje, basándose en las ubicaciones (URLs) y transforma el contenido de aprendizaje en una presentación multimedia interactiva y adaptable para el estudiante.
- 11) La escuela soporta todo el sistema educativo. Esta nueva tarea proporciona una manera de intercambiar información extra entre la Escuela, estudiantes y tutores, tal como: Respuestas a dudas y comentarios para expresar en general sus opiniones acerca de los cursos, tutores,

estudiantes y el propio sistema educativo. Es importante la orientación a componentes que se les ha dado a esta capa, ya que la finalidad es implementar una familia de aplicaciones de nueva generación, premisa de la Ingeniería de dominio.

El modelo de desarrollo de software se encuentra sustentado en un Sistema Multi-Agentes, emplea métodos y técnicas de la Ingeniería de dominio para el desarrollo de componentes de aprendizaje reutilizables e inteligentes orientados a objetos (CARIOO). El objetivo es crear una biblioteca multimedia de CARIOO para los sistemas de EBW, con el propósito de separar el contenido del control. En consecuencia los componentes utilizan diferentes niveles de código en Flash Player. Con esta estructura es posible generar componentes especializados que son pequeños, reusables y apropiados para integrarlos a un componente más grande en tiempo de ejecución. Los CARIOO se meta- etiquetan con el propósito de completarlos como una función similar a los código de barras de los productos. ActionScript adiciona el componente WebService-Connector para conectar a los servicios Web con los CARIOO. Este componente proporciona el acceso a los métodos remotos ofrecidos por un LMS (Learning Management System) a través de SOAP (Simple Object Access Protocol); esto da a un servicio Web la habilidad para aceptar parámetros y regresar un resultado al Script. El componente invoca el servicio Web utilizando SOAP y UDDI (Universal description, Discovery and Integration) a través del middleware y el servidor JUDDI (Java UDDI). Los servicios Web pueden descargarse y desplegarse dentro de un CARIOO.

En la capa "*Prioridades y perspectivas de los participantes*" (capa tres) se describe el sistema de componentes desde una variedad de perspectivas de acuerdo a los participantes (Escuela, Tutor y Estudiante). Si se analiza el diseño de la capa tres, con certeza se realizará de diferente forma de acuerdo a la percepción de los participantes, por ejemplo: Puede enfocarse desde el punto de vista del contenido de aprendizaje, o de la tecnología a emplear, etc.

En este caso, las prioridades y perspectivas del diseño de la capa tres se plantearon inicialmente tomando en cuenta a todos los elementos que intervienen en el entorno del estudiante y los elementos pedagógicos. Estos últimos se reflejan en el hecho de agregar un nuevo flujo de información a la capa tres, que se encarga de enviar directamente información del estudiante a la base de datos Registros del estudiante, su comportamiento dentro del escenario del aprendizaje. Esta información se analiza y proporciona elementos para decidir sobre la dirección (secuenciación dinámica), que debe tomar la instrucción del estudiante, pero desde una perspectiva pedagógica, la cual no sólo contempla a las evaluaciones escritas o de otra índole, como método para valorar el conocimiento adquirido de los estudiantes.

La capa "*Prioridades y perspectivas de los participantes*" (capa cuatro) proporciona un análisis sobre el diseño de la capa tres, desde la perspectiva de un sistema adaptable e inteligente y en donde intervienen principalmente los procesos de Entidad del estudiante y Tutor, dicho análisis es importante porque servirá para la implementación de la capa tres. Es importante precisar que el diseño de la capa tres se mantiene, lo único que cambia es el énfasis hacia ciertos procesos y flujos de información que tienen un papel importante de acuerdo a nuestra perspectiva y que representan una innovación dentro de esta arquitectura que se presenta.

## **Ciclo de vida doble**

La nueva arquitectura propuesta considera los aspectos pedagógicos, es decir, la inclusión de un modelo de diseño instruccional que permita traducir principios pedagógicos de aprendizaje, para una estrategia en el desarrollo de contenidos de aprendizaje y actividades. La incorporación de un modelo de diseño instruccional dentro del proceso de desarrollo de software para la EBW genera la necesidad de un ciclo de vida con esta característica, pero al revisar los estándares internacionales de ingeniería de software que rigen la industria en este campo como son los ISO/IEC 9001(ISO, 2000), 90003 (ISO, 2000), 12207 (ISO, 2002) y 15504 (ISO, 2004), es

evidente que básicamente sólo contemplan el desarrollo de software desde una perspectiva tecnológica y no profundizan dentro de los requerimientos del campo “aplicación” de software, como en este caso la educación y la metodología para llevarla a cabo. En todos los estándares de ciclos de vida citados, sólo en la sección de requerimientos es posible incluir los requerimientos pedagógicos.

La capa “*Ciclo de vida doble*” (capa cinco) se muestra en la figura 2. Este ciclo une el modelo de desarrollo de software y el modelo de diseño instruccional, lo que es una innovación en el proceso de desarrollo de software actual, ya que hasta el momento esta característica no es considerada en los estándares anteriormente citados. El propósito de este ciclo de vida no es solo el desarrollo de software educativo basado en cuestiones tecnológicas sino además, agregar elementos cognitivos que colaboren en la adquisición de conocimiento de los estudiantes. La unión de estos modelos representa un avance dentro del desarrollo de software educativo, debido a que el ciclo de vida doble integra los principios pedagógicos de aprendizaje dentro del desarrollo de software, debido a que éstos intervienen de manera directa para que se lleve a cabo el proceso enseñanza aprendizaje. Es importante precisar que el modelo de diseño instruccional se basa en la técnica de los mapas conceptuales, mientras que el modelo de desarrollo de software se rige por la ingeniería de dominio.

El ciclo de vida doble proporciona al desarrollador la facilidad para ir y venir de un extremo pedagógico a otro tecnológico para diseñar una aplicación más productiva. Esto se debe que la concepción del software tiene una finalidad educativa, porque los principios pedagógicos guiarán el desarrollo tecnológico del software. Además este ciclo de vida doble ayuda al rápido desarrollo y entrega de partes del software.

En la primera etapa del ciclo de vida doble se definen los objetivos de software y se hace un análisis del dominio, lo que equivale a obtener los requerimientos de software tanto pedagógicos como tecnológicos. Para la segunda etapa se obtiene un modelo de dominio, que es el proceso para desarrollar un modelo estructural suficientemente genérico que represente el dominio. Además se elige un modelo pedagógico para el desarrollo de cursos que en este caso son los MC, aunque cabe aclarar que se podría utilizar otro modelo pedagógico.

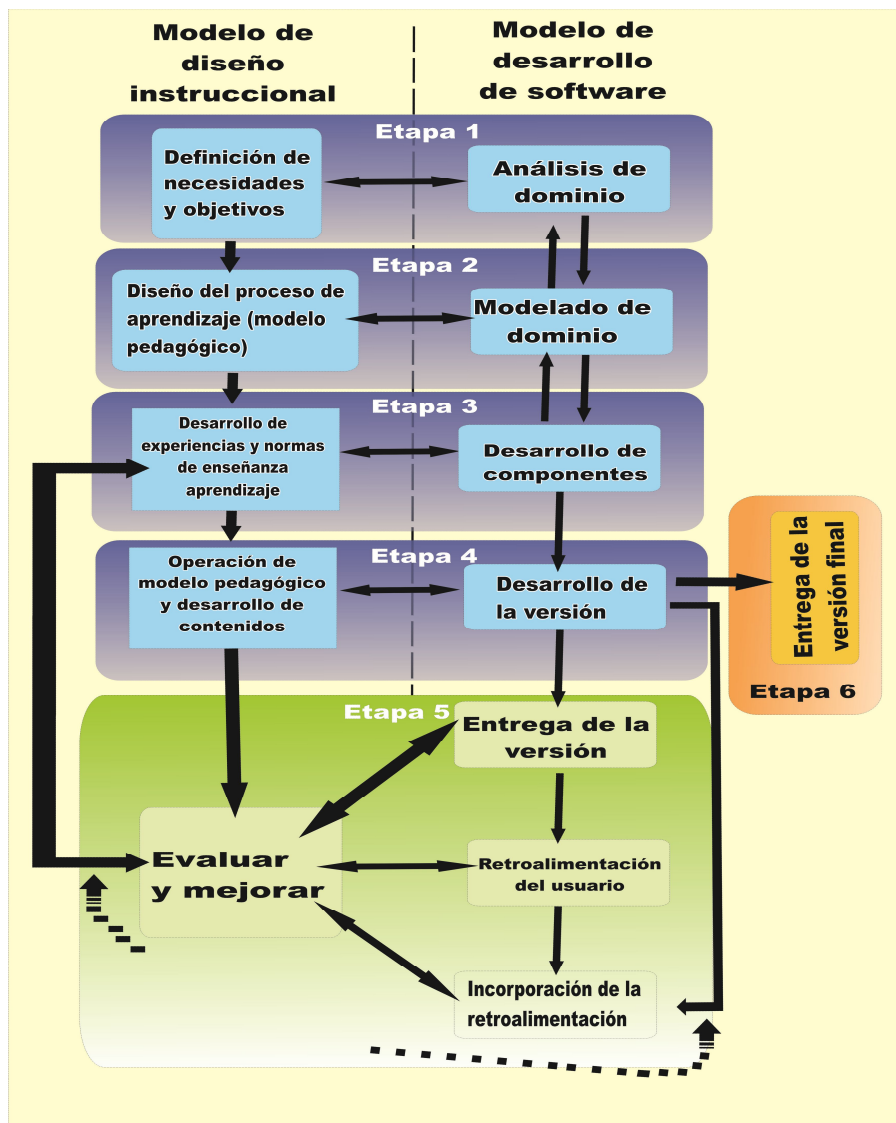


Figura 2. Ciclo de vida doble

En la tercera etapa se desarrollan las actividades de aprendizaje y el desarrollo de componentes reutilizables del sistema. Puede observarse que el ciclo de vida doble permite en estas tres etapas retroceder y avanzar entre ellas durante el desarrollo de software, con la finalidad de incorporar la información necesaria para diseñar el software hasta alcanzar las metas trazadas para cada etapa.

En la etapa cuatro se desarrolla una versión del software que posteriormente en la etapa cinco se muestra al usuario y se refina el producto en función de la retroalimentación del usuario. Este ciclo se repetirá hasta alcanzar un software adecuado a la necesidad particular, ya que la característica principal del ciclo de vida doble es ser evolutivo. Esto es muy importante ya que proporciona la posibilidad de cambiar la dirección del producto a medio camino, en respuesta a las peticiones del usuario.

### 3. Sistema de desarrollo de e- Cursos y de evaluación.

Un aspecto más a considerar en esta nueva Arquitectura es el “Sistema de desarrollo de eCursos (SIDEc)”, como una herramienta que facilita la autoría de contenido de aprendizaje, y soluciona

la carencia técnica de los profesores al simplificar la creación y publicación de cursos dentro de un ambiente Web. Otro elemento es el “*Sistema de evaluación*”, cuya funcionalidad se basa en las métricas que se obtienen del comportamiento del estudiante en tiempo de ejecución. La generación de nuevas secuencias de los cursos está en función de los resultados obtenidos, además del cálculo del nivel de adaptación. Este sistema combina los CARIIOO, meta-etiquetas adicionales y una plataforma de agentes en Java y considera tecnologías del campo de la inteligencia artificial para recrear un ambiente de semántica Web, como lo muestra la figura 3. Esta semántica se dirige a ayudar a los usuarios a lograr sus actividades en línea y ofrece ventajas como reducción de la complejidad para los desarrolladores potenciales y la estandarización de las funciones y atributos.

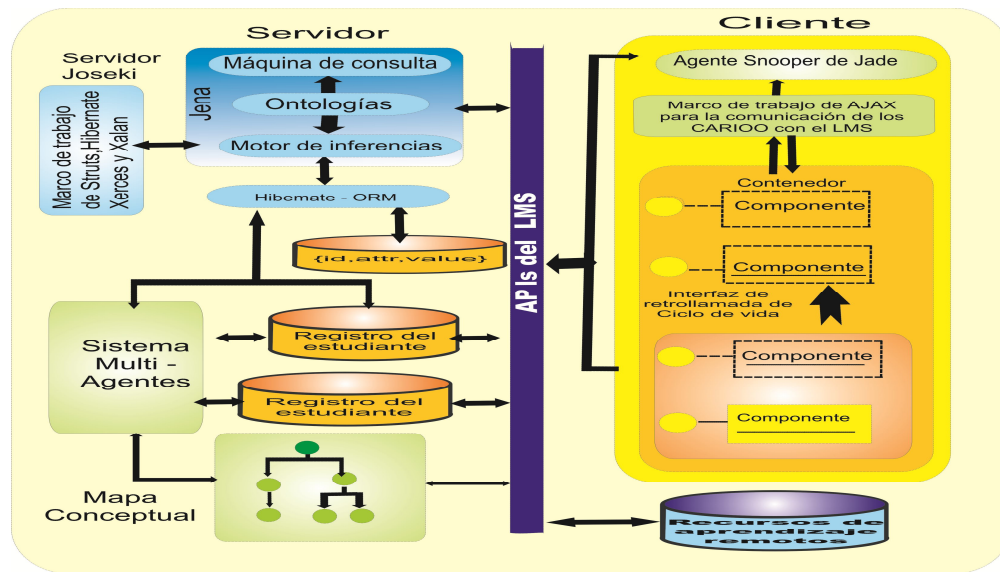


Figura 3. Plataforma semántica Web

#### 4. Conclusiones

Tanto el SIDEC como el sistema de evaluación se implementaron usando Struts (Holmes, 2004) y bajo el patrón de diseño Modelo-Vista-Controlador (MVC), mejorando la reusabilidad y mantenimiento del sistema. La generación de materiales educativos para el SIDEC y para el Sistema de evaluación los crea el Tutor considerando métricas del estudiante a lo largo de los diferentes cursos por medio de componentes CARIIOO y es evaluado en el contexto de interacciones multimedia. La evaluación produce métricas adicionales, el Tutor revisa la evaluación y busca los recursos de aprendizaje y extrae la ubicación de la información de los catálogos disponibles (URLs) y pasa las ubicaciones al proceso de envío o entrega y este proceso transforma el contenido de aprendizaje en una presentación de contenidos multimedia interactivos para el estudiante.

Este modelo de Arquitectura para el desarrollo de sistemas de EBW se centra en el estudiante y se adaptan a sus necesidades personales de forma inteligente, y se ha enfocado en la reusabilidad, accesibilidad, durabilidad e interoperabilidad de los contenidos de aprendizaje. Es un hecho que se está trabajando con seriedad en agentes inteligentes que utilizan XML como un lenguaje de comunicación y en la segunda generación de servicios Web.

#### 5. Referencias y fuentes bibliográficas.

Estrategias docentes para un aprendizaje significativo. Díaz Barriga F. Segunda edición. México D.F. Editorial MC Graw Hill. 2002

Distance Education : a System View. Grahame M. Belmont, CA.  
Wadsworth, EU. 1996. Pp. 128-132.

Desarrollo y gestión de proyectos informáticos. Mc Connel S. España. Editorial Mc Graw Hill.  
1997.

Trhee Types of Interaction. Grahame M.EU. The American Journal of Distance Education.  
1989.Pp. 1-6.

International Standars Organizations 9001 y 90003. Quality Management System and Software  
Engineering-Guide- lines for Application of ISO 9001:2000 to computer Software. 200 Edition.

IEEE 1484.1-2001Satndar for Learning Technology (LTSA) [en línea]. 2001. Disponible en:  
<http://ieee.ltsc.org/wg1>.

JOKESI Server. A. SPARQL Server for Jena [en línea]. 2005. Disponible en: <http://www.jokesi.org>.

XML. Extensible Markup Language, W3 Consortium [en línea]. 2003. Disponible en: <http://www.w3.org/XML>.