



Diseño del componente generación de tutoría para el aprendizaje de la programación orientada a objetos

Design of the component generation of tutoring for the learning of object-oriented programming

Ivelisse Teresa MACHÍN Torres [1](#)

Recibido: 27/11/2017 • Aprobado: 21/12/2017

Contenido

- [1. Introducción](#)
- [2. Metodología](#)
- [3. Resultados](#)
- [4. Discusión](#)
- [5. Conclusiones](#)

[Referencias bibliográficas](#)

RESUMEN:

La enseñanza de la Programación Orientada a Objetos y los sistemas tutores inteligentes presentan dificultades, lo cual fundamenta la utilidad de tutores inteligentes con técnicas de minería de datos. Se propone diseñar un componente generación de tutoría, usando técnicas de minería de datos que extraen y analizan la información de las interacciones del estudiante, que considere los tipos de inteligencia preponderante en cada alumno, favoreciendo soluciones individualizadas. Los resultados muestran mejoría en la solución de problemas, con un óptimo proceso de aprendizaje.

Palabras clave: Minería de datos, sistema tutor inteligente, inteligencias múltiples, programación orientada a objetos

ABSTRACT:

The teaching of Object Oriented Programming and intelligent tutoring systems present difficulties, which underlies the usefulness of intelligent tutors with data mining techniques. It is proposed to design a generation component of tutoring, using data mining techniques that extract and analyze the information of student interactions, which considers the types of intelligence prevailing in each student, favoring individualized solutions. The results show improvement in the solution of problems, with an optimal learning process.

Keywords: Data mining, intelligent tutor system, multiple intelligences, object-oriented programming.

1. Introducción

La Programación Orientada a Objetos (POO) constituye una nueva forma de pensar acerca de

qué significa computar, y cómo se puede estructurar la información dentro de un computador. El interés por el estudio de este tema aumentó desde la década del sesenta con la introducción de la computación en diversas aplicaciones, y expandió su línea de vida a partir de la implementación de Simula-67. Sin embargo, el aprendizaje eficaz de los estudiantes que cursan las temáticas de la POO en las universidades es aún insuficiente, a pesar de algunos intentos por mejorarlo. Las principales dificultades identificadas en el aprendizaje de la POO que pueden dar cuenta de los bajos resultados académicos son:

Dificultad 1: Los estudiantes no saben describir los problemas que se plantean.

Dificultad 2: Escriben código que no cumple con los estándares definidos.

Dificultad 3: En las soluciones no se expresa exactamente lo que se necesita.

Dificultad 4: Mal uso de términos o estructuras.

Dificultad 5: Abstraer la esencia de un problema y no extrapolar ideas.

Estas dificultades han sido descritas por diversos autores (Lezcano y Valdés, 1998; Ríos, 2009) que utilizaron en sus estudios sistemas tutores inteligentes (STI) para el aprendizaje de la programación. Por otra parte, Hakimzadeh y Adaikkalavan (2011) reportan los resultados de un estudio donde se evidencia que la enseñanza de la programación presenta deficiencias que se reflejan en los bajos índices de retención.

Diferentes trabajos proponen como alternativa a esta situación, complementar los conocimientos teóricos con actividades prácticas (Jurado, 2010). Lazar y Bratko (2014) sugieren la mejora de la efectividad mediante la praxis. Por otra parte, se presenta un estudio determinante para la retroalimentación, se impone pues, la necesidad de perfeccionar los medios que ayudan a optimizar la calidad del proceso enseñanza aprendizaje de esta materia, teniendo en cuenta que la misma tiene sus complejidades y puede estar viciada por técnicas aprendidas en otros paradigmas que nada tienen que ver con el de la POO (Stamper y Eagle, 2013).

Conviene recordar que los STI se desarrollaron con la idea de impartir conocimiento guiando en el proceso de aprendizaje a partir de alguna forma de inteligencia. Un STI ha sido definido como un sistema de software que utiliza técnicas de inteligencia artificial para representar el conocimiento e interactúa con los estudiantes para enseñárselo (VanLehn, 1988).

Generalmente se componen de cuatro módulos, el módulo del estudiante, el módulo del dominio, módulo del tutor y el módulo de interfaz.

Por otra parte, el uso de las inteligencias múltiples, se orienta fundamentalmente a proporcionar métodos objetivos de adaptabilidad que establezcan una ruta de aprendizaje personalizada, tomando en cuenta factores sensoriales, cognitivos, de personalidad o sociales que caracterizan y definen la unicidad de cada alumno (Gardner, 2011). Esto ha suscitado una búsqueda de modelos de enseñanza/aprendizaje que supone a su vez la aceptación de que no prevalece un único método válido de enseñanza (Gargallo, 2006). Lo que tiene que ofrecer el STI es una variedad de tareas que puedan adaptarse a los distintos estilos de aprendizaje.

Como resultado de la aplicación de STI se reportan ganancias significativas en el aprendizaje con respecto a otros tipos de sistemas que se usan para apoyar el proceso de enseñanza-aprendizaje (Martínez, 2009; Chi, VanLehn y Litman, 2010; Barbhuiya, Mustafa y Jabin, 2011; El Haddad y Naser, 2017; Elnajjar y Naser, 2017).

Autores como (Aleven, 2010; Eagle y Barnes, 2013) destacan como causa principal de estos resultados positivos la retroalimentación que recibe el estudiante durante la realización de actividades. Considerando que existe una relación directa entre la cantidad de retroalimentación y estas ganancias (Chi y otros, 2010).

Para realizar la tutoría adaptativa se usan técnicas de minería de datos que, aunque varían de un sistema a otro, responden a la representación de tipos de conocimiento similares: conocimiento sobre el dominio o materia objeto de estudio, para saber qué enseñar;

conocimiento sobre la persona que aprende, para saber a las características de quien adaptar; conocimiento sobre el proceso de tutoría, para saber cómo enseñar y conocimiento sobre cómo debe ser la interacción entre sistema y usuario.

La minería de datos consiste en extraer conocimiento útil, previamente desconocido, a partir de grandes volúmenes de datos (Baker y Siemens, 2014).

En la búsqueda bibliográfica realizada se encontraron múltiples ejemplos de STI que han sido desarrollados y aplicados con éxito para la programación. Dentro de este ámbito se encontraron trabajos que tienen como meta generar retroalimentación a través de las técnicas de minería de datos. Yeh y Lo (2006) presentan un trabajo que usa redes neuronales que evalúan automáticamente el nivel de conocimiento del alumno observando su comportamiento. En otro enfoque se desarrolla un STI basado en la teoría de respuesta a un ítem borroso, donde se recomiendan materiales del curso con niveles de dificultad acordes a las respuestas de realimentación del alumno inciertas o borrosas sobre la comprensión y la dificultad de los materiales propuestos (Chen y Duh, 2008). Existen propuestas alternativas que utilizan agentes recomendadores para sugerir actividades de aprendizaje en línea en un curso Web basándose en los historiales de acceso (Dahotre, 2011). Sin embargo, al limitarse la retroalimentación no se obtienen las ganancias observadas en otras áreas de aplicación.

En este reporte de investigación se estudian los enfoques que se han utilizado para generar retroalimentación, que permiten al estudiante trabajar sobre la base de los errores cometidos. Se propone el diseño de un componente generación de tutoría que favorezca la asimilación de contenidos a través de la retroalimentación lograda con técnicas de minería de datos y oriente al estudiante en la solución de problemas relacionados con las temáticas de la POO.

2. Metodología

La investigación se desarrolló en diferentes universidades cubanas durante el período de tiempo comprendido entre el año 2015 y el 2017; se estructuró en tres etapas:

2.1 Primera etapa (diagnóstico del estado del proceso de aprendizaje de la POO mediado por STI):

Con el objetivo de identificar la necesidad de un componente generación de tutoría para el aprendizaje de la POO, se realizó un estudio exploratorio con enfoque mixto, cuantitativo-cualitativo.

En esta etapa se establecieron los siguientes pasos.

Sub-etapa 1

Se realizó un estudio cuantitativo exploratorio durante el año 2015, en la Universidad José Martí Pérez de Sancti Spíritus y Universidad Agraria de la Habana.

Primeramente, se revisaron los registros docentes que contenían los exámenes finales, planteados en los expedientes de los estudiantes egresados.

Luego se seleccionaron los expedientes (población) de aquellos estudiantes con exámenes que presentaban bajas calificaciones en las materias de programación.

Se realizó un muestreo no probabilístico de tipo intencional que permitió definir una muestra de casos-tipo, considerando como criterio de selección la identificación (en el expediente) de los estudiantes que habían presentado bajas calificaciones en un año, teniendo en cuenta ese período de forma retrospectiva a partir de la fecha en que el estudiante egresó.

Los principales resultados esperados fueron: identificar cuántos estudiantes habían recibido bajas calificaciones en POO en el período de un año, teniendo como referencia la fecha en que el estudiante egresó, y cuántos de ellos recibieron orientación y ayuda por parte de los profesores.

Sub-etapa 2

Se realizó una investigación cualitativa exploratoria con el objetivo de profundizar en aspectos que caracterizaban el proceso de aprendizaje de la POO mediado por STI.

El estudio se desarrolló en diferentes universidades donde se imparte la POO, tomando como contexto el territorio cubano, durante el año 2015. Los escenarios escogidos fueron: la Universidad José Martí Pérez de Sancti Spíritus y Universidad Agraria de la Habana.

En cada uno de los escenarios se seleccionó la población, conformada por los profesores especialistas en programación, en tecnología educativa o en otras especialidades con perfil informático que laboraban tanto en las universidades como en los centros de estudios.

Se realizó un muestreo no probabilístico de tipo intencional que permitió seleccionar una muestra homogénea, considerando como criterio de selección que los profesionales fueran especialistas en programación, en tecnología educativa o en otras especialidades con perfil informático.

La muestra quedó constituida por 35 profesores, de ellos 25 especialistas en programación y cinco especialistas en tecnología educativa; además de 5 profesionales que laboraban como profesores en las carreras de perfil informático.

Estos últimos estuvieron distribuidos en las siguientes especialidades: dos especialistas en Bases de Datos, dos en Ingeniería del Software, y uno en Inteligencia Artificial.

Del total, 27 realizaban la docencia a tiempo completo y 3 eran a tiempo parcial, dos administrativos, dos vinculados al desarrollo de sistemas y uno jefe de la disciplina programación.

El análisis de la situación se realizó a partir de la aplicación de las siguientes técnicas: encuestas, entrevistas en profundidad, grupos focales y observación participante.

Análisis de la información en la Primera Etapa:

En la sub-etapa 1 se utilizó la estadística descriptiva: frecuencias absolutas y relativas.

En la sub-etapa 2 se realizó un análisis cualitativo en el que la información obtenida fue organizada en categorías. Una vez que se alcanzó el máximo de la información, se realizó una discretización de los datos, codificándolos en las siguientes categorías: cognitivos, estructurales y resultados. Posteriormente, se relacionaron las categorías y se triangularon los datos.

Atendiendo a los resultados obtenidos en el diagnóstico del estado real del proceso de aprendizaje de la POO mediado por STI se prosiguió a diseñar el componente generación de tutoría para mejorar la asimilación de contenidos y la orientación en la solución de problemas en el aprendizaje la POO.

2.2. Segunda etapa (diseño del componente):

Basado en fundamentos teóricos y prácticos se identificaron los artefactos y su modo de integración; se analizaron las propuestas de minería de datos para la solución de los problemas identificados en la primera fase; se implementó el algoritmo de selección de atributos relevantes como parte del proceso de diseño del componente generación de tutoría para mejorar la asimilación de contenidos y la orientación en la solución de problemas en el aprendizaje la POO.

2.3. Tercera etapa (evaluación del componente):

Con el objetivo de evaluar la calidad del componente propuesto para la enseñanza de la POO, teniendo en cuenta los factores críticos de éxito para identificar las fortalezas y debilidades de su puesta en práctica, el mismo fue validado mediante un cuasi experimento.

Se aplicaron los exámenes a un grupo de estudiantes, antes (pre prueba) y después (post

prueba) de la aplicación del componente para evaluar el comportamiento de las variables: inteligencia lógico matemática, lingüística, interpersonal, gestión del conocimiento, orientación en la solución de problemas y asimilación de contenidos.

El mismo examen fue aplicado a un grupo que desarrollaron sus estudios sin el empleo de la herramienta (pre prueba). Para llevar a cabo el cuasi experimento del tipo pre y post prueba con grupo de control se utilizó una muestra de 25 perfiles de alumnos pertenecientes a la carrera de Ingeniería Informática. El experimento se realizó a lo largo de un semestre y en ese período cada participante trabajó sobre un conjunto de actividades.

Para el grupo experimental al registrarse y comenzar a trabajar en los cursos se contó con el soporte descrito en este trabajo. A los estudiantes se les orientó trabajar en el curso Programación Orientada a Objetos, un enfoque práctico.

3. Resultados

3.1. Primera etapa- principales resultados del diagnóstico

Sub-etapa 1

En el año 2015 más del 78% de estudiantes egresados, presentaron dificultades en los temas de programación durante alguna etapa de sus estudios, de ellos el 21,6 % habían participado en extraordinarios dos o más ocasiones, en el período de un año, por alguno de los temas de programación. Ese porcentaje puede considerarse dentro del rango que aparece en la literatura, ya que se han publicado porcentajes entre 67 % y 89 % de estudiantes con dificultades en programación.

No obstante, la identificación de estas deficiencias, a pesar de las acciones correctivas desarrolladas por los profesores y el apoyo ofrecido por los centros educacionales, en ninguno de los expedientes se reflejó documentación referente a un estudio profundo realizado con el objetivo de identificar causas subyacentes responsables de las bajas calificaciones, y falta de motivación en los estudiantes.

La valoración cualitativa de los resultados obtenidos indicó, que aun cuando existieron factores incidentes en el desempeño de los estudiantes, todos los registros y acciones identificadas por parte de la docencia se basaron en el proceso tradicional con el cual el estudiante se somete a nuevos exámenes, en cada ocasión.

Sub-etapa 2

Las dos terceras partes de los encuestados refiere la no utilización de los STI debido a desconocimiento, poca importancia atribuida a los mismos o por considerarlos engorrosos, aunque todos reconocen la conveniencia de que sus instituciones lo asuman, siempre que sea interoperable y de fácil actualización, ya que constituye una forma de enfrentar los procesos de obsolescencia tecnológica.

Así pues, se identificó una primera dimensión concerniente a la necesidad de desarrollar un modelo que cuente con un STI dentro de sus componentes y que integre factores tecnológicos, metodológicos y pedagógicos para favorecer la gestión del conocimiento referente a la asignatura programación y mejorar la orientación en la solución de problemas.

Por otra parte, los encuestados destacan la necesidad de lograr una gestión colaborativa de los recursos educativos que forman parte del STI (simuladores, emuladores, directorios temáticos, foros, comunicación mediante redes sociales), porque favorece el intercambio de experiencias entre estudiantes, adaptándose con mayor flexibilidad a sus perfiles y promoviendo el pensamiento crítico.

En el análisis de la interfaz se pudo apreciar que las funcionalidades seleccionadas con mayor frecuencia para garantizar la adaptabilidad de los contenidos son: realización de test a los estudiantes, añadir contenidos de manera ponderada por parte del sistema, monitoreo de

forma continua en la resolución de actividades, particularización a partir de bases de datos, inclusión de evaluadores y métodos para la solución cooperativa.

Dentro de los elementos a tener en cuenta para garantizar la adaptabilidad del STI, los encuestados plantearon la necesidad de que el sistema diseñe el contenido de forma estratégica de acuerdo con las debilidades encontradas en el estudiante, y teniendo en cuenta su progreso, además realice y acepte sugerencias por parte del estudiante, acerca de la presentación de los contenidos. Esta segunda dimensión se denominó funcionalidades en un sistema tutor inteligente para garantizar la adaptabilidad de los contenidos y la orientación en la solución de problemas.

Los resultados de las encuestas y de los grupos focales aplicados mostraron diferentes criterios en cuanto a qué indicadores deben tenerse en cuenta para evaluar la adaptabilidad de los STI a los diferentes perfiles de estudiante, mediante metodologías e instrumentos de evaluación. Así mismo, en el grupo focal desarrollado se destacó la necesidad de incorporar analíticas diferentes a los existentes en Moodle en sus versiones desplegadas, basado en la experiencia de los usuarios, las trazas de inteligencia y desempeño de los administradores.

Se obtuvo también evidencia de que la participación colaborativa es deficiente en la resolución de actividades y el aprendizaje de contenidos a través del STI, pues casi la mitad de los sujetos investigados refiere que nunca se logra y el resto que se logra poco. Esta resultó la tercera dimensión importante a tomar en cuenta, o sea, la colaboración, comunicación e intercambio de conocimientos entre estudiantes que utilizan STI, ya que todos coinciden en la necesidad de que exista un STI que posibilite la elaboración conjunta. Las razones que afectan el trabajo colaborativo, según los encuestados son: las herramientas que utilizan los STI no propician el trabajo colaborativo, existe reticencia por parte de los profesores relacionada con el uso del STI, los profesores en ocasiones poseen pocas habilidades para el trabajo en equipo.

3.2. Segunda etapa- diseño del componente generación de tutoría

Atendiendo a los resultados obtenidos en el diagnóstico se define el componente generación de tutoría que pertenece al modelo para mejorar la asimilación de contenidos y la orientación en la solución de problemas en la Programación Orientada a Objetos en las carreras de perfil informático (MACPOO) (Machín, 2017a).

Los **principios** que sustentan el modelo son:

- La estandarización para hacer más eficiente el proceso de composición y reutilización de recursos.
- La interoperabilidad para lo cual es indispensable un diseño adecuado de los recursos, que permita que las aplicaciones sean compatibles y usables por cualquiera sin importar las plataformas tecnológicas donde se ejecuten proveedor y consumidor.
- La flexibilidad por estar basados en componentes con funcionalidades genéricas y adaptarse a las particularidades de las aplicaciones existentes para el proceso de composición.
- La pertinencia como garantía de la adecuación del modelo en el contexto de la orientación a servicios.
- La independencia funcional de los servicios lo que contribuye a su bajo acoplamiento y a su reutilización.

Las **cualidades** son:

- Amplitud que brinda la capacidad de analizar y de emplearse en aplicaciones informáticas de múltiples dominios.
- Enfoque Sistémico que se expresa en el modelo propuesto a través de los componentes que interactúan con vistas a perfeccionar el proceso de composición de aplicaciones.

-Integralidad dada por los componentes del modelo que cubren de manera integrada y coherente la mayoría de los elementos necesarios para la composición satisfactoria de aplicaciones.

-Mejora continua, al retroalimentarse con los resultados que se van obteniendo, en los diferentes módulos que dan tratamiento a la información.

Se parte de las siguientes **premisas**:

-La calificación de los desarrolladores necesaria para el uso eficiente de las herramientas propuestas para la gestión del conocimiento referente a la aplicación de la POO.

-La voluntad institucional que apoye la aplicación del modelo y la visibilidad de los recursos necesarios para la gestión del conocimiento referente a la aplicación de la POO.

-La existencia de materiales didácticos debidamente catalogados.

-La calidad de los datos contenidos en los logs en los LMS y otras fuentes externas.

Sobre la base de los principios teóricos antes expuestos y del diagnóstico inicial realizado se propone el componente generación de tutoría para el aprendizaje de la POO con sus entradas, salidas, sub-componentes y relaciones.

Entradas: información extraída de los LMS u otras fuentes (IL) y perfiles inteligentes. La IL está relacionada esencialmente con la información existente en los logs, siendo I la cantidad de atributos que puede tal como se define en la fórmula [1]:

$$IL = \{l_1, l_2, \dots, l_j\}, j > 0, j \in \mathbb{N} \quad [1]$$

Salidas: la información referente al listado de calificaciones (C), representada como se muestra a continuación:

$$C = \{c_1, c_2, \dots, c_e\}, e > 0, e \in \mathbb{N} \quad [2]$$

La información de reportes académicos (F), representada de la siguiente manera:

$$F = \{f_1, f_2, \dots, f_e\}, e > 0, e \in \mathbb{N} \quad [3]$$

La información del listado de recomendaciones (W), representada como se muestra a continuación:

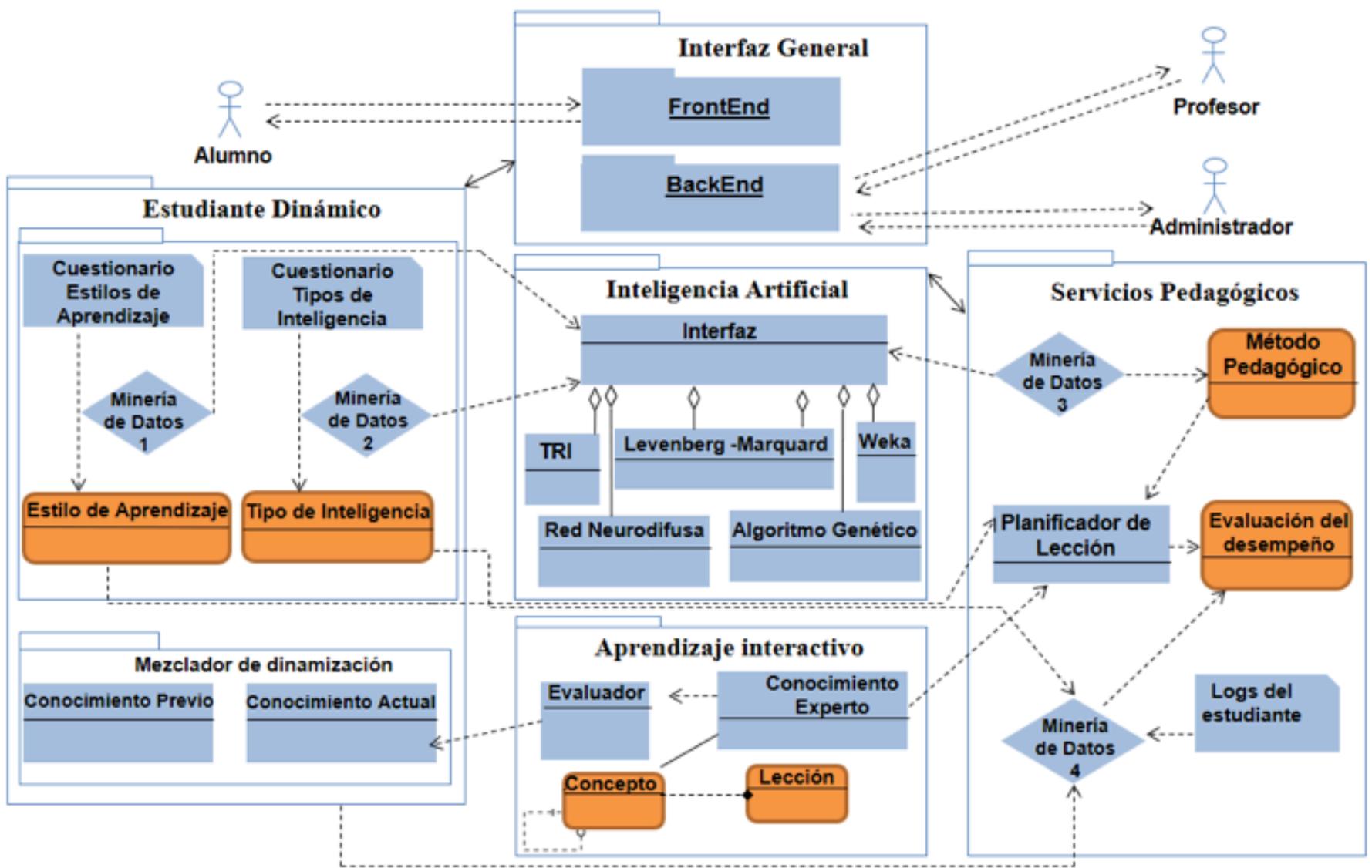
$$W = \{w_1, w_2, \dots, w_e\}, e > 0, e \in \mathbb{N} \quad [4]$$

La información de análisis de inteligencias (H), representada de la siguiente manera:

$$H = \{h_1, h_2, \dots, h_e\}, e > 0, e \in \mathbb{N} \quad [5]$$

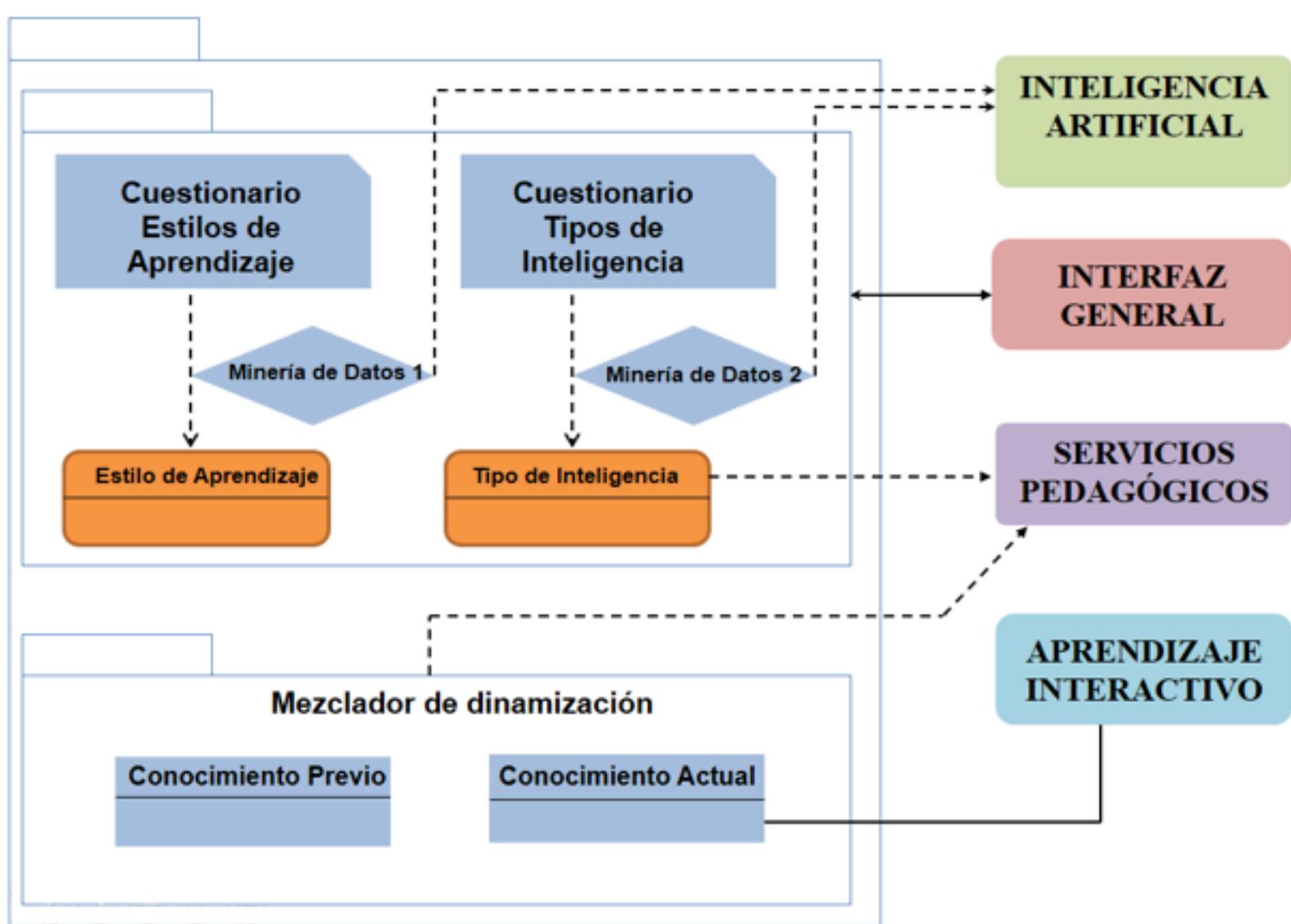
Con el objetivo de aumentar la formalidad y el entendimiento de este componente, se desarrolló el diagrama reflejado en la Figura 1, donde se muestran los diferentes módulos que lo integran.

Figura 1
Diagrama de bloques del componente generación de tutoría



El Módulo Estudiante Dinámico está compuesto por dos bloques diferenciados (Figura 2), uno que modela los rasgos que tienen poca variación a lo largo del tiempo (inteligencias múltiples y preferencias de aprendizaje) y otro que recoge las características dinámicas (estado del conocimiento del alumno), que se actualizan constantemente porque dependen de la interacción del alumno con el curso o Módulo Aprendizaje Interactivo.

Figura 2
Diagrama del Módulo Estudiante Dinámico



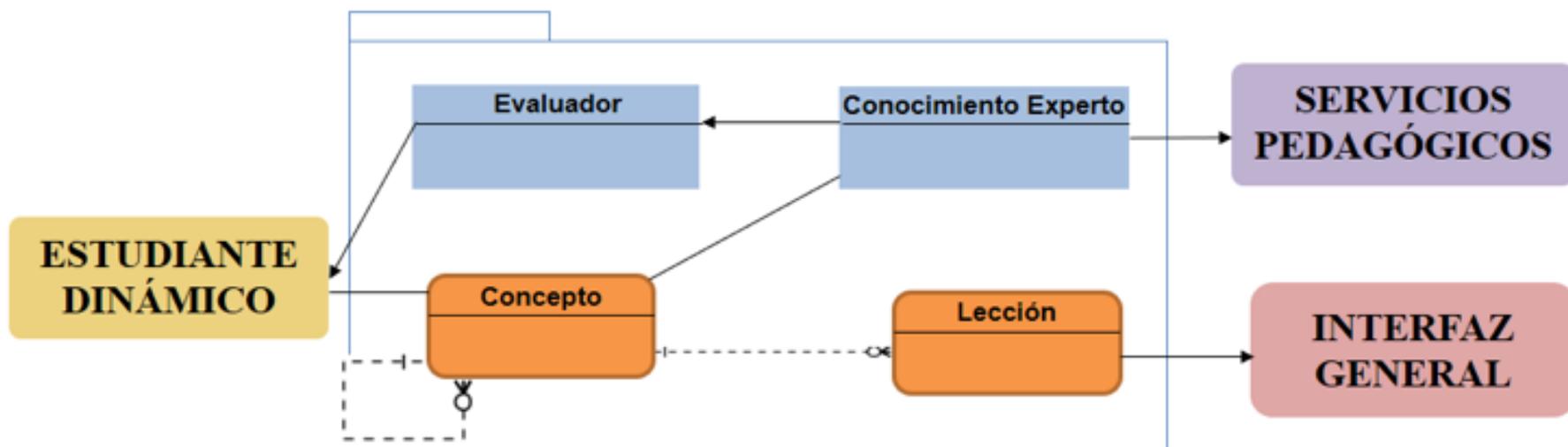
Fuente: elaboración propia.

A su vez, los dos elementos del Módulo Estudiante Dinámico, alimentan a varios aspectos que caracterizan el Módulo Servicios Pedagógicos, definiendo la adaptabilidad del STI. Con estas relaciones se establece el método pedagógico de enseñanza adaptativo, los contenidos adecuados (secuenciación de los recursos u objetos de aprendizaje) y el modo de presentarlos (tipo de recursos multimedia o colaborativos).

Lo anteriormente mencionado tiene el fin de planificar la ruta de aprendizaje óptima para el alumno en función de su perfil o modelo de aprendizaje. Por otro lado, también realimentan el Módulo Aprendizaje Interactivo, indicando las preferencias del alumno con respecto al aspecto y al conjunto de herramientas que configuran la interfaz de usuario.

En la Figura 3 se observa la estructura del Módulo Aprendizaje Interactivo, que está compuesto por la ruta de aprendizaje que viene definida por el Módulo Servicios Pedagógicos. Esta ruta de aprendizaje contiene la secuenciación a aplicar para definir la unidad didáctica, que se debe materializar en una serie de tareas que vienen diseñados y condicionados por el curso y que se ofrecen al alumno a través de una interfaz de usuario.

Figura 3
Diagrama del Módulo Aprendizaje Interactivo.

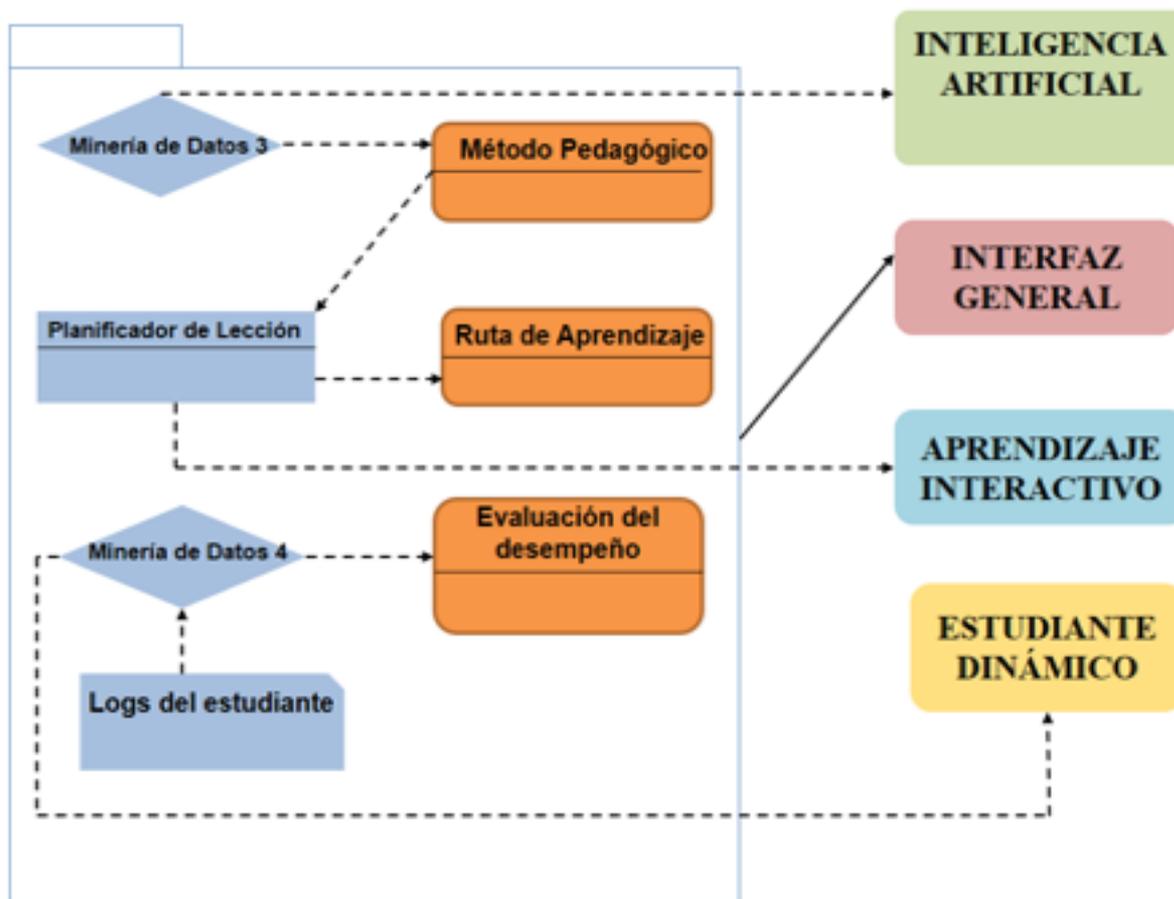


Fuente: elaboración propia.

La interacción del alumno con las tareas en la interfaz produce una serie de informes, que almacenan esta interacción (ficheros log) así como los resultados de la realización de las tareas, al comparar las respuestas o acciones del alumno, con las del Módulo Servicios Pedagógicos definido en el curso. El análisis de esta información (los ficheros log y los resultados) mediante minería de datos proporciona el progreso del alumno, que se puede descomponer en resultados o logros que afectarán a la motivación del alumno y en el nivel de conocimiento actual o aprendizaje adquirido hasta la fecha.

En la Figura 4 se observa la estructura del Módulo Servicios Pedagógicos que tiene la función de seleccionar la intervención educativa más apropiada. Ésta se realiza comparando el Módulo Estudiante Dinámico con los resultados esperados del Módulo Aprendizaje Interactivo (curso), las discrepancias observadas son señaladas para tomar una acción correctiva acorde.

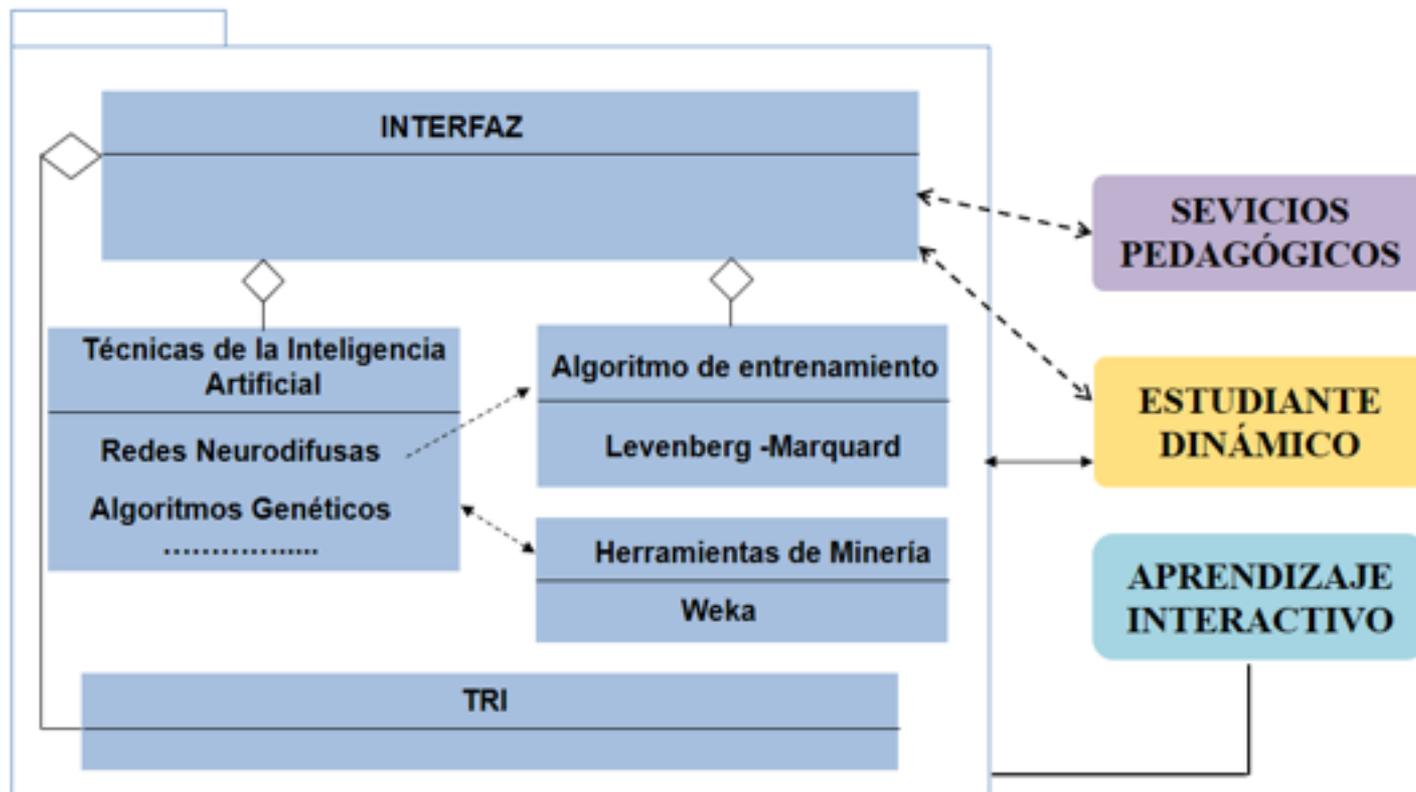
Figura 4
Diagrama del Módulo Servicios Pedagógicos



Fuente: elaboración propia.

Para lograr la selección del método pedagógico apropiado se produce la interacción entre el Experto y el Módulo Inteligencia Artificial, tal como se muestra en la Figura 5.

Figura 5
Diagrama del Módulo Inteligencia Artificial.



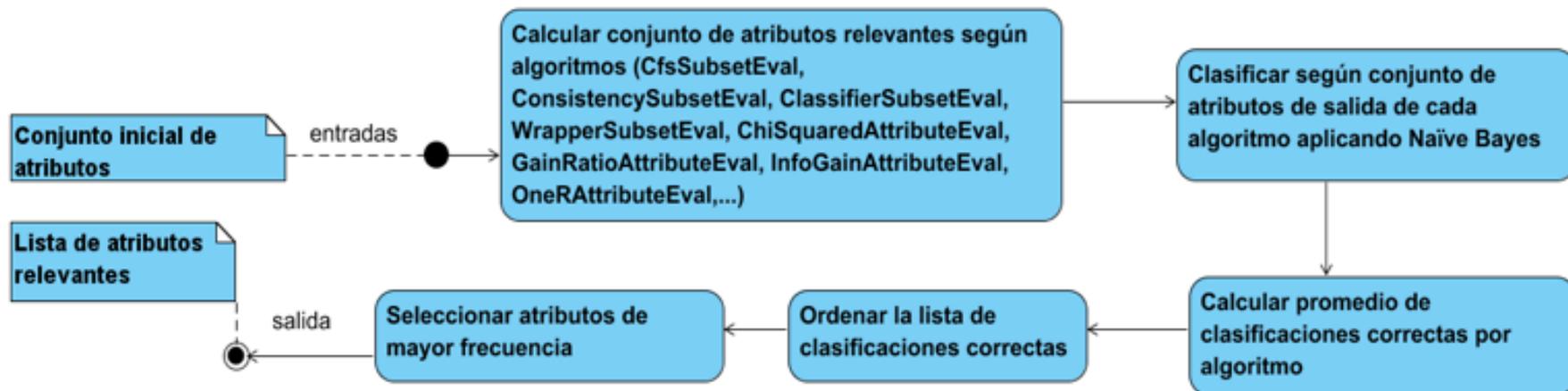
Fuente: elaboración propia

Este módulo recibe como entradas el perfil inteligente del estudiante, las trazas de desempeño dentro del curso matriculado, así como información académica de fuentes externas. El objetivo de este módulo es integrar técnicas de minería de datos aplicables a los sistemas tutores inteligentes para mejorar la asimilación de contenidos, la gestión del conocimiento y la orientación en la solución de problemas. Las entradas al módulo están definidas por los registros generados por el Módulo Estudiante Dinámico, el Módulo Servicios Pedagógicos y el Módulo Aprendizaje Interactivo, mientras que las salidas son los informes que reflejan el comportamiento del estudiante dentro del curso.

Se implementó un sistema híbrido (Machín, 2017b) que contiene a su vez un sistema neurodifuso estilo Mamdani-Anfis, que utiliza como algoritmos de entrenamiento para la red, los algoritmos genéticos enfocados a la obtención de una topología de red óptima con el menor número de neuronas e incluso de conexiones, utilizando un algoritmo híbrido, lo cual significa en la presente investigación un algoritmo evolutivo multiobjetivo que utiliza aparte un algoritmo específico para el entrenamiento de los individuos como es el de Levenberg Marquard (Filali, 2005).

El componente tiene en cuenta el conjunto de funcionalidades que le permiten tener una adaptación automática a los planes de estudio, los contenidos y los estilos pedagógicos del curso y el proceso de aprendizaje del alumno. A su vez tiene la capacidad de permitir el seguimiento del comportamiento y predicción de rendimiento de los estudiantes en el curso. En la Figura 6 se presentan los principales pasos del algoritmo de selección de atributos que inciden sobre el rendimiento académico.

Figura 6
Algoritmo para la selección de atributos.



Fuente: elaboración propia.

En una primera fase se carga el archivo de datos históricos (.arff) correspondiente al rendimiento académico de los estudiantes en un curso contiene la totalidad de los datos y atributos a ser procesados.

En una segunda fase se calcula el porcentaje de resultados correctos obtenidos utilizando como entrada para la red Naïve de Bayes cada uno de los datos correspondientes a los subconjuntos de atributos seleccionados.

En la tercera y última fase del algoritmo se ordenan los atributos por frecuencia de aparición en los subconjuntos. Se obtiene el conjunto de atributos relevantes para la predicción del rendimiento académico.

3.3. Tercera etapa- evaluación del componente generación de tutoría

A continuación, se presentan los resultados obtenidos en el cuasiexperimento comparando, mediante una d'cima de comparación de proporciones al nivel de significación del 0,05, los valores de pre prueba y post prueba (antes y después de la aplicación de la herramienta) en el grupo experimental en relación con cada uno de los indicadores de las variables mencionadas.

Los resultados se expresan en porcentaje calculado como la razón entre la cantidad de estudiantes de cada categoría entre el total (Tabla 1).

Tabla 1
Comparación de los valores de las variables dependientes y los tipos de inteligencias antes y después de la aplicación del componente.

Variables	Categorías	Pre	Post	Valor P*
Inteligencia Lógico Matemática	Baja	0,72	0,20	0,0001
	Media	0,20	0,48	0,0099
	Alta	0,08	0,32	0,0072
Inteligencia Lingüística	Baja	0,75	0,25	0,0002
	Media	0,21	0,49	0,0105
	Alta	0,04	0,26	0,0050
	Baja	0,77	0,36	0,0048

Inteligencia Interpersonal	Media	0,20	0,50	0,0063
	Alta	0,03	0,14	0,0385
Gestión del Conocimiento	Baja	0,82	0,45	0,0109
	Media	0,12	0,31	0,0338
	Alta	0,06	0,24	0,0186
Orientación en la solución de problemas	Baja	0,77	0,20	0,0001
	Media	0,20	0,41	0,0403
	Alta	0,03	0,39	0,0001
Asimilación de contenidos	Reproductivo	0,57	0,20	0,0119
	Productivo	0,20	0,44	0,0229
	Creativo	0,23	0,36	0,1586

Fuente: elaboración propia.

El análisis estadístico muestra los resultados de realizar una prueba de hipótesis relativa a la diferencia de dos proporciones de muestras provenientes de distribuciones binomiales.

En la mayoría de los casos se aprecian mejores resultados en el cuasiexperimento de post prueba (después de aplicado de la herramienta) lo que indica una mejora significativa en la contribución de la herramienta debido a que las variables evolucionan a niveles superiores. El aumento significativo de los estudiantes que potencian sus inteligencias queda establecido a partir del análisis del valor-P. Puesto que el valor-P para las pruebas resultó menor que 0,05, pudo rechazarse la hipótesis nula con elevados niveles de confianza.

4. Discusión

Con respecto a otros estudios realizados acerca de las dificultades existentes en el aprendizaje de la POO (Lezcano y Valdés, 1998; Ríos, 2009; Hakimzadeh y Adaikkalavan, 2011), en la presente investigación se ratifican las limitaciones identificadas, se incluyen nuevas limitantes y se propone realizar un análisis de posibles causas subyacentes para la situación de bajas calificaciones y falta de motivación existente en los estudiantes que cursan las temáticas de programación.

En relación al diseño y elaboración de STI, se identificó que deben incluir un conjunto de estándares y especificaciones e integrar factores tecnológicos, metodológicos y pedagógicos que permitan una gestión colaborativa de los recursos educativos.

Por otra parte, los modelos analizados en la literatura (El Haddad, y Naser, 2017; Elnajjar, y Naser, 2017) no toman en cuenta las inteligencias múltiples para la selección del método pedagógico correspondiente a cada estudiante, lo cual disminuye los niveles de personalización y adaptabilidad en relación a la propuesta.

5. Conclusiones

Del análisis de los resultados obtenidos, se puede afirmar que la propuesta de la presente investigación favorece la asimilación de contenidos y la orientación en la solución de problemas en comparación con otros trabajos que tienen un nivel de efectividad similar, lo cual se traduce en mayor aprendizaje por parte de los estudiantes.

La propuesta representa una reducción del tiempo de autoría de las actividades en relación con los métodos que necesitan de la especificación de variantes correctas e incorrectas.

De forma general la propuesta puede contribuir de forma positiva a la enseñanza de la programación.

Referencias bibliográficas

- Aleven, V. (2010). Rule-Based Cognitive Modeling for Intelligent Tutoring Systems. En R. Nkambou, J. Bourdeau y R. Mizoguchi (eds). *Advances in Intelligent Tutoring Systems* (pp. 33-62). Springer-Verlag.
- Baker, R., y Siemens, G. (2014). Educational data mining and learning analytics. En *Cambridge Handbook of the Learning Sciences* (pp. 256-368).
- Barbhuiya, R. K., Mustafa, K., y Jabin, S. (2011). Design specifications for a generic Intelligent Tutoring System. En *Proceedings of the International Conference on e-Learning, e-Business, Enterprise Information Systems, and e-Government* (pp. 399-409). Las Vegas, USA: CSREA Press.
- Chi, M., VanLehn, K., y Litman, D. (2010). Do micro-level tutorial decisions matter: Applying reinforcement learning to induce pedagogical tutorial tactics. En *Proceedings of the International Conference on Intelligent Tutoring Systems* (pp. 224-234). Springer International Publishing.
- Chen, C. M., y Duh, L. J. (2012). Personalized web-based tutoring system based on fuzzy item response theory. *Expert Systems with Applications*, 34 (4), 2298-2315.
- Dahotre, A. (2011). *jTutors: A Web-based tutoring system for Java APIs* (Tesis de Maestría). Oregon State University, School of Electrical Engineering and Computer Science.
- Eagle, M. J. y Barnes, T. (2013). Evaluation of Automatically Generated Hint Feedback. En *Proceedings of the International Conference on Educational Data Mining* (pp. 372-381). Memphis, Tennessee, USA: International Educational Data Mining Society.
- El Haddad, I.A, y Naser, S.S.A. (2017). ADO-Tutor: Intelligent Tutoring System for learning ADO.NET. *European Academic Research*, 4(10), 8810-8821.
- Elnajjar, A. E. A., y Naser, S. S. A. (2017). DSE-Tutor: An Intelligent Tutoring System for Teaching DES Information Security Algorithm. *International Journal of Advanced Research and Development*, 2 (1), 69-73.
- Filali, M. (2005). *Desarrollo y optimización de nuevos modelos de redes neuronales basadas en funciones de base radial* (Tesis Doctoral). Granada.
- Gardner, H. (2011). *La Teoría en la Práctica*. Buenos Aires: Paidós.
- Gargallo, B. (2006). Estrategias de aprendizaje, rendimiento y otras variables relevantes en estudiantes universitarios. *Revista de Psicología General y Aplicada* (59), 109-130.
- Hakimzadeh, H., Adaikkalavan, R., y Batzinger, R. (2011). Successful Implementation of an Active Learning Laboratory in Computer Science. In *Proceedings of the 39th annual ACM SIGUCCS conference on User services* (pp. 83-86). ACM.
- Jurado, F. (2010). *Proposal for Evaluating Computer Programming Algorithms to Provide Instructional Guidance and Give Advice* (Tesis Doctoral). Departamento de Tecnologías y Sistemas de Información, Universidad de Castilla-La Mancha.
- Lazar, T., y Bratko, I. (2014). Data-Driven Program Synthesis for Hint Generation in Programming Tutors. En *Proceedings of the International Conference on Intelligent Tutoring*

Systems (pp. 306-311). Honolulu: Springer.

Lezcano, M., y Valdés, V. (1998). Algunas Experiencias en la Utilización de Sistemas de EAC para la Enseñanza de la Inteligencia Artificial. *Revista Divulgaciones Matemáticas*, 6(2), 165-182.

Machín Torres, I. T. (2017a). Sistema para la selección del método pedagógico en sistemas tutores inteligentes/System for selection of pedagogical method in intelligent tutorial systems. *International Journal of Innovation and Applied Studies*, 20(2), 643-651.

Machín Torres, I. T., y Estrada Sentí, V. (2017b). Intelligent Tutor System for Learning Object Oriented Programming. *IRJET*, 4(2), 1-6.

Martínez, N. (2009). Modelo para diseñar Sistemas de Enseñanza-Aprendizaje Inteligentes utilizando el Razonamiento Basado en Casos (Tesis Doctoral). Universidad Central "Marta Abreu" de Las Villas.

Ríos, L. R. (2009). Ambiente de enseñanza-aprendizaje inteligente para la programación lógica (Tesis Doctoral). Universidad "José Martí Pérez", Sancti Spíritus.

Stamper, J., y Eagle, M. (2013). Experimental Evaluation of Automatic Hint Generation for a Logic Tutor. *International Journal of Artificial Intelligence in Education*, 22(1), 3-17.

VanLehn, K. (1988). Student Modelling. (E. M. Polson, Ed.) *Foundations of Intelligent Tutoring Systems*, 55, 78.

Yeh, S. W., y Lo, J. J. (2005). Assessing metacognitive knowledge in Web-based CALL: a neural network approach. *Computers & Education*, 44 (2), 97-113.

1. Investigadora en el área tecnología educativa. Profesora y escritora. Departamento de Informática. Universidad José Martí Pérez. Ingeniera en Informática. Ivelisse@gmx.com

Revista ESPACIOS. ISSN 0798 1015
Vol. 39 (Nº 10) Año 2018

[Índice]

[En caso de encontrar algún error en este website favor enviar email a webmaster]

©2018. revistaESPACIOS.com • Derechos Reservados