

Sistemas inteligentes para la enseñanza del diseño de bases de datos

Intelligent Systems for Database Design Teaching

*Jonathan Giovanni Camargo-Sanabria**

*Julio César Reina-Panche***

*Juan Sebastián González-Sanabria****

Recepción: 12 de diciembre de 2017

Aprobación: 18 de enero de 2018

Resumen

Los sistemas tutores inteligentes han tenido un gran avance en la última década, gracias al desarrollo de nuevas tecnologías y a la realización de múltiples investigaciones. De igual forma, su implementación se ha ampliado a diversas disciplinas, dentro de las que se encuentra el diseño de bases de datos. Por este motivo, el presente artículo explora, revisa y analiza las diferentes características, tecnologías, arquitecturas y proyectos llevados a cabo en el área de los ITS, para generar así una perspectiva real para el posterior desarrollo del sistema, el cual debe estar enfocado en el aprendizaje y la retroalimentación continua a través del planteamiento de ejercicios.

Palabras clave: diseño de bases de datos; modelo entidad-relación; normalización; retroalimentación continua; sistema tutor inteligente.

Abstract

Intelligent tutors systems have made a great progress in the last decade, thanks to the development of new technologies and to the multiple researches, similarly, their implementation have been extended to several disciplines like the design of

* Universidad Pedagógica y Tecnológica de Colombia (Tunja-Boyacá, Colombia). jonathangianny.camargo@uptc.edu.co.

** Universidad Pedagógica y Tecnológica de Colombia (Tunja-Boyacá, Colombia). julio.reina@uptc.edu.co.

*** M.Sc. Universidad Pedagógica y Tecnológica de Colombia (Tunja-Boyacá, Colombia). juansebastian.gonzalez@uptc.edu.co. ORCID: <https://orcid.org/0000-0002-1024-6077>.

databases. For this reason, the present paper explores, reviews and analyzes the different characteristics, technologies, architectures and projects carried out on ITS focused in this area, generating a real perspective about the future development of the system, which should be focused in the learning process and the continuous feedback through practical exercises.

Keywords: continuous feedback; databases design; entity relationship model; Intelligent Tutoring System (ITS); normalization.

Introducción

Los sistemas tutores inteligentes (ITS, por las siglas en inglés de Intelligent Tutors Systems) son aplicaciones de inteligencia artificial que brindan apoyo en el proceso de aprendizaje, a través de la especialización en un determinado tema sin dejar de lado la interacción con los usuarios. Entre las ventajas de los ITS se encuentran la capacidad de realizar una retroalimentación continua, realizar una evaluación de las soluciones dadas ante un problema y adaptarse a las características y modo de aprendizaje de cada estudiante.

Un sistema tutor inteligente es un *software* que interactúa con un usuario, en este caso un estudiante, para promover el aprendizaje a través de una o más funciones, como la elaboración de preguntas, la asignación y retroalimentación de tareas, la orientación automatizada o la evaluación continua del estudiante. Se debe tener en cuenta que para su funcionamiento este utiliza técnicas de inteligencia artificial, creando así un ambiente donde se considera el estado de conocimiento de los usuarios (Nesbit, Adesope, Liu & Ma, 2014).

Para el desarrollo de estos sistemas se emplea un modelo basado en restricciones y uno basado en agentes inteligentes, mediante los cuales un estudiante puede descubrir y corregir los errores individuales que comete, de tal forma que el *software* genera una representación de la solución, la cual es actualizada de acuerdo con las acciones tomadas por el usuario sobre el sistema tutor inteligente (Thinakaran & Ali, 2014). Cada estado se define como una restricción, pues al modificar los distintos indicadores que estos representan, se genera la realimentación adecuada.

El modelado y la normalización son procesos de gran importancia en el diseño de las bases de datos, por lo tanto, disponer de conocimientos sólidos permite a los profesionales realizar diseños consistentes. Los conceptos básicos de un modelo entidad-relación no tienen un alto grado de complejidad, sin embargo, para mejorar en el proceso de diseño se requiere adquirir experiencia empleando los conocimientos en diversas aplicaciones; por este motivo, el uso de los ITS en la enseñanza de diseño

de bases de datos genera posibilidades de mejora mediante el desarrollo continuo de ejercicios, así como nuevas oportunidades de análisis e investigación.

Para el entendimiento del trabajo se presentan inicialmente proyectos e investigaciones que se han desarrollado en el área de bases de datos, y se analizan a profundidad los enfocados en el diseño de base de datos. Posteriormente, se describen las características que debe cumplir un ITS enfocado en el diseño de bases de datos y se examinan las tecnologías clave para su desarrollo. Y en último término, se plantea una serie de sugerencias y consideraciones para tener en cuenta en la arquitectura de los sistemas tutores inteligentes.

Sistemas inteligentes en la enseñanza de base de datos

En el mundo se han desarrollado diversas investigaciones y trabajos para la elaboración de sistemas inteligentes enfocados en la enseñanza del diseño de las bases de datos, dentro de los que se destacan:

- NORMIT (Mitrovic, 2002), un ITS enfocado a la enseñanza de la normalización de una base de datos, aspecto importante a la hora de diseñarla. Este sistema se caracteriza por enfocarse en la web y en la resolución de problemas relacionados con las formas normales; ofrece a los aprendices un entorno donde se debe seleccionar el problema por tratar, seguido de una serie de pasos con los cuales se analiza la calidad en un determinado diseño.

Dentro de los conceptos que refuerza la herramienta se encuentran: la identificación de llaves primarias y de dependencias funcionales, la determinación de la forma normal en que se encuentra un modelo, la descomposición de las tablas y la identificación de atributos en un modelo dado. NORMIT ofrece una interfaz agradable, donde se puede revisar el historial de las sesiones, además brinda una retroalimentación en cada uno de los pasos, describiendo, si es el caso, los errores cometidos. La solución correcta también puede ser consultada por el estudiante.

NORMIT se compone de tres módulos: el estudiantil, que almacena los modelos que el alumno desarrolla, y analiza las soluciones dadas ante los problemas propuestos por el sistema; el pedagógico, que se encarga de decidir las acciones por seguir según las solicitudes del usuario, de generar la retroalimentación en cada ejercicio, además de determinar qué ejercicios asignar según lo registrado (conocimientos y problemas) en el módulo del estudiante; y el de problemas, el cual almacena los ejercicios que serán aplicados al estudiante. Es importante decir que el sistema incorpora el modelo basado en restricciones “Constraint-Based Modelling” (CBM) para evaluar y retroali-

mentar al estudiante en cada uno de los ejercicios. Así pues, existen restricciones que valoran la sintaxis y la semántica de la solución, comparando con la solución ideal.

- Mendjoge, Joshi y Narvekar (2016), ofrecen un sistema enfocado en dar soporte a la enseñanza de la normalización de las bases de datos. Para llevar a cabo esta labor, el *software* dispone de dos módulos: pedagógico y de autodesarrollo. El primer módulo se encarga de preparar los problemas relacionados con las formas normales y le permite al usuario seleccionar un ejercicio de una lista previamente cargada.

El segundo módulo indica los errores que se van cometiendo al empezar a plantear una solución, y le permite al usuario visualizar la solución final a través del seguimiento de ciertos pasos, dados por reglas, condiciones o restricciones previamente definidas.

Es importante resaltar que, para determinar el resultado de los problemas, el sistema tiene en consideración la dependencia funcional de los campos en la separación de atributos, creación de las tablas y de relaciones.

- KERMIT (Pramuditha & Antonija, 2004) es un ITS basado en el modelo de restricciones para asistir en el modelado de bases de datos. Para cumplir con este objetivo, se dispone de una serie de planteamientos, requerimientos y ejercicios para identificar los elementos para plasmar en el modelo entidad-relación. Posteriormente se traducen las entidades y relaciones a un formato de texto para comparar el resultado con una solución almacenada con la misma estructura de entidades y relaciones.

KERMIT cuenta con tres módulos principales: de restricciones, encargado de validar las relaciones y entidades seleccionadas y la forma en la cual se encuentran conectadas; módulo pedagógico, cuya función es determinar los ejercicios que debe realizar el estudiante de acuerdo con un nivel de dificultad asignado a cada uno; y el módulo de interfaz gráfica, que permite al usuario interactuar con el sistema. Se debe resaltar que los componentes mencionados cuentan con insumos de información tales como las restricciones, las soluciones a los ejercicios y la solución dada por el estudiante respectivamente.

- Yang (2013), plantea un *software* para la ayuda en el proceso de enseñanza de normalización de bases de datos, que consta de cinco módulos: el módulo estudiantil, de modelado de instrucciones, de conocimiento del dominio, de planificación y de interfaz de usuario. El primero de ellos se encarga de evaluar continuamente el progreso del estudiante, lo que lo convierte en la base para la generación de un plan adaptativo de enseñanza; el segundo módulo lleva a cabo la imitación de un tutor humano. Por su parte, el módulo de planificación configura una secuencia de sesiones de resolución de problemas que el estudiante tendrá que llevar a cabo. El módulo

de conocimiento del dominio implementa una selección de reglas para satisfacer la solución de los problemas y, por último, el módulo de interfaz brinda un ambiente adecuado y cómodo para el aprendizaje.

Se aclara que el sistema posee tres niveles de dificultad, cada uno de los cuales contiene tres sesiones de resolución de problemas y, dependiendo de los resultados en cada uno de ellos, el *software* evalúa si el estudiante está en facultad de aumentar de nivel o si debe ser degradado.

- Hall y Gordon (1998), indican que la elaboración del modelo entidad-relación es una tarea difícil a la que se enfrentan los estudiantes del área de la computación y los sistemas de información, pues se cometen errores como el modelar entidades como atributos o viceversa, todo lo cual es provocado por la falta de conocimiento y práctica. De igual forma, resaltan la existencia de herramientas que ayudan a la generación del modelo entidad-relación, pero no al aprendizaje y a la retroalimentación en cada uno de los ejercicios, por lo cual se hace necesario que un sistema cumpla esta función.

Así pues, se plantea un sistema que refuerza el uso de la metodología de diseño, genera una retroalimentación y proporciona ambientes de comunicación con los demás estudiantes, a través de la realización de tareas como la identificación y creación de entidades, atributos y relaciones, así como de sus asociaciones (González & González, 2013).

Por otra parte, el *software* se basa en una arquitectura cliente-servidor y en el lenguaje de programación para sistemas expertos CLIPS; brinda una interfaz compuesta de tres módulos: el primero de ellos permite al estudiante definir cada uno de los objetos y atributos, el segundo da la posibilidad de visualizar el resultado que se lleva del modelo, y el tercer módulo describe el problema por resolver.

- Yang (2011), presenta un tutor virtual para la normalización de las bases de datos, el cual tiene en cuenta cuatro módulos: 1) estudiante, donde se revisa el crecimiento del usuario tras cada lección; 2) planeación curricular, el cual selecciona los ejercicios y temáticas de acuerdo con los logros cumplidos y con la evolución del alumno; 3) de instrucción, cuya función es presentar al estudiante el material que le permite obtener una definición de los conceptos, así como ejemplos seleccionados por el dominio del conocimiento; y 4) apartado de ejercicios, que posibilita al usuario probar los conocimientos adquiridos.

El programa cuenta con una actividad de identificación para determinar los conocimientos iniciales del alumno empleando un ejercicio de normalización que requiere pasar de la primera forma normal a la tercera, y en caso de fallar, se procede a mostrar

el contenido temático. El proceso formativo busca que el usuario comprenda las dependencias funcionales, dependencias transitivas, llaves candidatas, atributos multivaluados y *joins*.

Además, el *software* cuenta con la posibilidad de solucionar ejercicios de normalización de modelos, en donde para pasar a la segunda forma normal, todos sus atributos deben ser totalmente dependientes de la llave primaria y no ser llaves. Así mismo, para pasar a la tercera forma normal, ninguno de sus atributos puede ser transitivo dependiente de la llave primaria, sin ser llaves.

En la Tabla I se comparan el enfoque, la arquitectura y la metodología de enseñanza de cada uno de los sistemas anteriormente mencionados:

Tabla 1. Comparación de los sistemas tutores inteligentes para el diseño de bases de datos

Sistema	Enfoque	Arquitectura	Metodología de enseñanza
NORMIT (Mitrovic, 2002)	Normalización de las bases de datos.	Módulo estudiantil Módulo pedagógico Módulo de problemas	Solución de problemas. Retroalimentación continua.
Intelligent tutoring system for database normalization. (Mendjoge et al., 2016)	Normalización de las bases de datos.	Módulo pedagógico. Módulo de tutor. Módulos de dominio (ejercicios y normalizador automático).	Material de lectura. Solución de ejercicios paso a paso.
KERMIT (Pramuditha & Antonija, 2004)	Diseño Conceptual (Modelo entidad relación)	Módulo de restricciones. Módulo pedagógico.	Solución de problemas a través de niveles de dificultad. Retroalimentación continua.
The dynamic curriculum planning for a database design tutoring System. (Yang, 2013)	Normalización de las bases de datos.	Módulo estudiantil. Módulo de modelado. Módulo de instrucciones. Módulo de conocimiento del dominio. Módulo de planificación. Módulo de interfaz de usuario.	Solución de problemas a través de niveles de dificultad. Retroalimentación continua.
A virtual learning environment for entity relationship modelling. (Hall & Gordon, 1998)	Diseño conceptual (Modelo entidad relación)	Módulo de problemas. Módulo de interfaz de usuario.	Solución de problemas. Retroalimentación continua.

Sistema	Enfoque	Arquitectura	Metodología de enseñanza
A virtual tutor for relational schema normalization (Yang, 2011)	Normalización de las bases de datos.	Módulo estudiantil. Módulo de modelado. Módulo de instrucciones. Módulo de conocimiento del dominio. Módulo de planificación. Módulo de interfaz de usuario.	Solución de problemas a través de niveles de dificultad. Retroalimentación continua. Material de lectura.

Características de los sistemas inteligentes para la enseñanza

Los sistemas tutores inteligentes utilizan reglas de razonamiento como la base para su funcionamiento, así pues, presentan diversas características, dentro de las que se pueden destacar (Zhang, 2005):

1. Individualización: el sistema proporciona tutorías de acuerdo con las características cognitivas de los estudiantes, así como con la evolución en el proceso de enseñanza.
2. Autonomía: el *software* puede detectar de manera dinámica el nivel de aprendizaje; tiene entonces la capacidad de ajustar el contenido de enseñanza de acuerdo con las necesidades del estudiante.
3. Inteligencia: los sistemas tutores pueden analizar las soluciones dadas por el estudiante ante un problema determinado, así como diagnosticar los errores y hacer una retroalimentación.

Por otra parte, al profundizar en estos sistemas, se puede evidenciar que cumplen con una serie de cualidades, dentro de las que se encuentran (Chu, Mitchell & Govindaraj, 1989):

- Adaptabilidad: expresada la mayoría de las veces como la velocidad con que el sistema realiza las tareas de reconocer una situación, evaluarla, buscar soluciones, seleccionar algunas de ellas y ejecutarla.
- Aprendizaje: los ITS deben ayudar a mejorar el rendimiento de los estudiantes, a través de la presentación de forma resumida y compacta de los diferentes temas.

- Representación del conocimiento: el sistema tutor inteligente debe tener un modelo completo del conocimiento que el estudiante va a adquirir; es decir, el sistema debe “conocer” de manera profunda los temas que se están tratando. De igual forma, este conocimiento debe ser representado por el *software* de forma relevante, en donde se puedan examinar e interpretar fácilmente los pasos para la solución de un problema.
- Estructura pedagógica: esta característica les permite a los sistemas definir “cómo y cuándo enseñar qué” y puede variar según el nivel de conocimiento que el estudiante haya adquirido. Igualmente, comprende la explotación de las capacidades gráficas, como visualización, animación e interactividad para hacer más fácil la conceptualización de temas.

Por otra parte, el sistema debe proporcionar escenarios para la práctica de conocimientos y habilidades e intervenir con una retroalimentación apropiada; así mismo debe suministrar herramientas potenciadoras, como la solución de un problema determinado a través de gráficos.

- Visualización de progresos: el sistema debe reflejar el progreso del aprendizaje del estudiante y contener un historial de los errores frecuentemente cometidos en la resolución de problemas planteados durante el proceso de enseñanza.

Teniendo en cuenta las diversas particularidades de los sistemas tutores inteligentes, es importante destacar que existe una clasificación que está dada según su autonomía, así pues, pueden existir tres tipos de sistemas:

1. Totalmente autónomos, que reemplazan al profesor.
2. De ayuda al profesor, manteniéndose autónomos en su uso (el estudiante puede practicar sin que el maestro esté presente).
3. De ayuda al profesor, pero no son autónomos (el maestro debe permanecer presente durante su uso).

A partir de dicha clasificación, los sistemas de tipo 2 y 3 incluyen a los docentes en su objetivo de ayudar a los estudiantes en el proceso de aprendizaje, pues en temas de alta complejidad o de falta de automatización son indispensables para el proceso de enseñanza.

Considerando este concepto, este tipo de ITS está dedicado tanto a los estudiantes como a los profesores, proporciona ayuda en la realización de tareas tediosas, realizando un seguimiento de los resultados de los estudiantes, reportando problemas

durante el transcurso del aprendizaje y brindando una retroalimentación continua junto con ejercicios personalizados. Así pues, dentro de las características por resaltar se encuentran (Yacef, 2002):

- Diagnóstico y evaluación del aprendizaje: un ITS puede proporcionar informes precisos, sistemáticos y rápidos sobre la evolución del estudiante.
- Generación de material de aprendizaje de acuerdo con los avances: un ITS puede generar preguntas o ejercicios dependiendo del nivel en que se encuentre el estudiante y las dificultades que haya tenido durante el proceso.
- Monitorización del estudiante: reporta al maestro resultados significativos desde diversos ángulos.
- Automatización de tareas complejas: trae consigo la disponibilidad del docente para tareas en las que se requiere de su presencia.
- Retroalimentación del proceso de enseñanza: genera en el maestro una mayor comprensión del modo de aprendizaje y de los aspectos en que se debe hacer énfasis.

En la Tabla II se presentan las características sugeridas para un sistema inteligente enfocado en la enseñanza del diseño de base de datos.

Tabla 2. Características del sistema inteligente para la enseñanza del diseño de bases de datos

Característica	Descripción
Módulos	Poseer al menos dos módulos, uno enfocado al docente y otro al estudiante.
Niveles de dificultad	Contar con niveles de dificultad en cada uno de los temas tratados y dependiendo de los resultados en cada uno de ellos el software evaluará si el estudiante está en la facultad de aumentar de nivel o si debe ser degradado.
Ejemplos seleccionados	Ofrecer ejemplos acerca del dominio del conocimiento.
Retroalimentación continua	A partir de la solución dada por el estudiante a un determinado problema, el sistema debe describir si es el caso, los errores cometidos.
Solución paso a paso	Debe proporcionar la solución correcta a cada uno de los ejercicios propuestos, de manera didáctica y paso a paso.
Historiales	Almacenar historiales acerca de las acciones realizadas por el estudiante y de los errores cometidos con mayor frecuencia.
Informes	Generar informes acerca del proceso de aprendizaje y evolución del estudiante.
Interfaz Intuitiva	Presentar de forma adecuada cada uno de los problemas y conceptualizaciones de los temas a tratar.

Tecnologías y teorías para el desarrollo de ITS

Dentro de las principales tecnologías y teorías que se deben tener en cuenta a la hora de realizar un sistema tutor inteligente se encuentran (Wang, Lu & Wang, 2012):

- Agentes inteligentes: los cuales son objetos capaces de apreciar su entorno, procesar las percepciones y responder o actuar de manera racional, brindando de esta forma un gran apoyo en el fortalecimiento del aprendizaje interactivo e independiente. Así pues, permiten el estudio de los niveles de dificultad que pueden ser asignados a un estudiante, de acuerdo con las características y la interacción que el usuario presente con el sistema.

Se debe tener en cuenta que estos sistemas pueden cumplir diferentes funciones dentro de un sistema tutor inteligente, como por ejemplo la interacción con el usuario, el manejo de problemas, la gestión de recursos y la evaluación del aprendizaje. De esta forma, el sistema tutor inteligente puede estar pendiente de las solicitudes de los estudiantes, deducir las necesidades de información, presentar y almacenar datos, recuperar problemas resueltos de acuerdo con el perfil del estudiante y evaluar cada uno de los pasos que el estudiante realiza durante el proceso de aprendizaje (González, Burguillo & Llamas, 2006).

- Motor de inferencia: el módulo más importante de los sistemas tutores inteligentes es el módulo del tutor, pues allí es donde se encuentra el motor de inferencia, el cual se basa en el modelo de aprendizaje de los estudiantes. Así pues, este motor es básicamente un programa de control, cuya función es seleccionar las reglas adecuadas para satisfacer un problema y alcanzar una solución o conclusión determinada, para lo cual utiliza una serie de estrategias de control sistemáticas o de estrategias heurísticas.

Para examinar las reglas acerca del conocimiento del dominio, existen dos métodos el pro razonamiento y el razonamiento inverso. El primero de ellos examina las reglas una tras otra en cierto orden. Se debe resaltar que el orden podría ser la secuencia en que las reglas se introdujeron, o podría ser alguna otra secuencia especificada por el usuario. El segundo de ellos, inicialmente selecciona una regla y la considera como un problema para resolver. Enseguida la analiza y determina que para resolverla debe primero resolver las anteriores, las cuales se constituyen en subproblemas (Li, Zhuying & Bing, 2010).

- Aprendizaje autónomo: es empleado en la mayoría de los ITS, con el objetivo principal de convertir información “proveniente de la base de datos o del sistema” a

conocimiento del dominio, lo cual se lleva a cabo empleando técnicas de búsqueda de patrones de datos y ajustando las acciones para realizar por el sistema de acuerdo con los mismos.

Es importante resaltar que el aprendizaje automático se basa en algoritmos para cumplir su función, los cuales, y básicamente se suelen dividir en dos grandes ramas, de aprendizaje supervisado y no supervisado. La primera se basa en la predicción a partir de una serie de datos, y la segunda, es decir, los algoritmos no supervisados, utilizan el análisis para descubrir patrones y tendencias.

- Procesador inteligente de lenguaje natural: la interacción que se desarrolla entre el usuario y el sistema es de gran importancia, por lo cual es posible añadir un procesamiento de expresiones semánticas que le permitan al tutor inteligente brindar ayuda de una forma más eficiente y adecuada a las dudas del estudiante, eliminando parcialmente la rigidez que presenta un *software* en el proceso de enseñanza. De esta forma, el sistema debe identificar, extraer y clasificar la información dada por el alumno, para determinar los recursos que le pueden ser de ayuda.
- Teoría del aprendizaje inteligente: consiste en enseñar a través de la entrega de herramientas que permitan a los estudiantes involucrarse e interactuar en cada uno de los contenidos, considerando que el conocimiento se asimila aplicando lo aprendido en un problema base en distintos ambientes. Así pues, lleva a los alumnos a captar información a través de la experiencia, estimulando el interés en los temas tratados.
- Aprendizaje de máquina: tiene en cuenta sistemas que permiten a los usuarios añadir posibles soluciones o análisis al *software*; estos sistemas deben contener técnicas de aprendizaje de máquina, los cuales admiten resumir y analizar el proceso que sigue un estudiante para resolver determinados problemas, así como para cuantificarlos y cualificarlos empleando técnicas estadísticas como la media, mediana, moda, proceso más usado, entre otras.

Arquitectura de un ITS

Al hablar de la arquitectura de un sistema tutor inteligente, se debe tener presente que tiene principalmente cuatro componentes: módulo de dominio, módulo del estudiante, módulo del tutor y módulo de interfaz de usuario (Figura 1).

El módulo de dominio utiliza la inteligencia artificial, principalmente los agentes inteligentes, para interactuar de una manera eficaz y autónoma con el entorno, además realiza una retroalimentación adecuada en el proceso de aprendizaje y analiza las diferentes soluciones presentadas ante un problema propuesto. Dentro de este módulo pueden existir diversos agentes con múltiples enfoques: la interacción con el usuario

(agentes de interfaz), la gestión de problemas (agentes de recuperación), la gestión de soluciones (agentes adaptativos) y la gestión de la evaluación (agentes de revisión).

En el proceso de interacción con el usuario, los agentes de interfaz llevan a cabo tareas tales como atender las solicitudes de los estudiantes, presentar y guardar información y delegar tareas a otras entidades; de igual forma, los entes de recuperación se encargan de generar una estrategia de enseñanza de acuerdo con el comportamiento del usuario.

Los agentes adaptativos, por otra parte, se ocupan de almacenar, reutilizar y organizar las diversas soluciones generadas por el estudiantado en cada problema, y por último, el agente de revisión se encarga de evaluar las soluciones y de actualizar el modelo de aprendizaje del estudiante.

El segundo de los módulos, es decir, el del estudiante, se hace cargo de mantener la información acerca de habilidades, conocimientos, aptitudes cognitivas, históricos y el nivel actual de aprendizaje de los usuarios. De igual forma, el módulo tutor hace posible el acceso a las funciones proporcionadas por los agentes del módulo de dominio y se responsabiliza de gestionar los usuarios.

En último término, la interfaz de usuario integra la información de los tres componentes, interpretando los procesos mediante diálogos (Kunyanuth, Pubet & Pattarapan, 2015).

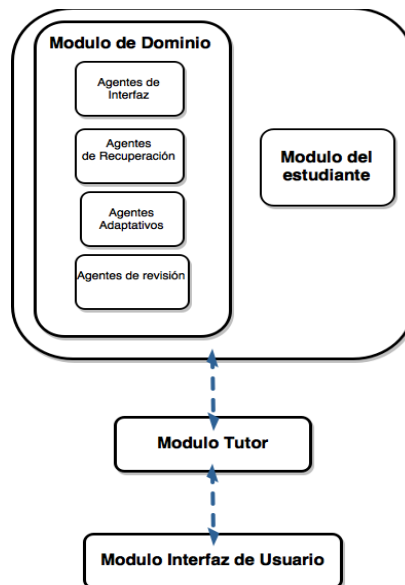


Figura 1. Arquitectura sistema tutor inteligente. Adaptado de [13].

Conclusiones

A partir del análisis y la revisión realizados anteriormente, se puede concluir que para el desarrollo de un sistema tutor inteligente enfocado en el diseño de las bases de datos, existen diferentes tecnologías, dentro de las que se encuentran los agentes inteligentes, así como una arquitectura fiable compuesta de cuatro módulos.

Por otra parte, el desarrollo y la amplia capacidad de aplicación de la inteligencia artificial permiten su implementación en los sistemas tutores, así como en el progreso de los métodos de enseñanza y el proceso de aprendizaje de los estudiantes, pues presenta la capacidad de percibir y analizar el entorno y de generar una acción adecuada ante cualquier evento.

Por lo tanto, es importante resaltar que los ITS orientados al área de las bases de datos ayudan en el proceso de enseñanza de diseño de estas, pues proporcionan ayuda en la realización de tareas, llevan a cabo el seguimiento de los resultados de los estudiantes, reportan los errores cometidos durante el transcurso del modelado y brindan una retroalimentación continua junto con ejercicios personalizados.

Referencias

- Chu, R. W., Mitchell, C.M. & Govindaraj, T. (1989). Characteristics of an ITS that evolves from tutor to operator's assistant. In *IEEE International Conference on Systems, Man and Cybernetics*, 778-783. DOI: <https://doi.org/10.1109/ICSMC.1989.71399>.
- González, C., Burguillo, J. C. & Llamas, M. (2006). A Case-Based Approach for Building Intelligent Tutoring Systems. In *7th International Conference on Information Technology Based Higher Education and Training*, 442-446. DOI: <https://doi.org/10.1109/ITHET.2006.339795>.
- González Pinzón, M., & González Sanabria, J. (2013). Aplicación del estándar ISO/IEC 9126-3 en el modelo de datos conceptual entidad-relación. *Revista Facultad de Ingeniería*, 22(35), 113-125. DOI: <https://doi.org/10.19053/01211129.2519>.
- Hall, L. & Gordon, A. (1998). A virtual learning environment for entity relationship modelling. *SIGCSE Bulletin (Association for Computing Machinery, Special Interest Group on Computer Science Education)*, 30(1), 345-349. DOI: <https://doi.org/10.1145/273133.274327>.
- Kunyanuth, K., Pubet, K. & Pattarapan, R. (2015). Developing an Adaptive Web-based Intelligent Tutoring System Using Mastery Learning Technique. *Procedia Social and Behavioral Sciences*, 191, 686-691. DOI: <https://doi.org/10.1016/j.sbspro.2015.04.619>.
- Li, D., Zhuying, L. & Bing, Y. (2010). The application of AI technology in Intelligent Tutoring System. In *2nd International Conference on Education Technology and Computer*, V1-490-V1-493.
- Mendjoge, N., Joshi, A. R. & Narvekar, M. (2016). Intelligent tutoring system for Database Normalization. In *International Conference on Computing Communication Control and automation (ICCUBEA)*, 1-6. DOI: <https://doi.org/10.1109/ICCUBEA.2016.7860013>.
- Mitrovic, A. (2002). NORMIT: A Web-enabled tutor for database normalization. In *International Conference on Computers in Education*, 1276-1280. DOI: <https://doi.org/10.1109/CIE.2002.1186210>.
- Nesbit, J. E., Adesope, O., Liu, Q. & Ma, W. (2014). How Effective are Intelligent Tutoring Systems in Computer Science Education? In *14th International Conference on Advanced Learning Technologies*, Athens, 99-103. DOI: <https://doi.org/10.1109/ICALT.2014.38>.

- Pramuditha, S. & Antonija, M. (2004). An Intelligent Tutoring System for Entity Relationship Modelling. *International Journal of Artificial Intelligence in Education*, 14, 375-417.
- Thinakaran, R. & Ali, R. (2014). Review on constraint based modelling in intelligent learning environment. In *4th International Conference on Engineering Technology and Technopreneuship (ICE2T)*, 32-36. DOI: <https://doi.org/10.1109/ICE2T.2014.7006214>.
- Wang, L. J., Lu, W. G. & Wang, C.H. (2012). Research on key technology of cheerleading intelligent tutoring system design. In *7th International Conference on Computer Science & Education (ICCSE)*, 1498-1501. DOI: <https://doi.org/10.1109/ICCSE.2012.6295348>.
- Yacef, K. (2002). Intelligent teaching assistant systems. In *International Conference on Computers in Education*, 136-140. DOI: <https://doi.org/10.1109/CIE.2002.1185885>.
- Yang, F. (2011). A virtual tutor for relational schema normalization. *ACM Inroads*, 38-42. DOI: <https://doi.org/10.1145/2003616.2003632>.
- Yang, F. (2013). The Dynamic Curriculum Planning for a Database Design Tutoring System. In *Proceedings of the World Congress on Engineering and Computer Science*.
- Zhang, J. (2005). Intelligent tutoring systems: research status and its development in China. In *International Conference on Natural Language Processing and Knowledge Engineering*, 683-689.