

Sistema tutor inteligente para reducir el índice de reprobación en estudiantes de ingeniería informática

Edgar Degante Aguilar

Instituto Tecnológico Superior de Teziutlán

Maestría en Sistemas Computacionales

Aire Libre Fracción I y II, Teziutlán, Puebla, C.P 73960.

egdar.da@teziutlan.tecnm.mx

Resumen

Se identificó un área de oportunidad para el diseño, desarrollo y futura implementación de un Sistema Tutor Inteligente en la carrera de Ingeniería Informática, específicamente para contribuir en la reducción del índice de reprobación estudiantil de las asignaturas: Fundamentos de Programación, Programación Orientada a Objetos y Estructura de Datos. Para la recolección de datos se utilizaron encuestas enfocadas a aspectos esenciales de estudio y estilos de aprendizaje con la finalidad de elegir el algoritmo de clasificación más adecuado para el tratamiento de los datos generados por el estudiante en el proceso de evaluación al término de cada proceso evaluativo. El diseño del Sistema Tutor Inteligente contempla los tres módulos principales: Tutor Inteligente, Modelo de dominio y Estudiante además de tener al docente como un usuario con posibilidad de monitorear el progreso académico de cada educando. Se utiliza la taxonomía de Bloom para realizar el proceso de evaluación correspondiente a cada proceso afrontado por el estudiante dentro del sistema. Se usará Python como lenguaje de desarrollo y base para el procesamiento de datos con Machine Learning debido a su capacidad de tomar decisiones basadas en redes neuronales y probabilidad. Se pretende realizar un cambio de paradigma del estudiante en el proceso de enseñanza-aprendizaje basado en el Sistema Tutor Inteligente con enfoque en lógica de programación.

Palabras clave: Sistema Tutor Inteligente, STI, Reprobación, Informática, Python

1. Introducción

1.1 Inteligencia Artificial en la educación

Una de las causas de deserción del estudiante de nivel superior es la reprobación y ésta se da principalmente en los primeros semestres de la carrera. Guadalupe Cu Balan (2005), expone las causas probables de deserción, reprobación y bajo rendimiento escolar de los alumnos en los dos primeros semestres de la carrera; falta de orientación vocacional, falta de motivación de los educandos y bajos conocimientos adquiridos en el nivel medio superior.

El alto índice de reprobación estudiantil de la carrera en Ingeniería Informática se debe, en gran parte, a la deserción escolar que sufre en los primeros semestres de la formación profesional. Es en los primeros cuatro semestres de la carrera que los estudiantes afianzan de manera correcta los aprendizajes construidos en programación orientada a objetos y el logro de las competencias se convierte en un aspecto importante en la formación del educando.

La deserción, en un 70%, es ocasionada por un fenómeno activo y latente denominado reprobación y suele darse en la mayor parte de las asignaturas, sin embargo, las materias de programación no aprobadas constituyen una de las causas de desánimo del estudiante, reprobación de asignaturas consecuentes, atraso en continuidad normal de formación profesional y, por último, la deserción.

Los STI son ambientes computacionales que están diseñados para impartir instrucción y apoyar inteligentemente los procesos de enseñanza y de aprendizaje mediante la interacción con el alumno (Vicari, 1990; Ovalle y Jiménez, 2004). También reciben los nombres de Materiales Educativos Computarizados Inteligentes (Rueda, 1992) o Instrucción Inteligente Asistida por Computador (Carbonell, 1970; Andriessen y Sandberg, 1999; Silveira, 2001). Se les da el calificativo de "inteligentes" para contrastarlos con los sistemas tradicionales de instrucción asistida por computador siendo la diferencia el uso de técnicas de la AI (Larkiny Chabay, 1992).

Estos sistemas representan herramientas pedagógicas más avanzadas y suministran más experiencias de aprendizaje individualizado, lo que permite que los procesos de enseñanza y de aprendizaje sean más adaptables a las necesidades específicas o al nivel de aprendizaje de cada alumno; esto se logra por medio de la construcción y análisis que reflejan los aspectos cognoscitivos y de comportamiento. Las recolecciones de datos proporcionados por los participantes en las encuestas aplicadas sirven como base para el análisis de las características propias de cada estudiante.

Las implementaciones de los STI generan nuevas expectativas de aprendizaje en los estudiantes por medio de un plan instruccional basado en las necesidades del alumno, las unidades básicas de aprendizaje y los objetivos instruccionales. Este plan se diseña para identificar y definir los métodos que ayudarán al estudiante a adquirir el conocimiento. Es constantemente replanificado de acuerdo con los hallazgos encontrados en el alumno.

El desarrollo del Sistema Tutor Inteligente propone una nueva forma de acompañar el proceso de enseñanza-aprendizaje del estudiante de Ingeniería Informática desde el uso de las tecnologías de la información, web semántica y algoritmos de tratamiento de datos. La implementación de test de identificación de estilos de aprendizaje es primordial para el seguimiento autónomo y adecuado del proceso de asignación de actividades académicas dentro del sistema.

2. Antecedentes

Los STI se han implementado en diversos entornos y el ámbito educativo es uno de los principales; partiendo de niveles educativos de nivel básico hasta el posgrado. Los STI a medida que ha transcurrido el tiempo y ha incrementado la tecnología e innovación han sufrido varios cambios o evolución.

A continuación, se da a conocer la evolución de los mismos dividiéndolos por generaciones.

- La Primera generación: Constraint Based Tutors (Tutores basados en restricciones): Se centran en el estado de la interface, en relación a la información que se muestra/oculta, gráficos, ayudas, etc.

- Segunda generación: Model-Tracing Tutors / Cognitive Tutors (Tutores basados en modelos cognitivos). Se centran en las acciones del estudiante y las reglas que generan la solución correcta a una tarea propuesta basada en un modelo mental del estado cognitivo (y de la memoria de trabajo) del alumno.
- Tercera generación: Dialogue Based Tutors (Tutores basados en diálogo-lenguaje natural): Se centran en la comunicación con el alumno a través del dialogo. Las ayudas, las explicaciones a los ejercicios, críticas y discusiones sobre un tema o problema se realizan por medio del dialogo entre el sistema y el alumno.

Entre los STI más destacados se puede citar al proyecto Azoulay-Schwartz y Hani (2006) que permite con la retroalimentación enriquecer el conocimiento del alumno. Este sistema se basa en algoritmos inteligentes para determinar el nivel actual del estudiante y poder elegir preguntas apropiadas al nivel determinado.

En Wijekumar, Meyer, y Lei (2013) se implementa un sistema tutor inteligente para mejorar la comprensión de lectura en los estudiantes que cursan el cuarto y quinto grado de básica primaria. Tutor Inteligente para Propuestas de Investigación. Se presenta un Tutor Inteligente basado en la web para proporcionar asesores a los estudiantes en la estructuración de borradores de proyectos de investigación. Proponemos un modelo del estudiante basado en una red para seguir el progreso de cada estudiante en el desarrollo del proyecto, así como una retroalimentación personalizada en cada evaluación.

Sistema de Tutoría Inteligente para la mejora de la competencia escrita del alumnado de primaria. Es un STI denominado CSRI-OL (Cognitive Self-Regulation Instruction On-Line), que precisamente aúna estas dos dimensiones instruccionales: la instrucción estratégica y autorregulada para la enseñanza de estrategias de planificación, redacción y revisión textual (ver Fidalgo & Torrance, 2018;

Fidalgo, Torrance, Rijlaarsdam, van den Bergh, & Álvarez, 2015; Fidalgo, Torrance, & García, 2008; Torrance, Fidalgo, & García, 2007) y las potencialidades del planteamiento de dicha instrucción en un entorno on-line a través de una aplicación informática (ver para una revisión MacArthur, 2016).

Sistema Entrenador Inteligente con Tecnología Multimedia. Óptima-Geometría. Se trata de un STI cuya finalidad es la de acompañar el proceso de aprendizaje de los estudiantes del nivel medio superior. Entre las bondades se puede notar la integración sistémica de la informática, las comunicaciones, la pedagogía, la psicología cognitiva y la inteligencia artificial.

La inclusión del Sistema Tutor Inteligente como apoyo en la reducción del índice de reprobación de las asignaturas de Programación en la carrera de Ingeniería Informática del Instituto Tecnológico Superior de Teziutlán representa un avance en el tratamiento y seguimiento académico de los estudiantes no regulares de los sistemas escolarizado y mixto. Se pretende fortalecer el proceso de enseñanza-aprendizaje y mejorar los resultados académicos a mediano plazo.

3. Metodología

En este apartado se describen los instrumentos utilizados para la recolección de datos.

3.1 Recolección de datos

Se aplicaron tres encuestas a cada estudiante de los 10 grupos pertenecientes a la carrera de Ingeniería Informática en los sistemas escolarizado y semi-escolarizado con la finalidad de identificar los factores dominantes en su aprovechamiento académico, el estilo de aprendizaje del grupo, organización de estudio y técnicas de estudio. La primera encuesta sirvió para determinar la organización del estudio por lo que las preguntas están enfocadas a aspectos prioritarios de organización de actividades escolares y extraescolares tales como tareas y espacio de trabajo donde se labora en actividades académicas.

La segunda encuesta contiene preguntas relacionadas a técnicas de estudio, la finalidad es identificar la mejor manera en que el estudiante puede aprender sin estresarse y lograr su objetivo. En tercer lugar, se aplicó la encuesta denominada Encuesta sobre motivación para el estudio, ésta encuesta trata acerca de aspectos motivacionales para el estudiante y que son clave para continuar con su proceso de aprendizaje dentro y fuera del aula. El medio de recolección fue a través de la plataforma Google Drive y Formularios de Google para hacer más dinámica la consulta y una más rápida colaboración de los participantes de la encuesta.

Después de realizar las encuestas y análisis de datos se identificaron las principales deficiencias en los temas de programación. A continuación, se describen las más sobresalientes:

- No siempre se logra identificar las variables de manera correcta desde la concepción y análisis del problema y/o caso de estudio.
- En ocasiones se usa métodos que no expresan claramente lo que se necesita dentro de la aplicación.
- Se presenta dificultad en el manejo de la sintaxis del lenguaje de programación de manera reiterada.
- La comprensión lectora de los estudiantes interviene enormemente en el análisis de problemas y/o casos de estudio.

3.2 Componentes del Sistema Tutor Inteligente

Componente del estudiante: debe ser capaz de reconocer las carencias y fortalezas del alumno. Puede ser visto como la unión de dos submódulos, uno que contenga la información referente a los estilos de aprendizaje y el otro que contenga los datos que van a ser actualizados, en dependencia de la interacción del alumno con el sistema, se define así el nivel de conocimiento (Tarongí, 2010).

Los estilos de aprendizaje pueden ser medidos por varios instrumentos, entre los que se destaca, el cuestionario de índice de estilos de aprendizaje. Para medir el nivel de conocimiento se utiliza la taxonomía de Bloom (1956).

Para identificar es estilo de aprendizaje de cada estudiante se implementa un Test de PNL cuyo formato está constituido por 40 preguntas con opción múltiple de 3 respuestas y al final los resultados se contrastan con una tabla concentradora para identificar el estilo de aprendizaje predominante.

Módulo del tutor: según Lage & Cataldi (2004), este define y aplica una estrategia pedagógica de enseñanza, contiene los objetivos que deben ser alcanzados y los planes utilizados para alcanzarlos. Monitorea el desempeño y selecciona el material de aprendizaje para el estudiante. Integra el conocimiento acerca del método de enseñanza, las técnicas didácticas y del dominio a enseñar. Módulo del dominio: contiene y administra el conocimiento y contenidos del área o tema específico de enseñanza. Se compone de los temas o unidades en los que se clasifica el dominio a tratar dentro del curso en que está enmarcado (Ovalle, 2007).

Scikit-learn es usado para la implementación de algoritmos de clasificación, regresión y análisis de grupos. Se trata de una biblioteca de aprendizaje automático de software libre y altamente compatible con Python 3.7 en el desarrollo del Sistema Tutor Inteligente. La viabilidad para computar datos y aplicar los métodos matemáticos hace factible su uso.

Módulo del dominio. Contiene la información o habilidad en la que el estudiante será instruido. Debe organizar el material de estudio, representado como artículos, notas de clase, videos o cualquier material que pueda ayudar a entender un concepto; también debe organizar los ejemplos prácticos que ilustran los conceptos y los casos de evaluación que para este caso se centran en preguntas de selección múltiple con única y múltiple respuesta y que presentan un enunciado, el cual puede incluir un concepto o un caso práctico y las respuestas, resaltando la correcta. El uso del Lenguaje Ontológico para la Web es usado para el marcado construido sobre RDF y codificado en XML.

De esta manera, todos los datos generados por los usuarios del Sistema Tutor Inteligente podrán ser procesados automáticamente y evaluados con los algoritmos de Machine Learning.

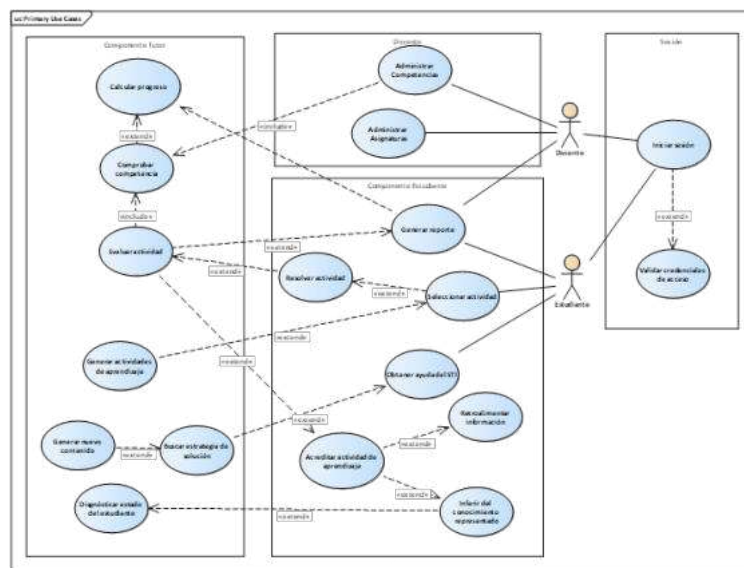


Figura 1. Diagrama de casos de uso de la arquitectura del Sistema Tutor Inteligente dividido en los módulos correspondientes.

Metodología de desarrollo

Se usará el modelo OOHDM u Object Oriented Hypermedia Design Methodology, para diseño de aplicaciones hipermedia y para la Web, fue diseñado por D. Schwabe, G. Rossi, and S. D. J. Barbosa y es una extensión de HDM con orientación a objetos, que se está convirtiendo en una de las metodologías más utilizadas. Ha sido usada para diseñar diferentes tipos de aplicaciones hipermedia como galerías interactivas, presentaciones multimedia y, sobre todo, numerosos sitios web («Modelo OOHDM», s. f.). OOHDM propone el desarrollo de aplicaciones hipermedia mediante un proceso de 5 etapas (Sánchez & Ortega Santamaría, 2001).

5. Resultados y discusión

Los factores detonantes del bajo aprovechamiento académico del alumnado son variados y van desde el estado socio-económico hasta aspectos de salud que no pueden controlarse; en la medida de la adecuación didáctica de las actividades se han realizado mejoras respondiendo a los distintos tipos de inteligencia de los estudiantes mostrados

durante el periodo de formación profesional correspondiente en las asignaturas de Fundamentos de Programación y Programación Orientada a Objetos. La inclusión del Sistema Tutor Inteligente en el desarrollo académico del estudiante propiciará un cambio de paradigma en el proceso de aprendizaje.

6. Conclusiones

La literatura argumenta que los STI implementados en la educación fungen como herramientas que ayudan a mitigar la problemática de la enseñanza: considerar que todos los estudiantes tienen las mismas necesidades de aprendizaje y la misma manera de aprender.

El desarrollo del Sistema Tutor Inteligente está en marcha y será destinado a la carrera de Ingeniería Informática del Instituto Tecnológico Superior de Teziutlán.

Nomenclatura

STI Sistema Tutor Inteligente

7. Referencias Bibliográficas

Aguilar, R. M. R., González, J. L. M. C., & Campos, A. L. L. (2013). Diseño de un sistema tutorial inteligente. *Apertura*, 5(1), 36-47.

Bustos, G. N., Roldán, P. R., & Guzmán, R. Z. (s. f.). Factores de reprobación en los alumnos del Centro Universitario de Ciencias de la Salud de la Universidad de Guadalajara. 9.

Cataldi, Z., & Lage, F. (2010). Modelado del Estudiante en Sistemas Tutores Inteligentes. *Revista Iberoamericana de Tecnología en Educación y Educación en Tecnología*, no. 5.

Competencias, calidad y educación superior—Sergio Tobón—Google Libros. (s. f.). Recuperado 12 de octubre de 2019, de https://books.google.com.mx/books?id=jW7G7qRhry4C&pg=PA114&lp g=PA114&dq=Cuarenta+preguntas+sobre+competencia+laboral,+Monte video,+Cinterfor&source=bl&ots=itNSIJ1098&sig=ACfU3U3Vt65buoLo a X o d q 2 c Q V K _ N p - H m X w & h l = e s - 419&sa=X&ved=2ahUKEwjQjK2W55XIAhVQKKwKH a L-BtQQ6AEwCXoECAoQAQ#v=onepage&q=Cuarenta%20preguntas%20 sobre%20competencia%20laboral%2C%20Montevideo%2C%20Cinterf or&f=false

Escalante, J. E., Mariño, S. I., Marchisio, S. T., & Guglielmone, M. V. G. (2018). Modelo de sistema tutor inteligente de apoyo en la realización de proyectos de investigación en posgrados en ingeniería.

Espinosa, M. L., & Valdivia, Z. G. (2008). La Inteligencia Artificial en la Informática Educativa. 5, 8.

Huapaya, C. R., Arona, G. M., & Lizarralde, F. A. (2005). Enseñanza de la Ingeniería con Sistemas Tutoriales Inteligentes. *Información tecnológica*, 16(5), 75-78. <https://doi.org/10.4067/S0718-07642005000500012>

Huapaya, Constanza Raquel, & Lanzarini, L. L. (s. f.). *Sistemas Tutoriales Inteligentes*. 74.

Hurtatiz, Y., Pascuas, Y., & Millán Rojas, E. (2015). SISTEMAS TUTORES INTELIGENTES COMO APOYO EN EL PROCESO DE APRENDIZAJE. *REDES DE INGENIERÍA*, 6, 25. <https://doi.org/10.14483/udistrital.jour.redes.2015.1.a02>

Instituto Tecnológico Superior de Teziutlán. (s. f.). Recuperado 14 de octubre de 2019, de Instituto Tecnológico Superior de Teziutlán website: <http://www.itsteziutlan.edu.mx/>

Modelo OOHDM. (s. f.). Recuperado 12 de octubre de 2019, de <http://www.hipertexto.info/documentos/ooohdm.htm>

Pytel, P., Vegega, C., Deroche, A., Acosta, M. P., & Cattaneo, M. F. P. (2015). Modelo bayesiano para el diagnóstico del aprendizaje en alumnos de inteligencia artificial.

Saavedra, L. F. Q. (2016). PROTOTIPO DE TUTOR INTELIGENTE PARA EL APRENDIZAJE DE LA PROGRAMACIÓN DE COMPUTADORES. 87.

Sánchez, J., & Ortega Santamaría, S. (2001, septiembre 26). Metodología para el Desarrollo de Contenidos en Hipermedia.

SIMPLIFY. (s. f.). Recuperado 14 de octubre de 2019, de Gradient website: <https://www.gradient.org/proyectos/simplify/>

Sowa, J. F. (s. f.). *Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project*. 10.