

Diseño e implementación de un sistema de aprendizaje inteligente y afectivo para la enseñanza de las matemáticas del segundo año de primaria

Ramón Zatarain Cabada, María Lucía Barrón Estrada, Francisco González Hernández,
Raúl Oramas Bustillos

Instituto Tecnológico de Culiacán, Juan de Dios Bátiz s/n, Col. Guadalupe, Culiacán Sinaloa,
80220, México
{rzatarain, lbarron}@itculiacan.edu.mx

Resumen. En este artículo se presenta un sistema de aprendizaje inteligente y afectivo, que se integra en una red social para el aprendizaje de las matemáticas. El sistema está diseñado para ayudar a los estudiantes de segundo grado de educación primaria a mejorar su proceso de aprendizaje. El sistema evalúa aspectos cognitivos y afectivos del estudiante mediante una red neuronal y utiliza un sistema experto difuso para decidir el siguiente ejercicio que deberá resolver el estudiante, lo que posibilita un aprendizaje personalizado.

Palabras clave: Sistema Tutor Inteligente, Redes Sociales, Red Neuronal, Sistema Experto, Reconocimiento de Emociones.

1 Introducción

Las matemáticas son parte fundamental de la formación básica de los estudiantes en cualquier grado escolar. Los informes publicados sobre el rendimiento en matemáticas por la OCDE[1] y SEP[2] en el 2010 reflejan que existe un alto porcentaje de alumnos con fracaso escolar; en otros estudios, se observa una falta generalizada de conocimientos matemáticos, habilidades cognitivas, ausencia de motivación, interés y afecto[3].

Dentro del campo de la educación, tradicionalmente solo se consideraban los estados cognitivos o pedagógicos del estudiante. En el campo de las ciencias computacionales, hasta hace poco tiempo, los sistemas tutores inteligentes (STI) no consideraban el estado afectivo de un estudiante.

En la actualidad se considera que el estado afectivo o emocional juega un papel importante dentro del proceso enseñanza-aprendizaje de los estudiantes. Conocer el estado emocional de un estudiante, le otorga a un sistema de software basado en cómputo afectivo capacidad de decisión sobre cómo debe responder ante él. El reconocimiento automático de las emociones puede mejorar el desempeño, usabilidad y, en general, la calidad de

interacción hombre-computadora, la productividad del aprendizaje de los estudiantes, y la atención de un sistema a los usuarios [4,5,6].

En este artículo presentamos la implementación de un sistema de aprendizaje inteligente y afectivo para el aprendizaje de las matemáticas de segundo año de primaria que se integra en una red social educativa. El sistema hace uso de una red neuronal para reconocer el estado afectivo del estudiante y un sistema experto difuso que integra datos cognitivos del estudiante (como errores cometidos, tiempo y número de ayudas al resolver un ejercicio) con datos afectivos como su último estado emocional. Esto permite al STI calcular la complejidad del siguiente ejercicio que resolverá el estudiante.

La organización del artículo es la siguiente: en la sección 2, describimos el diseño e implementación del sistema con su arquitectura de capas, componentes y algoritmos. Los resultados y pruebas de la aplicación se muestran en la sección 3 y las conclusiones y trabajos futuros se discuten en la sección 4.

2 Diseño e implementación del sistema

La estructura del STI sigue el modelo tradicional de este tipo de sistemas [7], el cual contiene cuatro componentes principales: una interfaz de usuario, representada por la red social de aprendizaje, desde donde se establece la comunicación con los otros tres módulos llamados experto (o dominio), tutor y estudiante.

En un escenario normal del uso del sistema, el estudiante accede al STI a través de la red social e interactúa con una interfaz propia que se encarga de mostrar los temas y ejercicios además de tomar fotografías para reconocer el estado emocional del estudiante.

Se agregan también detalles de la implementación usadas para realizar los componentes diseñados en la arquitectura del sistema. La implementación se realiza sobre una red social de aprendizaje que se encuentra en un ambiente web. La arquitectura de todo el sistema se muestra en la figura 1.

A continuación se describen cada una de las capas así como los componentes que conforman al sistema.

2.1 Capa de presentación

La *Capa de Presentación* representa la interfaz gráfica del STI; su tarea es interactuar con el usuario a través de una interface web en una computadora ó dispositivo móvil; en la pantalla se muestran los ejercicios que el usuario debe resolver. Por cada respuesta se obtiene información para mostrar el siguiente ejercicio. Esta capa contiene tres componentes:

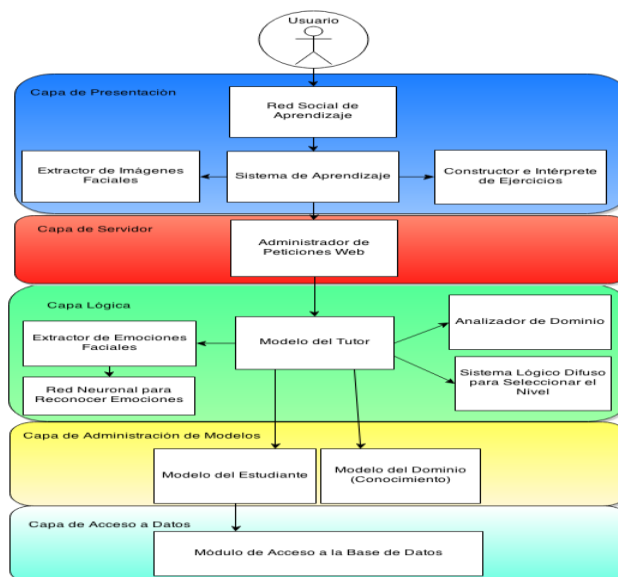


Fig. 1. Arquitectura del STI.

- **Extractor de Imágenes Faciales:** tiene la tarea de extraer una imagen del rostro del usuario cada cierto intervalo de tiempo y regresarla al *Sistema de Aprendizaje*. El formato de la imagen es JPEG y utiliza la API de manejo de cámara de HTML 5 para evitar el uso de complementos de terceros.
- **Sistema de Aprendizaje:** inicia la interacción con el usuario. En él se presenta la interfaz de los ejercicios de matemáticas a resolver (ver figura 2), así como las ayudas e información de la meta que el estudiante debe conseguir. El componente recolecta los datos cognitivos generados por el usuario, así como la imagen obtenida del *Extractor de Imágenes Faciales*. Los datos que se obtienen se envían al componente *Administrador de Peticiones Web* que se encuentra en la capa del servidor. Una vez procesados los datos, el administrador envía la respuesta de regreso y el componente *Sistema de Aprendizaje* utiliza el componente *Constructor e Intérprete de Ejercicios* para generar la nueva representación visual de los datos recibidos. La representación del menú principal del STI (ver figura 2) puede ser una página web de escritorio o móvil. Este componente utiliza tecnologías como jQuery Mobile, HTML 5, CSS 3 y JavaScript.
- **Constructor e Intérprete de Ejercicios:** El componente realiza dos tareas: La primera es interpretar los datos generados por el *Modelo del Tutor*, enviados desde el componente *Administrador de Peticiones Web*. Durante esta interpretación genera los contro-

les que utilizara el *Sistema de Aprendizaje*. Cada ejercicio es interpretado y construido de forma distinta.



Fig. 2. Página web móvil del sistema de aprendizaje.

2.2 Capa de servidor

La *Capa de Servidor* se encarga de procesar las peticiones web realizadas por los clientes al servidor y envía una respuesta para que los clientes puedan realizar la interpretación de los datos enviados. La capa se encarga de la parte tecnológica web. Esta capa contiene solo un componente que utiliza tecnologías Java Web por medio de Servlets.

- Administrador de Peticiones Web: este componente construye por medio de los datos enviados desde la capa de presentación los objetos necesarios para el modelo del tutor. También se encarga de tomar la respuesta del tutor y enviarla a los clientes.

2.3 Capa lógica

La *Capa Lógica* implementa el procesamiento de la información del sistema. Contiene los componentes que realizan la toma de decisiones sobre los datos del estudiante, dicha información corresponde sus datos cognitivos y afectivos.

En la capa lógica se encuentran la parte de inteligencia artificial, ya que aquí es donde se utilizan los componentes de la red neuronal para la extracción de emociones y el sistema lógico difuso para la elección de niveles en los ejercicios resueltos. Los componentes son:

- **Modelo del Tutor:** es el componente principal de todo el sistema, ya que es el componente que ejecuta el algoritmo principal para la construcción de los ejercicios (ver figura 3) que se enviarán al *Intérprete de Ejercicios*. El componente utiliza la información entregada por los componentes de *Extractor de Emociones Faciales*, *Analizador de Dominio* y *Sistema Lógico Difuso* que se encuentran en la capa.
- **Extractor de Características Faciales:** este componente analiza una imagen y extrae en búsqueda de las características del rostro. Estas características forman un conjunto de 10 coordenadas. Estas coordenadas se envían al componente de la *Red Neuronal para reconocer Emociones* y devolver la clasificación encontrada de dichas características.
- **Red Neuronal para Reconocer Emociones:** este componente utiliza una red neuronal de propagación hacia atrás que fue entrenada con un corpus de emociones faciales que contiene varios conjuntos de 10 coordenadas cada uno. El componente recibe un conjunto de 10 coordenadas enviadas por el *Extractor de Características Faciales*, posteriormente envía dichas coordenadas a la red que está previamente entrenada con un corpus de coordenadas. Finalmente se envía de vuelta la clasificación encontrada por la red para el conjunto de coordenadas provistas.
- **Analizador de Dominio:** tiene la tarea de analizar un dominio enviado por el *Modelo del Tutor*. El componente debe tener el conocimiento de cómo realizar las distintas tareas de los distintos dominios y formatos. El dominio de matemáticas de segundo año de primaria se encuentra en un archivo con formato de XML. Se utiliza XPath para realizar el análisis de este archivo.
- **Sistema Lógico Difuso para Seleccionar el Nivel del Ejercicio:** recibe los datos cognitivos enviados por el *Modelo del Tutor*, donde realiza la evaluación de las distintas reglas difusas. El componente utiliza la API jFuzzyLogic [8]. La API utiliza un archivo de configuración que contiene la definición de las variables de entrada, de salida, el método de defusificación y las reglas difusas.

2.3.1 Descripción del algoritmo

La figura 3 presenta el *Modelo del Tutor* con anotaciones que describen el flujo lógico de ejecución para la construcción de los ejercicios que serán enviados al *Constructor e Intérprete de Ejercicios*.

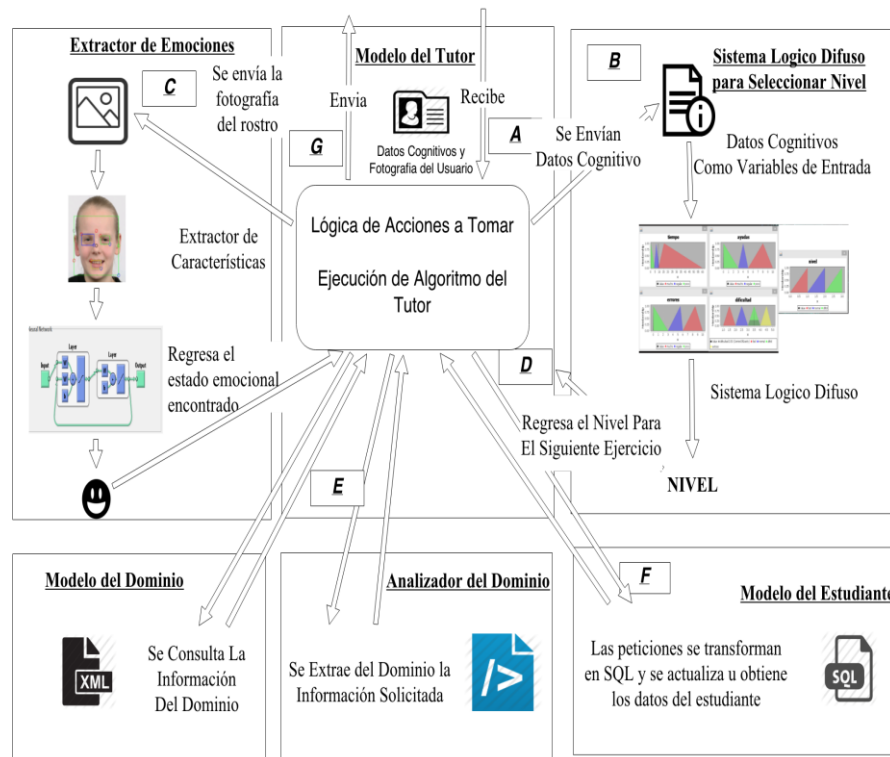


Fig. 3. Modelo del tutor de acuerdo al algoritmo.

El siguiente algoritmo muestra los pasos llevados a cabo por el *Modelo del Tutor* en la toma de decisiones para formar los ejercicios devueltos:

- A. El *Modelo del Tutor* recibe un objeto estudiante, el cual contiene los datos cognitivos, la imagen facial y datos sobre el ejercicio resuelto por el estudiante.
- B. El *Modelo del Tutor* envía los datos cognitivos del ejercicio resuelto al *Sistema Lógico Difuso* quien los recibe como entrada para las variables de entrada difusas y regresa como resultado la variable difusa de salida al *Modelo del Tutor*. Esta variable representa el nivel recomendado para el siguiente ejercicio.
- C. El *Modelo del Tutor* envía la imagen facial del usuario al componente de *Extractor de Características Faciales*. Este utiliza el método Haar-like features Cascades [9] (HLC) el cual crea áreas de imágenes conocidas como regiones de interés (RDI). Posteriormente se vuelve a aplicar dicho método a las RDI detectadas, esta vez para las partes del rostro que

son la boca, ojos y cejas. Para la boca y los ojos se le ubican cuatro coordenadas que son la parte más altas y bajas y las partes más a la izquierda y a la derecha. Para las cejas se realiza el mismo procedimiento pero solo con 3 coordenadas. Después se dibujan líneas entre las distintas coordenadas. Esas líneas representan “distancias”. Por cada distancia se aplica el teorema de Pitágoras entre dos coordenadas y se guarda el resultado de la hipotenusa de los triángulos. Cada una de las distancias son divididas por el ancho del rostro. Esta división normaliza los valores de las distancias entre 0 y 1.

Las distancias se crean y guardan en un nuevo archivo de texto con una letra que representa la emoción y las distancias. Termina la extracción de características del rostro y se repite todo el procedimiento para un rostro nuevo. El archivo de texto es mandado a una red neuronal previamente entrenada. La red evalúa y determina la emoción de acuerdo al archivo de texto y la representa con un valor. Este valor es devuelto al Modelo del Tutor.

- D. El *Modelo del Tutor* empieza a tomar decisiones sobre el siguiente ejercicio que va a construir, considerando los valores de la emoción y del nivel que se obtuvieron en los pasos B y C. La construcción de un ejercicio implica mandar ayudas, datos, metas y recursos relacionados con el ejercicio.
- E. El Modelo del Tutor solicita un dominio en particular al componente de modelo de dominio que se encuentra en la capa de administración de modelos. En este caso pide el dominio de matemáticas del segundo año de primaria. Posteriormente envía dicho modelo al componente Analizador de Dominio junto con los parámetros de lo que necesita realizar u obtener. Posteriormente el analizador de dominio ejecuta dichas instrucciones y devuelve el resultado correspondiente al Modelo del Tutor.
- F. El Modelo del Tutor envía datos al Modelo del Estudiante, para establecer su avance actual con respecto al dominio, su estado emocional y el nivel de dificultad en el que se está manejando respecto a los ejercicios.
- G. El Modelo Tutor finaliza la construcción del siguiente ejercicio y lo transforma a un formato que el intérprete entienda. Este se envía a la capa del servidor que se encarga de hacer llegar el ejercicio a los distintos clientes.

En el paso B se necesita cargar un archivo FCL (Fuzzy Control Language) [10] que contiene toda la información sobre el sistema lógico difuso. Este es un archivo de configuración y en él se encuentra definido 4 variables de entrada, 1 variable de salida y 74 reglas difusas. El método de defusificación utilizado es centro de gravedad y la evaluación de reglas es por restricción de tipo AND.

2.4. Capa de administración de modelos

Contiene los componentes que estén relacionados con el *Modelo del Tutor: Modelo del Dominio y del Estudiante*. Estos componentes manejan la administración de sus respectivos modelos y responden a las solicitudes del *Modelo del Tutor*.

- **Modelo del Dominio (Conocimiento):** contiene el conocimiento que el estudiante aspira a aprender y practicar. Este componente tiene la capacidad de ejecutar las consultas que el *Modelo del Tutor* solicite. El dominio de matemáticas de segundo año de primaria se encuentra en un archivo con formato XML. La construcción del *Modelo del Dominio* es a menudo el primer paso para representar el conocimiento del estudiante, los modelos de dominio son representaciones cuantitativas del conocimiento del experto en un dominio específico [1], por lo tanto se busca que la representación del conocimiento del dominio coincida con la estructura mostrada en los libros de segundo año de primaria de la SEP, seleccionando la teoría del espacio del conocimiento [11] para su representación.

La teoría de los espacios de conocimiento utiliza conceptos de teoría combinatoria y probabilidad para modelar y describir empíricamente ámbitos particulares del conocimiento. En términos formales la teoría nos dice que una estructura de conocimiento es un par (Q, K) en el que Q es un conjunto no vacío, y K es una familia de subconjuntos de Q , que contiene, al menos, una Q y el conjunto vacío \emptyset .

El conjunto Q es llamado el dominio de la estructura del conocimiento. Sus elementos se conocen como preguntas o ítems y los subconjuntos en la familia K están etiquetados (conocimientos) como estados. Ocasionalmente se puede decir que K es una estructura del conocimiento sobre un conjunto Q en el sentido de que (Q, K) es una estructura de conocimiento. La especificación del dominio se puede negar sin ambigüedad ya que tenemos $\cup K = Q$. Por ejemplo, para representar el modelo del conocimiento:

Conocimiento = $\{\emptyset, \{\text{Bloque 1, Lección 1.1, Lección 1.2}\}, \{\text{Bloque 2, Lección 2.1}\}\}$

- **Modelo del Estudiante:** recibe las peticiones de consulta del *Modelo del Tutor*, y las transforma de acuerdo a la plataforma tecnológica y envía de regreso la información. La representación del conocimiento del estudiante se realiza por medio una red semántica. Múltiples STI (ActiveMath, Wayang Outpost, Guru Tutor) utilizan la red semántica como representación del conocimiento [1]. La ventaja de utilizar una red semántica como *Modelo del Estudiante* es que estas son sencillas de representar e implementar tecnológicamente. La categoría usada fue una Red IS-A [12].

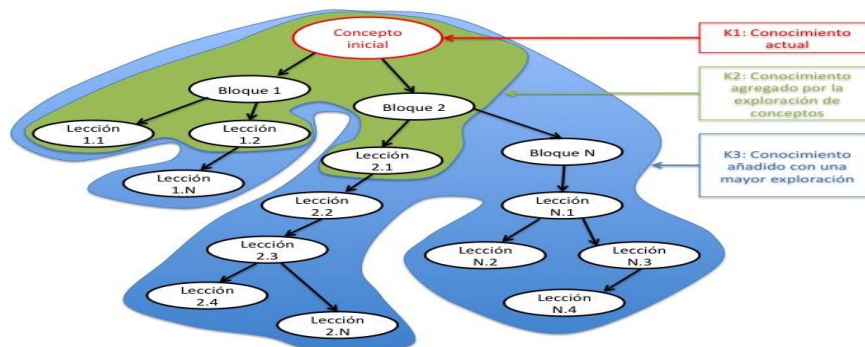


Fig. 4. Diseño de la representación del conocimiento del curso de matemáticas de segundo grado.



Fig. 5. Representación del conocimiento del estudiante en una red semántica

2.5 Capa de acceso a datos

Contiene los componentes que manejan el acceso a fuentes de datos externas como archivos o base de datos. A continuación se detallan los componentes que la conforman:

- Módulo de Acceso a la Base de Datos: el sistema tiene una base de datos relacional que contiene la representación del *Modelo del Estudiante* a nivel físico, mientras el *Modelo del Estudiante* administra las peticiones del *Modelo del Tutor*, el *Módulo de Acceso a la Base de Datos* convertirá las peticiones del modelo del estudiante a peticiones que se ejecuten en el manejador de la base de datos.

3 Resultados, evaluación y discusión

Actualmente se ha probado el sistema en forma interna por nuestro grupo de investigación. Los resultados obtenidos muestran que la primera versión del sistema cumple con los objetivos que se establecieron al principio del proyecto. En la figura 6 se muestra una sesión de pruebas del sistema.



Fig. 6. Sesión de prueba del sistema de aprendizaje.

Los objetivos del sistema de aprendizaje se diseñaron en base a las características recomendadas para la completitud de un STI [1]. En la tabla 1 se muestra una tabla de características deseadas para un sistema tutor inteligente y se evalúa el avance en cada una de ellas.

Tabla 1. Características deseadas en un STI.

Característica	Descripción	Avance en STI
Generalidad	La capacidad para generar problemas apropiados, consejos, y la ayuda personalizada para las necesidades de aprendizaje de los estudiantes	Los niveles de los ejercicios se seleccionan de acuerdo a la habilidad del estudiante, además las ayudas dadas son tomadas en cuenta de acuerdo al estado emocional del estudiante.

Modelo del Estudiante	La habilidad para representar y razonar acerca de conocimiento actual del estudiante y sus necesidades de aprendizaje	Se construye un Modelo del Estudiante a partir de la resolución de ejercicios, estos modelos se utilizará para que el tutor sugiera los tipos de ejercicios que el estudiante deberá practicar mas.
Modelo Experto	Una representación y la manera de razonar sobre el desempeño experto en el dominio y la capacidad implícita de responder ofreciendo Instrucción	El tutor tiene la capacidad de modificar el dominio y este sea reflejado automáticamente, se planea complementarlo un modulo el cual el experto modifique el dominio en línea.
Iniciativa Mixta	La posibilidad de iniciar interacciones con un estudiante, así como para interpretar y responder de manera útil a las interacciones iniciadas por los alumnos	El tutor no tiene la habilidad de iniciar una interacción con el estudiante, solo de comunicarse a través del agente pedagógico, se planea agregar un módulo de procesamiento de lenguaje natural, para que el tutor tenga la capacidad de interactuar con el estudiante.
Aprendizaje Interactivo	Actividades que requieren un compromiso auténtico aprendizaje de los estudiantes y se contextualiza y dominios adecuado apropiadamente	Esta característica se esta trabajando para presentar actividades de mayor interacción hacia el usuario, se planea el uso de API de sitios externos como por ejemplo YouTube (Sistema de administración de videos), como mensajes de diálogos del avatar relacionados al contexto del dominio.
Modelo Instruccional	La capacidad de cambiar el modo de enseñanza basado en inferencias sobre el aprendizaje del estudiante	Se esta trabajando en la parte de personalizar la enseñanza, por ejemplo el alumno podrá presentar pruebas generales para poder validar el aprendizaje de bloques completos.
Auto-Mejora	La capacidad de un sistema para monitorear, evaluar y mejorar su propio desempeño en la	Esta parte no esta contemplada su realización a corto plazo en el desarrollo del STI.

	enseñanza basada en su experiencia con antiguos alumnos	
--	---	--

A continuación en la tabla 2 se evalúa una métrica de 0, 1 y 3. Donde 0 es que la característica se dejó para versiones futuras o no está contemplada. 1 que la característica no está cubierta completamente y 3 cuando la característica se cubrió en su totalidad. Las flechas verdes representan el 3. Las amarillas el 1 y las cruces rojas el 0. Se agregan otros tutores inteligentes evaluados en otros textos [1]. Para realizar una comparación con ellos.

Tabla 2. Comparación de distintos STI con el Sistema de Aprendizaje.

Característica	Sistema de Aprendizaje Inteligente y Afectivo	ActiveMath	Wayang Outpost	Animal Watch
Generalidad	3	3	0	0
Modelo del Estudiante	3	3	3	3
Modelo Experto	3	3	3	3
Iniciativa Mixta	1	1	0	0
Aprendizaje interactivo	1	3	3	3
Modelo Instruccional	1	1	3	1
Auto-Mejora	0	1	0	1
Total de la Evaluación	12	15	12	11

De acuerdo al total de evaluación, el sistema de aprendizaje inteligente y afectivo logra una puntuación de 12 puntos, lo que lo deja con la misma puntuación que Wayan Outpost y Arriba de Animal Watch.

4 Conclusiones y trabajo futuro

Actualmente se considera que las emociones juegan un papel importante dentro del proceso de aprendizaje de una persona. En los últimos años, el desarrollo de sistemas tutores inteligentes están incorporando el reconocimiento automático de emociones lo que

permite mejorar el aprendizaje a través de interacciones adaptadas a cada estudiante considerando aspectos cognitivos y afectivos.

El desarrollo del Sistema de Aprendizaje Inteligente presentado en este artículo representa un reto por su complejidad y diversidad de áreas que participan. En relación a las herramientas de software utilizados para la construcción del sistema se utilizaron: HTML 5, para la estructuración del contenido web, CSS3, para la parte de la representación visual del contenido, Java, para programar el STI del lado del servidor, JavaScript (JS), para programar la interfase del lado del cliente, jQuery, plataforma de trabajo que agiliza el uso de JavaScript, jQuery Mobile para las interfases gráficas de la web móvil, Servlets y Java Server Pages (JSP), para la creación dinámica de contenido HTML usando Java, MySQL, como manejador de la base de datos relacional, JavaScript Object Notation (JSON), eXtensible Markup Language (XML) para el intercambio de datos, XML Path Language (XPath), para poder construir expresiones que recorran y procesen archivos XML.

Referencias

1. PISA Country Profiles, de <http://pisacountry.acer.edu.au/>, [Recuperado el 11 de Julio de 2014].
2. Secretaría de Educación Pública. Enlace boletín informativo. [En línea]. Obtenido de http://enlace.sep.gob.mx/content/gr/docs/2013/historico/00_EB_2013.pdf. [Citado el: 11 de agosto 2014].
3. Díaz Velarde María Eugenia, Villegas Quezada C.: Las matemáticas y el dominio afectivo. *Revista Multidisciplina, Matemáticas e Ingeniería*, Num 16, Sep-Dic, pp. 139-164, (2013)
4. Arroyo, I., Woolf, B., Cooper, D., et al.: Emotions sensors go to school. En *Proceedings 14th International Conference on Artificial Intelligence in Education*, pp. 17-24, (2009)
5. D’Mello, S.K., Picard, R.W. y Graesser, A. C.: Towards an affective-sensitive AutoTutor. *Special issue on Intelligent Educational Systems IEEE Intelligent Systems*. Vol 22, No. 4, pp 53-61, (2007)
6. Conati C. Y., Maclaren H.: Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User Adapted Interaction*. Vol. 19, No. 3, pp. 267-303, (2009)
7. Woolf, B.P.: *Building intelligent interactive tutors: Student-centered strategies for revolutionizing e-learning*. USA: Morgan Kauffman Publishers/Elsevier, (2009).
8. FuzzyLogic, <http://jfuzzylogic.sourceforge.net/html/index.html> , [último acceso 30 junio, 2014]
9. Gary B., Kaehler. A.: *Learning computer Vision with OpenCV library*; Oreally, (2008)
10. Cingolani, P., and Alcalá J.: jFuzzyLogic: a Java Library to Design Fuzzy Logic Controllers According to the Standard for Fuzzy Control Programming, *International Journal of Computational Intelligence Systems*, vol. 6, no. sup1, pp. 61–75, (2013)

11. Doignon, J. -P. and Falmagne, J. C.: Knowledge Spaces. Springer-Verlag (1999)
12. Brachman, R. J.: What IS-A Is and Isn't: An analysis of Taxonomic Links in Semantic Networks, IEEE Computer, Vol. 16, No. 10: 30-36, (1983)