

# **Simulador Educativo para la enseñanza de gráficas cónicas en coordenadas rectangulares en el área de matemáticas asistido por agentes inteligentes**

Jorge Luis Bacca Acosta, Gloria Andrea Cavanzo, Juan Carlos Guevara

<sup>1</sup> Universidad Distrital Francisco José de Caldas, Facultad Tecnológica, Grupo de Investigación METIS. Transversal 70B 73a 35 Sur,  
Bogotá, Colombia  
jorbacca@yahoo.com, gacavanzon@udistrital.edu.co, jcguevarab@udistrital.edu.co

**Resumen.** Este documento destaca los aspectos más importantes de diseño e implementación de una herramienta de software educativo tipo simulador que fue construida para apoyar los procesos de enseñanza y aprendizaje de la temática de gráficas cónicas en coordenadas rectangulares en el área de las matemáticas. El simulador educativo está basado en un sistema multiagente con acceso a una ontología soportado en una aplicación web, presentando una propuesta innovadora en el diseño y desarrollo de software educativo para el área de matemáticas acompañada de un componente pedagógico basado en el constructivismo y un enfoque didáctico de aprendizaje basado en la resolución de problemas. Como resultado se plantea una herramienta educativa innovadora desarrollada completamente con herramientas open source y como propuesta para mejorar el desarrollo de aplicaciones para el aprendizaje en ambiente web.

**Palabras clave:** Simulador educativo, agentes inteligentes, ontología, secciones cónicas, constructivismo.

## **1 Introducción**

Los simuladores educativos han sido usados para apoyar los procesos de enseñanza-aprendizaje en diversas áreas del conocimiento debido al grado de interactividad que ofrecen, es así como se han propuesto simuladores educativos para la enseñanza de la física como el descrito en [1], en el cual se estudio la forma en que los estudiantes interactúan y aprenden en un ambiente de simulación para física, evaluando 2 tipos de retroalimentación: una gráfica con animaciones y retroalimentación textual con resultados numéricos. En esta misma área se destaca el trabajo descrito en [2], en el cual se apoya el proceso de enseñanza-aprendizaje en mecánica Newtoniana, con un componente gráfico muy interactivo y cuyos resultados demuestran que “la enseñanza utilizando un simulador educativo aumenta significativamente los resultados de los estudiantes y la retención de la temática tratada”. Otro ejemplo que se puede tomar en cuenta es el descrito en [3], en el cual se utilizó una herramienta de simulación para preparar los estudiantes en las prácticas previas de preparación para los laboratorios

de química en un ambiente real, y en el cual se evaluó el aprendizaje de los estudiantes usando la herramienta de forma sincrónica y de forma asincrónica.

En el área de la química otro de los simuladores desarrollados con un enfoque constructivista y soportado en un modelo didáctico de aprendizaje basado en la resolución de problemas, que permite al estudiante comprender de forma totalmente interactiva con un ambiente animado el proceso de montaje para destilación fraccionada y destilación simple se describe en [4].

Teniendo en cuenta los ejemplos descritos anteriormente se evidencia que los simuladores educativos son herramientas muy importantes y que ofrecen muchas ventajas tales como: ambientes animados, reducción de costos, optimización de infraestructura, aprendizaje por descubrimiento, mejora en las habilidades del estudiante, entre muchas otras que permiten que los entornos virtuales simulados faciliten para los docentes y estudiantes vivir experiencias educativas más allá de los contextos educativos clásicos del salón de clase común.

Teniendo en cuenta los aspectos anteriores el desarrollo de herramientas de carácter educativo para apoyar los procesos de enseñanza-aprendizaje para diversas áreas del conocimiento demanda el uso de tecnologías eficientes e innovadoras que permitan optimizar los recursos disponibles en los contextos educativos y con posibilidad de adaptarse a las necesidades de los estudiantes y profesores. En este sentido cabe destacar que el desarrollo de la web semántica y de la tecnología de agentes inteligentes alrededor del mundo pone a disposición de los desarrolladores herramientas altamente eficientes creando nuevas perspectivas en la implementación de software educativo con orientaciones hacia la web inteligente.

En el área de educación la Web semántica ha comenzado a tener un desarrollo importante, por ejemplo, las ontologías son usadas para “modelar dominios educativos con el objetivo de organizar y actualizar los recursos educativos como objetos de aprendizaje, perfiles de aprendizaje y rutas de aprendizaje”, como es descrito en [5], pero uno de los inconvenientes ha sido la falta de integración de esta tecnología en el desarrollo de las herramientas educativas.

Por otro lado en el área de las matemáticas los simuladores educativos son herramientas útiles para la enseñanza de temas que son difíciles de presentar en un salón de clase común debido a que a veces no se cuenta con los medios apropiados para ello o cuando las temáticas tienen un alto contenido gráfico como por ejemplo la representación de secciones cónicas tales como la parábola, la elipse, la hipérbola o la circunferencia en coordenadas rectangulares usando papel y lápiz con la precisión apropiada y cuando se requiere mostrarle al estudiante de forma clara distintas transformaciones obtenidas por la rotación de los ejes y no se cuenta con medios como una pizarra digital o una calculadora graficadora. A partir de estos aspectos se plantea la siguiente pregunta de investigación: ¿Cómo desarrollar un simulador educativo para apoyar el proceso de enseñanza-aprendizaje en la temática de gráficas cónicas en coordenadas rectangulares?

En este sentido un simulador educativo para el área de las matemáticas diseñado con una combinación equilibrada y clara entre tecnología eficiente y óptima junto con un enfoque pedagógico y didáctico moderno se convierte en una herramienta que brinda la posibilidad tanto al estudiante como al profesor de vivir diversas experiencias en los contextos de enseñanza – aprendizaje de la matemática apoyando y fortaleciendo el trabajo en el aula de clase y fuera de ella.

## **2. El simulador Educativo**

Los avances tecnológicos en el área de computación y la disponibilidad de nuevas herramientas para el desarrollo de software han permitido el surgimiento de nuevas herramientas en el campo del software educativo para apoyar los procesos de enseñanza y aprendizaje en diversas áreas del conocimiento. De acuerdo con Peré Marques el software educativo son “los programas para ordenador creados con la finalidad específica de ser utilizados como medio didáctico, es decir, para facilitar los procesos de enseñanza y de aprendizaje.” [6]. Adicionalmente el autor incluye en esta definición todos los programas que usan sistemas expertos e inteligencia artificial y cuyo objetivo principal es: “...pretenden imitar la labor tutorial personalizada que realizan los profesores y presentan modelos de representación del conocimiento en consonancia con los procesos cognitivos que desarrollan los alumnos”, por otro lado las características que debe tener el software educativo según Marques son: Facilidad de uso e instalación, versatilidad, calidad del entorno audiovisual, la calidad de contenidos, navegación e interacción, originalidad y uso de tecnología avanzada, capacidad de motivación, adecuación a los usuarios, enfoque pedagógico actual [6].

Por otro lado según Álvaro Gálvis Panqueva el software educativo “son aquellos programas que permiten cumplir o apoyar las funciones educativas y aquellos que apoyan la administración de los procesos en educación”. Las características que según él debe tener el software educativo son: Tiene en cuenta el nivel de aprendizaje, se acomoda el nivel de dominio del tema por parte del usuario, tiene la posibilidad de llenar vacíos conceptuales y promover el desarrollo de habilidades y promueve la participación activa y apropiación del conocimiento [7].

Estos conceptos de software educativo toman forma en diversas herramientas como los ambientes virtuales de aprendizaje, sistemas tutores inteligentes, entre otros incluyendo los simuladores educativos cuyo impacto en la educación estará orientado hacia un diseño más constructivista [8], siendo estas herramientas elementos muy útiles en los procesos de enseñanza-aprendizaje.

El simulador educativo propuesto tiene en cuenta los anteriores planteamientos y se pretende contar con una herramienta que permita dibujar graficas cónicas en coordenadas rectangulares de forma eficiente, interactiva y clara, para apoyar los procesos de aprendizaje de los estudiantes en esta temática. Así como permitirle al profesor identificar fortalezas y debilidades de los estudiantes que hacen uso del simulador. De forma similar se busca la optimización de recursos informáticos, de red y logísticos siendo accesible desde Internet para la comunidad de estudiantes y profesores de la Universidad Distrital Francisco José de Caldas de la ciudad de Bogotá – Colombia y de los visitantes que deseen utilizar la herramienta.

El simulador web educativo fue diseñado por componentes dentro de los cuales se destacan el componente pedagógico y didáctico, el componente tecnológico y el componente estructural. La combinación de los 3 componentes anteriores propone un modelo que ha sido diseñado pensando en el equilibrio entre tecnología y pedagogía, para obtener una herramienta funcional, óptima y adecuada a las necesidades de los estudiantes y profesores, estos componentes se describen en las siguientes secciones.

## **2.1 Componente Estructural**

El componente estructural hace referencia a la división por módulos del sistema para facilitar el diseño e implementación del mismo, en este caso la herramienta educativa cuenta con los siguientes módulos: Sistema de registro, sistema didáctico, sistema de seguimiento, sistema de simulación, sistema teórico y sistema de ayuda. Las principales características de cada uno de los sistemas se describen a continuación:

- El sistema de registro es responsable de administrar el acceso de los usuarios como por ejemplo el administrador, los profesores y los estudiantes permitiendo un acceso seguro y personalizado, habilitando un perfil web para cada uno. En este sistema el administrador tiene la posibilidad de definir los cursos, los estudiantes para cada curso y el profesor a cargo de cada curso. Finalmente por medio de este sistema el administrador tiene la posibilidad de manejar la plataforma de agentes.
- El sistema didáctico fue diseñado con base en el modelo constructivista y en el modelo de aprendizaje basado en la resolución de problemas [9], sobre este sistema el estudiante puede proponer un problema a ser solucionado o seleccionar uno del sistema y adicionalmente proponer una hipótesis acerca de la posible solución al problema, esto se realiza siempre antes del proceso de simulación. Este sistema se encuentra en conexión directa con el sistema de seguimiento y el sistema de simulación para proporcionar un reporte acerca de los problemas y las hipótesis planteadas por los estudiantes, este reporte es presentado al profesor por cada uno de los cursos que tenga a cargo.
- El sistema de simulación le permite al estudiante ver la gráfica de una sección cónica en el sistema de coordenadas rectangulares, una vez el estudiante define los coeficientes de la ecuación que describe la sección cónica. Este sistema está compuesto por 2 agentes inteligentes, uno de ellos es el agente monitor, el cual valida y maneja las acciones del estudiantes sobre el simulador. El otro agente utiliza un motor de inferencia para consultar la ontología de secciones cónicas para hallar el tipo de sección cónica que debe ser dibujada.
- El sistema de seguimiento está conectado con los sistemas de simulación y didáctico; este sistema le permite a los profesores la creación de 2 tipos de test, uno de ellos es el test de ideas previas para conocer que sabe el estudiante antes del proceso de simulación, y el otro test es el test de ideas posteriores, el cual es usado para saber lo que el estudiante comprendió después del proceso de simulación. Este sistema también crea un reporte de los resultados cuando los estudiantes solucionan los test y con información de las simulaciones realizadas por los estudiantes.
- El sistema teórico provee información sobre la temática, en este caso acerca de secciones cónicas en coordenadas rectangulares, estos contenidos pueden ser consultados por el estudiante en cualquier momento para apoyar el proceso de aprendizaje cuando se usa el simulador.

- El sistema de ayuda proporciona información acerca de cómo usar el simulador, explicando los diferentes componentes y su respectiva funcionalidad.

## **2.2 Componente Pedagógico y Didáctico**

El componente pedagógico del proyecto gira en torno al modelo pedagógico constructivista y al modelo didáctico de resolución de problemas, de tal manera que mediante el uso del simulador se presenta al estudiante problemas centrados en el aprendizaje por descubrimiento e investigación, orientados por hipótesis que plantea el estudiante y por mecanismos de resolución de dichos problemas, a partir de experiencias y conocimientos previos en el tema de graficas cónicas y nuevos conocimientos proporcionados por la investigación y las experiencias con el uso del simulador.

Estos elementos permiten que el estudiante se cuestione, comprenda y analice sus conocimientos, además de descubrir e interpretar nuevos conocimientos provenientes del uso de la herramienta educativa. Estas características están dadas por el sistema didáctico descrito anteriormente.

Adicionalmente el profesor por medio del “sistema de seguimiento” puede conocer en tiempo real los resultados que obtiene el estudiante, así como los problemas que ha intentado resolver y sus hipótesis al respecto, de tal manera que a partir de estos datos el profesor tiene la capacidad de identificar las fortalezas y debilidades de sus estudiantes y de esta manera orientar su clase de una mejor manera. El sistema le permite al profesor colocar observaciones a sus estudiantes sobre cada reporte de simulación de tal manera que los estudiantes reciben una retroalimentación del proceso de simulación que realizaron y en este sentido ir construyendo su conocimiento. Adicionalmente la herramienta le permite al profesor conocer las ideas previas de un estudiante antes de realizar las simulaciones y de igual manera conocer las ideas posteriores de los estudiantes después de las simulaciones, todo esto por medio de los test que es posible definir. Teniendo en cuenta estos aspectos el simulador está diseñado de una forma particular, proponiendo un modelo que difiere de los tradicionales simuladores, en los cuales simplemente se realizan las simulaciones y no se tiene en cuenta el proceso de retroalimentación para el estudiante que es uno de los factores más importantes en el proceso de enseñanza [10] ni el seguimiento que hace el profesor de la actividad, el cual resulta ser muy importante en el proceso de enseñanza [11] y marca la diferencia del sistema que se propone en relación a la mayoría de los sistemas de software existentes para la enseñanza de la temática de gráficas cónicas.

## **2.3 Componente Tecnológico**

El simulador educativo es una aplicación web dinámica desarrollada bajo la especificación J2EE 1.4 [12] con el framework Java Server Faces. El servidor de aplicaciones usado para desplegar la aplicación es Jboss con Apache Tomcat 5.5. Adicionalmente el simulador educativo utiliza agentes inteligentes [13] para mejorar

el desempeño de la aplicación en los procesos de simulación, reduciendo el consumo de recursos por parte de los clientes y fortaleciendo la retroalimentación sobre los estudiantes.

El sistema de simulación ha sido diseñado con 2 agentes inteligentes: el agente de simulación y el agente monitor. La tarea principal del agente monitor es validar y manejar las acciones del estudiante sobre el simulador, por ejemplo cuando el estudiante va a simular una sección cónica, el agente monitor toma los parámetros de la ecuación definidos por el estudiante y envía un mensaje al agente de simulación preguntando por la ecuación que será dibujada. Por otro lado el agente de simulación recibe el mensaje y su tarea principal es consultar la ontología [14], la cual es la base de conocimientos del agente, y usando el motor de inferencia del framework de web semántica de Jena (Semantic web Framework for java) [15], para “conocer” la sección cónica descrita por la ecuación, esta consulta es hecha por medio del lenguaje SPARQL [16]. Por otro lado, las reglas para el motor de inferencia han sido diseñadas para funcionar con encadenamiento adelante y encadenamiento hacia atrás.

La comunicación con el motor de inferencia de Jena es permitida usando la interfaz de programación AgentOWL [17], usando llamados por medio de XML-RPC.

Los agentes son ejecutados sobre la plataforma de agentes JADE [18], la cual provee servicios tales como el de páginas amarillas, el servicio de agente sniffer, el agente de monitoreo remoto para manejar el funcionamiento de la plataforma. La plataforma de agentes es desplegada sobre el servidor de aplicaciones y los agentes son creados sobre el contenedor. Los estudiantes y profesores usan la herramienta educativa a través de internet desde cualquier lugar y en cualquier momento. Debido a que los agentes y la plataforma JADE son ejecutados del lado del servidor, el consumo de recursos es reducido porque el cliente solo necesita un navegador de internet para usar todas las funcionalidades de la herramienta educativa. Además el sistema ha sido diseñado para soportar un gran volumen de usuarios accediendo a la aplicación de forma concurrente

La ontología usada por los agentes fue diseñada por medio del modelado del conocimiento acerca de secciones cónicas y fue implementada usando el editor de ontologías Protégé desarrollado por el Stanford Center for Biomedical Informatics Research y teniendo en cuenta algunos lineamientos para la construcción de ontologías definidos en [19]. En esta ontología se definieron los conceptos básicos sobre secciones cónicas, tales como parábola, hipérbola, circunferencia, elipse, de tal forma que los agentes puedan tener acceso a estos conceptos a través del motor de inferencial y puedan determinar el tipo de sección cónica que el estudiante desea ver. La ontología fue definida en el lenguaje OWL-DL.

### 3 Proceso de Simulación de una Sección Cónica

En esta sección se describirá el proceso que sigue un estudiante cuando interactúa con la herramienta educativa y desarrolla un ejercicio completo para obtener como resultado la gráfica de una sección cónica que puede ser una parábola, una hipérbola, una circunferencia o una elipse.

El proceso de simulación comienza cuando el estudiante ingresa a la aplicación y selecciona la opción de simulación, el primer paso es seleccionar o proponer un problema acerca de la temática de secciones cónicas en coordenadas rectangulares de acuerdo con el modelo de aprendizaje basado en la resolución de problemas. Posteriormente el estudiante debe proponer una hipótesis acerca del problema que se busca solucionar. Si el profesor ha definido un test de ideas previas para los estudiantes del curso entonces el siguiente paso para el estudiante es solucionar dicho test, si no, el estudiante continúa hacia la fase de simulación final. El siguiente paso es el proceso de simulación como tal, en este paso, el estudiante puede definir los coeficientes de la ecuación que describe una sección cónica con el fin de obtener la gráfica correspondiente, en esta parte el sistema de agentes inteligentes comienzan a trabajar con la ontología y para cada valor que el estudiante define el sistema multiagente muestra el tipo de sección cónica que será dibujado usando el motor genérico de inferencia de Jena.

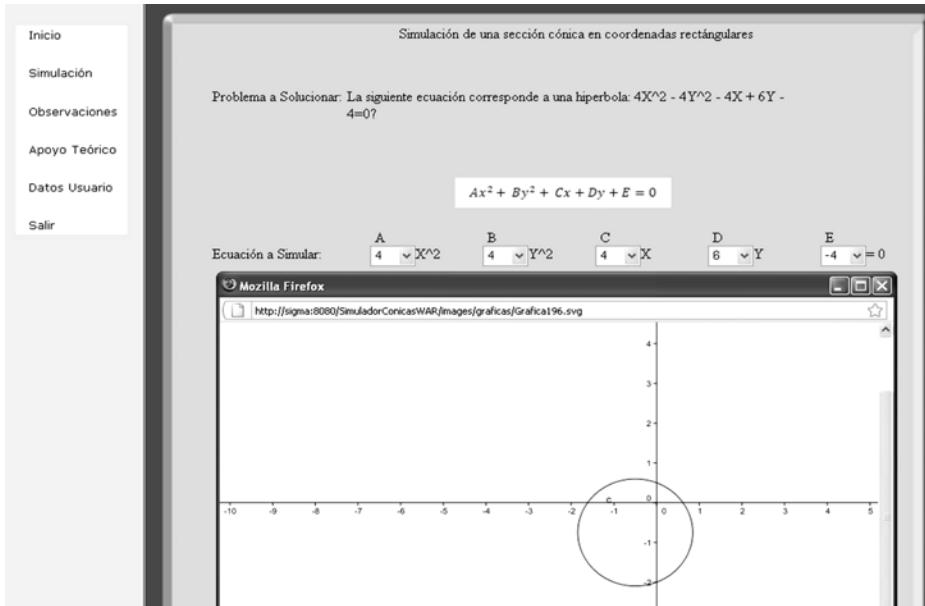
Otro de los aspectos importantes en este punto es que el sistema le muestra al estudiante una explicación en tiempo real de porque se genera cada uno de los tipos de secciones cónicas a medida que va definiendo los coeficientes de la ecuación, esto se realiza debido a que es importante la retroalimentación para el estudiante, es decir que el estudiante sepa e identifique porqué motivo se genera un tipo de sección cónica y porque razones el tipo de sección cónica cambia cuando se modifican los valores de la ecuación. Estos aspectos hacen del simulador educativo una herramienta muy eficiente donde el proceso de retroalimentación para el estudiante es vital en el proceso de enseñanza-aprendizaje de ésta temática. Una vez el estudiante ha definido los valores para la ecuación de una sección cónica (1), el sistema multiagente consulta la ontología, usando el motor de inferencial de Jena y muestra la grafica al estudiante.

Cuando el estudiante termina el proceso de simulación, el siguiente paso es resolver el test de ideas posteriores para verificar cambios en los conceptos aprendidos por los estudiantes.

Una vez el estudiante ha terminado la fase de simulación, el sistema de seguimiento crea un reporte que contiene los siguientes datos: el problema y la hipótesis propuesta, las respuestas de los test de ideas previas y de ideas posteriores y un resumen de las simulaciones realizadas por el estudiante. Este reporte es mostrado en el perfil del profesor y el profesor puede añadir un comentario para los estudiantes, dicho comentario aparecerá en el perfil del estudiante. El comentario del profesor es importante ya que le sugiere al estudiante ejercicios adicionales con el fin de fortalecer los conceptos en los cuales tiene alguna debilidad.

$$Ax^2 + By^2 + Cx + Dy + E = 0 . \quad (1)$$

La figura 1 muestra el perfil del estudiante cuando la gráfica de la sección cónica es presentada al estudiante luego de seleccionar los valores de cada uno de los coeficientes de la ecuación. Adicionalmente en este paso del proceso de simulación se le muestra al estudiante el problema que está intentando solucionar con el objetivo de que su aprendizaje este centrado en la solución del problema específicamente.



**Fig. 1.** Perfil del estudiante durante la simulación de una sección cónica.

## 4 Metodologías

El marco metodológico del proyecto está definido a partir de la combinación entre distintas metodologías particulares, entre las que se encuentran, la metodología para el desarrollo de software educativo de Gálvis Panqueva [7], el Ontology Development Process [20] del Methontology Framework, la metodología MasCommonKADS [21] en el aspecto de los agentes de software y la metodología R.U.P. El cuadro metodológico establecido para el diseño e implementación de la herramienta se muestra en la tabla 1. Es importante destacar que la metodología sigue un proceso iterativo de tal manera que en cada iteración se van mejorando diversos aspectos del software dependiendo del análisis inicial de requerimientos.

**Tabla 1.** Marco metodológico para el desarrollo del simulador educativo.

Metodología	Actividades	Descripción
Metodología de Software Educativo de Álvaro Gálvis Panqueva.	Análisis de las necesidades educativas.	Se analiza el entorno educativo, se identifican las principales falencias de los estudiantes y los puntos en los cuales se necesita el apoyo de la herramienta educativa
	Planeación del desarrollo del sistema	Se define el modelo pedagógico y didáctico apropiados para el caso. Se define la estructura general de sistema.

		Selección de las tecnologías para el desarrollo del software educativo.
	Diseño del material educativo	Definición de las funciones que debe cumplir el material educativo. Definición de la estructura de control del software educativo.
	Diseño educativo del sistema	Definición de las metodologías de enseñanza de acuerdo con los modelos pedagógico y didáctico. Cómo monitorear el aprendizaje del estudiante.
	Implementación del material educativo	Codificación y desarrollo del software teniendo en cuenta el diseño propuesto con anterioridad.
	Pruebas piloto del material educativo	Pruebas del software en ambientes educativos reales para evaluar su eficiencia.
Metodología para el diseño de ontologías Methontology	Definición del Glosario de términos	Lista de términos que se usan en el dominio del sistema a modelar.
	Especificación de la ontología	Clasificación de los conceptos del dominio y establecimiento de sus relaciones.
	Definición de clases, relaciones, cardinalidad y reglas de inferencia.	Formalización de las clases del dominio a partir de los conceptos, diseño de la cardinalidad y diseño de las reglas de inferencia.
	Implementación de clases, relaciones y reglas de inferencia.	Codificación de la ontología en un lenguaje y editor apropiado e implementación de las reglas de inferencia de acuerdo con el motor de inferencia seleccionado.
	Evaluación de la ontología.	Pruebas de la ontología.
	Pruebas de inferencia	Integración de la ontología con el motor de inferencia y las reglas de inferencia.
Metodología MasCommon-KADS	Definición de casos de uso.	Definición y documentación de los casos de uso de los agentes inteligentes del sistema.
	Modelos de agente, de tareas, de conocimiento, de comunicación y de organización.	Desarrollo de los modelos de cada agente con sus comportamientos y objetivos, modelo de tareas a desarrollar por cada agente, modelos de comunicación entre los agentes y organización en el sistema multiagente.

	Especificación de Servicios de red	Definición de los mecanismos de funcionamiento del sistema de agentes sobre la red y comunicación remota.
	Diseño de agentes y configuración de la plataforma de agentes.	Diseño de cada uno de los agentes y configuración de la plataforma de agentes para su funcionamiento sobre el servidor apropiado.
	Implementación de agentes	Codificación de los agentes en un lenguaje de programación.
	Pruebas de funcionalidad.	Pruebas de comunicación y operación de los agentes en el ambiente implementado.
RUP – Desarrollo de software.	Definición de requerimientos	Especificación de los requerimientos de implementación del sistema de software.
	Identificación de actores	Descripción de los actores que interactúan con el sistema.
	Lista de casos de uso	Definición y documentación de los casos de uso por cada actor.
	Diagrama de clases	Desarrollo del modelo de clases integrando todas las clases que conforman el sistema.
	Diagramas de secuencia, actividad, colaboración, estado	Desarrollo de los diagramas de actividad, colaboración, secuencia, estado que representan la relación y comunicación entre las clases.
	Modelo Físico y Lógico	Definición de los modelos lógico y físico del sistema.
	Modelo Objeto-Relacional	Definición del modelo de base de datos del sistema.
	Implementación de las clases	Codificación de todas las clases e interfaces que integran el sistema.
	Pruebas de sistema y de integración.	Pruebas de funcionamiento del sistema base de software.

## 5 Resultados

Como resultado del proceso de diseño e implementación de la herramienta educativa se obtuvo un modelo para el desarrollo de simuladores educativos utilizando agentes inteligentes y web semántica, el cual puede ser usado para el desarrollo de simuladores educativos para diversas áreas del conocimiento.

Adicionalmente se obtuvo un marco metodológico por etapas integrando diversas metodologías y que permite obtener como resultado un producto innovador utilizando diversas tecnologías con el fin de apoyar los procesos de enseñanza y aprendizaje en diversas áreas del conocimiento.

Finalmente se obtuvo una herramienta innovadora integrando las tecnologías de agentes inteligentes y web semántica para apoyar los procesos de enseñanza y aprendizaje en la temática de gráficas cónicas en coordenadas rectangulares para el área de matemáticas.

## 6 Conclusiones

La implementación de la herramienta educativa descrita en este artículo propone una alternativa en el desarrollo de simuladores educativos para el campo de las matemáticas usando agentes inteligentes y web semántica y planteando un esquema de diseño que puede ser usado para desarrollar simuladores óptimos para fortalecer los procesos de enseñanza y aprendizaje en matemáticas en los colegios y universidades. Por otro lado mediante el desarrollo del simulador educativo se exploran nuevas perspectivas en el desarrollo de herramientas para apoyar los procesos de enseñanza – aprendizaje en ambientes educativos en la web.

Los agentes inteligentes son elementos versátiles, eficientes y pueden ser usados para resolver diferentes tipos de problemas y es una alternativa para afrontar desafíos de la ingeniería de software en cuanto al desarrollo de aplicaciones.

El uso de tecnologías de alto desempeño junto con un modelo pedagógico y didáctico moderno permite el desarrollo de aplicaciones educativas en ambiente web eficientes para optimizar los recursos tecnológicos disponibles y al mismo tiempo fortalecer los procesos de enseñanza y aprendizaje.

El desarrollo de software educativo utilizando agentes inteligentes y web semántica mejora el desempeño de dichas aplicaciones debido a que los sistemas de seguimiento y retroalimentación de conceptos para el estudiante son más efectivos y se puede mejorar la respuesta del software hacia sistemas más adaptativos.

## 6. Referencias

1. Lloyd P. Rieber, Shyh-Chii Tzeng, Kelly Tribble.: Discovery learning, representation, and explanation within a computer-based simulation: finding the right mix. In: Learning and Instruction, Volume 14, Issue 3, Dynamic Visualisations and Learning. Pp. 307-323, ISSN 0959-4752. (2004)
2. Athanassios, J., Vassilis K.: Computer simulations in physics teaching and learning: a case study on students' understanding of trajectory motion. In: Computers & Education, Volumen 36, Serie 2, Pp. 183-204, ISSN 0360-1315. (2004)
3. Maria L., Nikos P., Christopher W.: Integration of simulation into pre-laboratory chemical course: Computer cluster versus WebCT. In: Computers & Education, Volume 52, Issue 1, Pp. 45-52, ISSN 0360-1315 (2009).
4. Guevara, J.C.: "Model of simulator for the teaching of chemical distillation," In: Proc. of the 39th IEEE Frontiers in Education Conference, 2009. FIE '09. pp.1-6, 18-21 Oct. (2009).
5. M. Gaeta, F. Orciuoli, P. Ritrovato.: Advanced ontology management system for personalised e-Learning. In: Knowledge-Based Systems, Volume 22, Issue 4, Artificial Intelligence (AI) in Blended Learning - (AI) in Blended Learning, Pp 292-301, ISSN 0950-7051 (2009).

6. Marques, P.: El software educativo, [http://www.lmi.ub.es/te/any96/\\_marques\\_software/#capitol1](http://www.lmi.ub.es/te/any96/_marques_software/#capitol1). (2003).
7. Galvis, Álvaro.: “Ambientes de enseñanza-aprendizaje enriquecidos con computador”.[http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-126374\\_archivo.pdf](http://www.colombiaaprende.edu.co/html/mediateca/1607/articles-126374_archivo.pdf) (1998).
8. Magee, M.: State of field review: Simulation in education. Alberta Online Learning Consortium, Canada (2006).
9. Kiley, M. et al., “Problem Based Learning”. The University of Adelaide. Australia. <http://www.adelaide.edu.au/clpd/resources/leap/leapinto/ProblemBasedLearning.pdf>. (2007)
10. Heuvelink, A., Mioch, T.: FeGA: a Feedback-Generating Agent. in: The IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology, pp. 567-572. IEEE Press. (2008)
11. Zeng, B., Nie, J., y Wei, J.: Design for Active Monitor System in Distance Learning Environments. In: The Seventh International Conference on Information Technology 2010, pp.809 – 813. IEEE Press. (2010)
12. Shannon, Bill. (2003). “Java 2 Platform Enterprise Edition Specification, v1.4”. [http://java.sun.com/j2ee/j2ee-1\\_4-fr-spec.pdf](http://java.sun.com/j2ee/j2ee-1_4-fr-spec.pdf)
13. Barella, A., Carrascosa, C., Botti, V.: Agent Architectures for intelligent virtual environments. In: IEEE/WIC/ACM International Conference on Intelligent Agent Technology, pp. 532-535. IEEE Press. (2007)
14. Gruber, Tom: “What is an Ontology?”. <http://www-ksl.stanford.edu/kst/what-is-an-ontology.html> (2007)
15. JENA – A Semantic Web framework for Java. <http://jena.sourceforge.net/documentation.html>
16. Seaborne, Andy. Prud'hommeaux, Eric. (2008) “SPARQL Query Language for RDF” <http://www.w3.org/TR/rdf-sparql-query/>
17. Michael, Laclavick., Balogh, Zoltan. Babik, Marian: “AgentOWL: Semantic Model and agent Architecture”. In: Computing and informatics, Vol. 25, pp. 419 - 437. (2007)
18. Bellifemine, F. et al.: JADE Programmer’s Guide . Java agent development framework. DOI: <http://jade.tilab.com> (2007)
19. Noy, N. F., et. al.: Creating Semantic Web Contents with Protégé 2000. In: IEEE Intelligent Systems, March – April. pp. 60-71. IEEE Press. (2001)
20. Blázquez, M. et. al: Building Ontologies at the Knowledge Level using the Ontology Design Environment. <http://ksi.cpsc.ucalgary.ca/KAW/KAW98/blazquez/>. (2007)
21. Iglesias, Carlos. “Definición de una metodología para el desarrollo de sistemas multiagente”. <http://www.gsi.dit.upm.es/tesis/pdf/tesiscif.pdf>. (1998)