

Ambiente inteligente y afectivo de aprendizaje para compiladores

María Lucia Barrón-Estrada, Ramón Zatarain-Cabada, H.E. Gómez-Pérez

Instituto Tecnológico de Culiacán,
Culiacán, Sinaloa, México

{rzatarain, lbarron, hgomez}@itculiacan.edu.mx

Resumen. Este artículo presenta el diseño e implementación de un Ambiente Inteligente de Aprendizaje (AIA) capaz de considerar el perfil completo de un estudiante (incluyendo su estado emocional), para la enseñanza de un tema tan complicado como lo es compiladores. El AIA se encuentra enfocado en el método de *aprendiz* (Apprenticeship) el cual permite a los estudiantes aprender mediante la práctica, cambiando el arraigado paradigma del aprendizaje meramente teórico. Se incorporan características que permiten al AIA elegir el tipo de ejercicio que mejor convenga al estudiante, comenzando por los más sencillos e incrementando la dificultad, haciendo uso de clasificación con lógica difusa y reconocimiento de emociones.

Palabras clave: computación afectiva, sistemas tutores inteligentes, redes neuronales, sistemas difusos, aprendizaje máquina, ambientes inteligentes.

1. Introducción

Un ambiente inteligente de aprendizaje (AIA) es un nuevo tipo de sistema educacional, el cuál combina lo mejor de los sistemas tutores inteligentes (STI) y los ambientes de aprendizaje. Se ha comprobado la utilidad de los AIA para la enseñanza meramente práctica, poniendo a prueba a los estudiantes en la resolución de problemas reales y desafiantes.

En los últimos años los AIA han incorporado la habilidad para reconocer el estado afectivo de los estudiantes y, con ello, ser capaces de cambiar la manera de interactuar [1-5]. La detección de emociones en los usuarios en muchos trabajos se realiza utilizando sensores especiales como sillas de postura o brazaletes [6]. Estos dispositivos resultan intrusivos y, a veces, molestos para los usuarios. Tratando de evitar esta incomodidad al usuario, se han realizado trabajos en donde incluye la detección de emociones dentro del tutor [7].

En el presente trabajo se busca integrar en el AIA diferentes aspectos como son el estado emocional, nivel de conocimientos en compiladores, además de la metodología del *aprendiz* (apprenticeship) para la enseñanza [8]. Esto se logra mediante el componente neurodifusor, que trabaja con una API externa fuera del entorno del AIA, para el reconocimiento de emociones, además utiliza reglas y conjuntos difusos para

detectar el nivel del estudiante en base a la resolución de los ejercicios. La interfaz ingresa al estudiante directamente a la práctica y de acuerdo al nivel detectado propone la complejidad de los ejercicios, simulando así un experto con su discípulo.

Un AIA es tan inteligente como el conocimiento introducido por sus expertos, por lo que al implementar una plataforma que sea fácilmente extensible por múltiples especialistas, obtendrá experiencia que le permitirá igualar e incluso superar a cualquier otro AIA. Es por ello que mostramos resultados haciendo comparación con diversos tutores [9-11] que son considerados los mejores en sus respectivas áreas.

El artículo está dividido de la siguiente manera: la sección 2, *Dominio del AIA* explica los distintos objetivos y habilidades que el estudiante debe adquirir al utilizar el sistema. La sección 3, *Interfaz del AIA* describe las distintas áreas que conforman la interfaz y como cada una cumple su propósito. La sección 4, *Arquitectura del AIA* muestra la estructura y relación de los componentes más importantes, haciendo énfasis en los componentes que manejan la parte de inteligencia artificial y afecto. La sección 5, *Trabajo relacionado* presenta la descripción de otros trabajos y los diversos aspectos que fueron seleccionados para realizar una comparación entre el sistema propuesto y los otros existentes. Por último las *Conclusiones* expresan los resultados obtenidos y el trabajo que se planea para el futuro.

2. Dominio del ambiente inteligente de aprendizaje

Compiladores es un microcosmo del área de ciencias de la computación, ya que contiene problemas de todos los tipos incluyendo entre otros (pero no limitado a) inteligencia artificial (búsquedas heurísticas), algoritmos, lenguajes formales, autómatas y arquitectura [12].

El ambiente inteligente de aprendizaje busca cumplir objetivos específicos, limitando el amplio panorama teórico y enfocándose en las habilidades prácticas que requiere el aprendiz. Para esto, el AIA abarca diferentes temas que el estudiante debe cubrir en una o más sesiones, alcanzando los objetivos siguientes:

- Comprende la notación Backus-Naur (BNF).
- Crea un lenguaje en BNF y puede escribir en él.
- Relaciona un conjunto de símbolos y los agrupa (tokens).
- Genera una interfaz de uso común para la clasificación de tokens.
- Comprende la función de un scanner.
- Crea un scanner en java utilizando la interfaz de clasificación de tokens.
- Comprende la función de un parser.
- Genera un parser utilizando el scanner previo.

Todos estos objetivos son enseñados con las mejores prácticas, de esta manera el estudiante desarrollará un compilador básico y adquirirá las habilidades prácticas que se requieren en la materia.

3. Interfaz del ambiente inteligente de aprendizaje

El AIA cuenta con una interfaz diseñada responsivamente, la cual se adapta al tamaño de pantalla del dispositivo en el que se esté utilizando. En la Fig. 1 se puede observar el área principal del AIA, donde el estudiante aprende mediante la práctica.

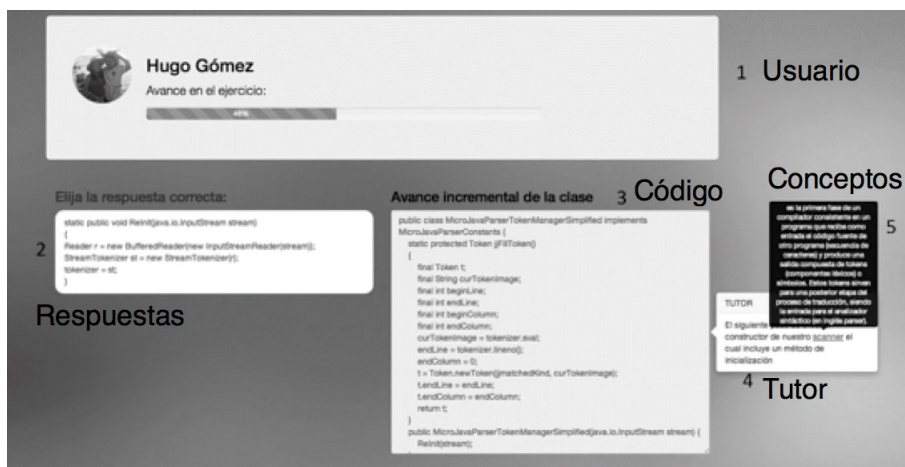


Fig. 1. Interfaz en el área de ejercicios del AIA.

La interfaz se encuentra organizada en cinco secciones delimitadas por áreas físicas, en donde cada una tiene un objetivo específico. Las áreas son las siguientes:

1. **Usuario.** Presenta los datos del estudiante, mostrando su foto de perfil de Facebook y el avance del ejercicio actual.
2. **Respuestas.** El área de respuestas es donde influye la detección inteligente del ambiente, ya que dependiendo de la clasificación del estudiante, estas cambian de complejidad. Existen tres niveles de complejidad para las respuestas solicitadas. El primer nivel muestra una parte del código del compilador que el estudiante tiene que teclear tal cual. El segundo nivel muestra una variedad de respuestas, de las cuales el estudiante tiene que elegir la mejor. La tercera y más desafiante, muestra código incompleto, el cual tiene que ser completado por el estudiante usando sus habilidades de intuición, así como conocimientos previos en la materia.
3. **Código.** Aquí se muestra el avance del código que el estudiante ha completado, de esta manera puede analizar lo que ha realizado y darse una mejor idea de cómo los pequeños avances poco a poco toman forma, para terminar una parte del compilador.

4. **Tutor.** Esta zona muestra el conocimiento del experto, donde explica que es lo que está pasando conforme se avanza bloque a bloque de código, de manera que el estudiante vea cuáles son las mejores prácticas y alternativas con las que se pueden realizar los mismos objetivos.
5. **Conceptos.** Además de la explicación del experto, las palabras clave se encuentran resaltadas para que el estudiante pueda ver sus definiciones sin necesidad de salir del AIA.

4. Arquitectura del ambiente inteligente de aprendizaje

El AIA cuenta con una arquitectura de capas relajadas que se puede visualizar en la Fig. 2. A continuación se detallan los componentes más relevantes de cada capa.

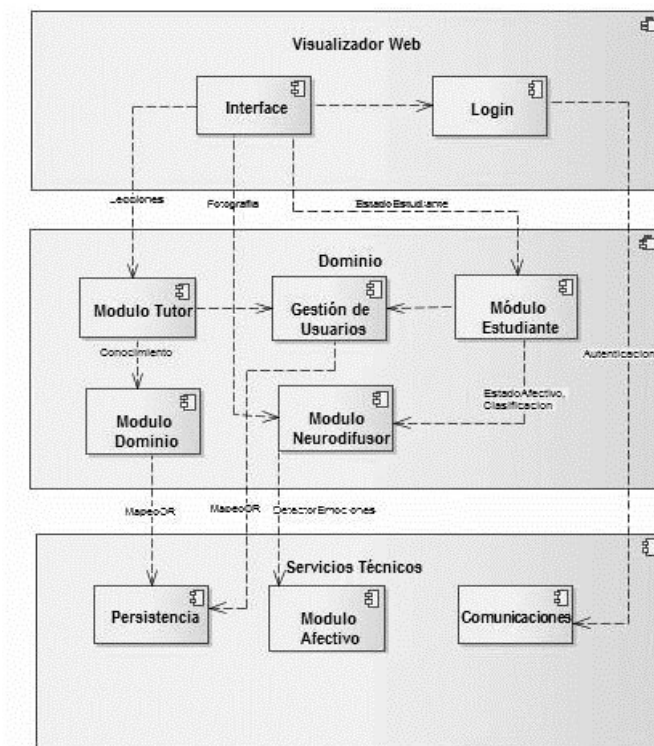


Fig. 2. Arquitectura del ambiente inteligente de aprendizaje.

4.1. Capa de visualizador Web

Es utilizada para la presentación del AIA y es con lo que interactúa el usuario. Esta cuenta con dos componentes llamados Interface y Login.

4.1.1. Interface

Se encarga de presentar la interfaz al usuario y está hecha mediante HTML5, utilizando el framework Bootstrap de twitter para conseguir un diseño responsivo capaz de adaptarse a cualquier tamaño de visualización.

4.1.2. Login

Provee la comunicación para ingresar al tutor por medio de Facebook actualmente, pero se planea acrecentarlo a google+ y twitter.

4.2. Capa de dominio

La capa de dominio implementa todo lo relacionado con el conocimiento del AIA, inteligencia artificial y pase de la información hacia el almacenamiento, todo en esta capa está implementado en PHP, mediante el framework Laravel y consta de cinco componentes que se describen a continuación.

4.2.1. Módulo tutor

Codifica los métodos de enseñanza que sean adecuados para el área de estudio y el alumno. Basado en los datos de las debilidades y fortalezas, nivel de conocimiento y emoción, el AIA selecciona el modelo de intervención de enseñanza más apropiado. Por ejemplo, si un estudiante ha sido evaluado como principiante en un determinado ejercicio, el módulo muestra algunos ejemplos paso a paso del procedimiento antes de invitar al usuario a realizarlo el mismo. También puede proporcionar retroalimentación, explicaciones y el entrenamiento a medida que el participante realiza el proceso. Conforme el estudiante gana experiencia, el módulo puede decidir presentar los escenarios cada vez más complejos. También puede decidir tomar el asiento de atrás y dejar que la persona explore libremente los ejercicios, intervenir con las explicaciones y el entrenamiento sólo bajo petición. Además, el módulo se encarga de poder elegir temas y ejemplos que subsanen las deficiencias de competencia del usuario.

4.2.2. Módulo dominio

Contiene una representación computacional de un experto en el área de Compiladores. Este conocimiento permite al AIA comparar las acciones del estudiante y las selecciones con las de un experto a fin de evaluar lo que el usuario conoce y lo que necesita aprender. Por ejemplo, se capta el conocimiento del estudiante a base de reglas que permite que el AIA genere problemas sobre la marcha.

Además permite que el desarrollador del curso pueda crear plantillas, que especifican una secuencia permitida de acciones correctas. Este método evita la codificación de la capacidad de resolver todos los problemas posibles. En cambio, sólo requiere la capacidad de especificar cómo el alumno deberá responder en cada

escenario. Estas plantillas se encuentran implementadas en formato JSON para su fácil manejo.

4.2.3. Módulo gestor de usuarios

Es el componente que accede la capa de servicios para que la persistencia de la información sea totalmente encapsulada y el acceso se vea comprensible y simple.

4.2.4. Módulo estudiante

Evalúa el desempeño de cada alumno para determinar sus conocimientos, habilidades perceptivas y habilidades de razonamiento. Al mantener y hacer referencia a un módulo con el detalle de las fortalezas y las debilidades de los usuarios, el AIA puede ofrecer muy específicamente, la instrucción pertinente.

4.2.5. Módulo neurodifusor

Contiene a las clases necesarias para la lógica difusa y la conexión hacia el componente afectivo, que proporcionan la información necesaria para que el AIA tome las decisiones correspondientes en base a estas técnicas de inteligencia artificial.

Lógica difusa

El STI utiliza la teoría de conjuntos difusos y reglas difusas para calibrar lo vago o ambiguo en las 4 variables difusas que maneja, como son tiempo, emoción, nivel en el ejercicio, como variables de entrada y la cuarta es la variable de salida, llamada nivel de dificultad, esta será tomada en cuenta para el próximo ejercicio que se le presentará al estudiante.

Un conjunto difuso puede ser definido simplemente como un conjunto con límites difusos [13].

Visto de otra manera, en la teoría difusa, un conjunto difuso A de universo X esta definido por la función $\mu_A(x)$ llamada función de membresía del conjunto A la cual es representada como: $\mu_A(x): X \rightarrow [0, 1]$ donde:

- $\mu_A(x) = 1$ si x esta totalmente en A ;
- $\mu_A(x) = 0$ si x no esta en A ;
- $0 < \mu_A(x) < 1$ si x esta parcialmente en A .

Una regla difusa puede ser definida como un enunciado condicional con variables lingüísticas y valores lingüísticos determinados por los conjuntos difusos en el universo del discurso.

Algunas reglas difusas que se manejan en el AIA son las siguientes:

1. **If** (nivelEnEjercicio is bajo) **and** (tiempoResolviendo is lento) **and** (estadoAfectivo is enganchado) **then** (dificultadEjercicio is baja)
2. **If** (nivelEnEjercicio is regular) **and** (tiempoResolviendo is rapido) **and** (estadoAfectivo is distraido) **then** (dificultadEjercicio is alta)
3. **If** (nivelEnEjercicio is regular) **and** (tiempoResolviendo is lento) **and** (estadoAfectivo is neutral) **then** (dificultadEjercicio is baja)
4. **If** (nivelEnEjercicio is regular) **and** (tiempoResolviendo is regular) **and** (estadoAfectivo is enganchado) **then** (dificultadEjercicio is alta)

5. **If** (nivelEnEjercicio is bueno) **and** (tiempoResolviendo is rapido) **and** (estadoAfectivo is enganchado)
then (dificultadEjercicio is alta)

4.3. Capa servicios técnicos

Contiene todos los servicios de bajo nivel, como lo son base de datos y comunicación entre sistemas y componentes de detección de emociones. Esta capa cuenta con tres componentes que se describen a continuación.

4.3.1. Módulo afectivo

Se encarga de procesar las imágenes y devolver la emoción que se detectó hacia la capa superior, esto lo hace mediante una API externa de Sky Biometry la cual utiliza algoritmos de aprendizaje máquina y una inmensa base de datos con millones de caras de alrededor del mundo, para identificar las coordenadas de la boca, ojos y nariz y así determinar la emoción actual [14].

4.3.2. Persistencia

Maneja todo lo relacionado con base de datos, debido a que en ninguna otra parte del AIA existen accesos para guardar la información. Este componente provee las interfaces necesarias para el guardado y al utilizar tecnología ORM mediante el framework laravel este puede guardar para distintos proveedores.

4.3.3. Comunicaciones

Se utiliza para la interacción con otros sistemas, como lo es el log in para Facebook. Además es el componente que provee la conexión hacia el compilador en un servidor apache tomcat, permitiendo realizar los ejercicios de comprobación del lenguaje BNF.

5. Trabajo relacionado

Se decidió realizar una evaluación comparativa con otros sistemas que tienen una finalidad similar [9-11], la cual se puede observar en la Tabla 1. En esta se listan diversas características de evaluación propuestas por Woolf [5] y se muestra una escala de Likert con valores de 5 a 1 representando valores desde Totalmente de acuerdo hasta totalmente en desacuerdo respectivamente.

Los sistemas con los que se hace la comparación son ambientes inteligentes de aprendizaje diseñados para otro dominio (programación de computadoras) debido a la falta de sistemas de aprendizaje en el área de compiladores.

El ambiente inteligente para el aprendizaje de introducción a la programación (Intelligent Environment for Learning Introductory Programming (ITEM/IP)) [11]

soporta un curso de introducción a la programación basado en el mini lenguaje turingal, como base para dominar los conceptos principales de la programación.

NoobLab [9] trata de replicar el dinamismo que existe entre el tutor y el estudiante dentro de los talleres de trabajo para programación, en un ambiente totalmente en línea.

Web-Based Intelligent Tutoring System for Computer Programming (WITS) [10] se encuentra dirigido por un proceso de decisiones realizado a partir de redes bayesianas, las cuales permiten recomendar metas de aprendizaje y generar secuencias de lectura.

Tabla 1. Evaluación comparativa del AIA.

Característica del Tutor Inteligente	Descripción de la característica	ITEM/ITEM P	NoobLab	WITS	AI A
Generatividad	Capacidad de generar problemas apropiados, pistas y ayuda necesaria, adaptados a las necesidades de aprendizaje del estudiante.	3	5	4	5
Modelo del estudiante	Capacidad de representar y razonar acerca del conocimiento del estudiante y las necesidades de aprendizaje en respuesta a la proporción de instrucciones.	3	5	4	5
Modelo del experto	Representación y ruta de razonamiento acerca del rendimiento dentro del dominio y la capacidad implícita de respuesta por instrucción proporcionada.	5	5	3	5
Iniciativa Mixta	Capacidad de iniciar la interacción con el estudiante, así como la interpretación y respuesta de manera útil a interacciones iniciadas por los estudiantes.	5	3	4	4
Aprendizaje interactivo	Actividades de aprendizaje que requieren compromiso y autenticidad contextualizados en el dominio apropiado.	5	4	4	3
Modelo instruccional	Capacidad de cambiar el modo de enseñanza basado en inferencias sobre el aprendizaje del alumno.	4	4	4	4
Auto-mejora	Capacidad del sistema para monitorear, evaluar y mejorar su propia enseñanza basado en su experiencia con otros alumnos.	3	3	3	2

6. Conclusiones

Durante el desarrollo se buscó implementar todas las características que requiere un buen AIA, estas fueron: reconocimiento afectivo, reacción inteligente hacia las necesidades cognitivas del estudiante y una metodología de enseñanza eficiente.

Distintas metodologías fueron evaluadas, pero resultó triunfante la del *aprendiz* (apprenticeship) ya que promueve el uso de la práctica e interacción continua con el AIA. Un desafío muy grande fue la implementación de la interfaz para distintos dispositivos, que conservara la filosofía del *aprendiz*, esto fue resuelto mediante un diseño responsivo web, que permite el acceso a cualquier persona de todo el mundo.

Otra de las aportaciones valiosas en el proyecto fue la implementación del acceso a través de las redes sociales, las cuales son utilizadas día con día por un gran número de estudiantes, permitiendo esto, abrir un espacio para la participación del AIA. Además de que genera un gran impacto hacia estudiantes y profesores que requieren más ayuda para un tema tan complicado como lo es compiladores.

Hasta este momento, las pruebas realizadas al sistema permiten visualizar que el uso de un ambiente inteligente de aprendizaje para compiladores ha permitido que los estudiantes logren entender en un menor tiempo los temas referentes a análisis lexicográfico y análisis sintáctico en compiladores, al realizar de manera práctica algunos ejercicios desarrollando el código necesario para implementar la funcionalidad.

En trabajos futuros se pretende aplicar un mejor algoritmo de clasificación con redes bayesianas, agregar otras maneras de reconocer el estado cognitivo del estudiante y extender las lecciones para que el AIA realice compilación en tiempo de ejecución.

Referencias

1. Alevan, V., Koedinger, K.: An effective metacognitive strategy: learning by doing and explaining with a computer-based cognitive tutor. *Cognitive Science*, 26(2), pp. 147–179 (2002)
2. Woolf, B., Burleson, W., Arroyo, I., Dragon, T., Cooper, D., Picard, R.: Affect-aware tutors: recognising and responding to student affect. *Int. J. Learn. Technol.* 4, pp. 129–164 (2009)
3. McQuiggan, S.W., Lester, J.C.: Diagnosing self-efficacy in intelligent tutoring systems: an empirical study. In: *Proceedings of the 8th international conference on Intelligent Tutoring Systems*, pp. 565–574, Springer-Verlag, Jhongli, Taiwan (2006)
4. Im Jraidi, N., Chalfoun, P., Frasson, C.: Implicit strategies for intelligent tutoring systems. In: *Proceedings of the 11th international conference on Intelligent Tutoring Systems*, pp. 1–10, Springer-Verlag, Chania, Crete, Greece (2012)
5. Woolf, B. P.: *Building intelligent interactive tutors*. Morgan Kaufmann (2009)
6. Arroyo, I., Woolf, B., Cooper, D., Burleson, W., Muldner, K., Christopherson, R.: Emotions sensors go to school. In: *Proceedings of the 14th international conference on artificial intelligence in education (AIED 2009)*, Brighton, UK, pp. 17–24, IOS press, Amsterdam (2009)
7. Zatarain-Cabada, R., Barrón-Estrada, M.L., Beltrán, J.A., Cibrian, F.L., Reyes-García, C., Hernández, Y.: Fermat: merging affective tutoring systems with learning social networks. In: *Proceedings of the 12th IEEE international conference on advanced learning technologies*, Rome, Italy, pp. 337–339, IEEE Computer Society (2012)
8. Patrick, A., Rainbird, H.: *Apprenticeship: Towards a New Paradigm of Learning*. Taylor & Francis (2014)
9. Neve, P.; Hunter, G.; Livingston, D.; Orwell, J.: NoobLab: An Intelligent Learning Environment for Teaching Programming. In: *Web Intelligence and Intelligent Agent*

- Technology (WI-IAT), IEEE/WIC/ACM International Conferences on, vol. 3, pp. 357–361 (2012)
10. Butz, C. J., et al.: A Web-Based Intelligent Tutoring System for Computer Programming. Web Intelligence, 2004. In: WI 2004, Proceedings of IEEE/WIC/ACM International Conference on (2004)
 11. Brusilovsky, P.: Towards an Intelligent Environment for Learning Introductory inProgramming. In: Cognitive Models and Intelligent Environments for Learning Programming. E. Lemut, B. du Boulay and G. Dettori, Springer Berlin Heidelberg. vol. 111, pp. 114–124 (1993)
 12. Appel, A. W.: Modern Compiler Implementation in Java. Cambridge. MA: Cambridge University Press (2002)
 13. Negnevitsky, M.: Artificial Intelligence a guide to intelligent systems. Pearson Education (2005)
 14. Sky Biometry: Cloud-based Face Detection and Recognition API. <https://www.skybiometry.com>. Accessed 15 Nov (2014)