

## *Propuesta de arquitectura de Sistema Tutor Inteligente para desarrollar las habilidades algorítmicas*

**Guillermina Sánchez Román<sup>1</sup>, Josefina Guerrero García<sup>1</sup>, César Alberto Collazos Ordóñez<sup>2</sup>,  
Carolina Tapia Cortes<sup>3</sup>, Daniel Mocencahua Mora<sup>3</sup>**

Facultad de Ciencias de la Computación<sup>1</sup>, Facultad de Ingeniería Electrónica y Telecomunicaciones<sup>2</sup>, Facultad de Ciencias de la Electrónica<sup>3</sup>,  
Benemérita Universidad Autónoma de Puebla<sup>1,3</sup>,  
14 sur y Av. San Claudio, Puebla Puebla.  
Universidad del Cauca, Popayán, Colombia<sup>2</sup>  
{guillesroman, joseguga01,d.mocencahua}@gmail.com  
ccollazo@unicauca.edu.co, carolina\_tapia\_cortes@hotmail.com

**Resumen:** Desde hace varias décadas se ha notado una deficiencia en el aprendizaje de programación, sobre todo en el desarrollo de habilidades algorítmicas. Es importante contemplar la forma en que el estudiante aprende de forma autónoma para mejorar las carencias identificadas. El presente trabajo describe la adaptación de la arquitectura de un Sistema Tutor Inteligente para desarrollar habilidades algorítmicas. La arquitectura formada por los módulos de estudiante, dominio, tutor y de interfaz, ahora integra el aspecto de gamificación basado en el modelo de ChildProgramming. Como resultado se obtuvo la integración de aspectos de gamificación en interrelación con los otros módulos. Se pretende que el sistema determine su conocimiento previo, motive y acompañe al estudiante de nivel superior a desarrollar sus habilidades de pensamiento algorítmico. Lo anterior forma parte previa sobre la creación de un Sistema Tutor Inteligente (STI) que ayude al estudiante a su aprendizaje, en el área de programación.

**Palabras clave:** habilidades del pensamiento algorítmico, Sistemas Tutores Inteligentes, gamificación, estudiante, nivel superior.

**Abstract:** For several decades there has been a deficiency in programming learning, especially in the development of algorithmic skills. It is important to contemplate how the student learns autonomously to improve the identified deficiencies. The present work describes the adaptation of the architecture of an Intelligent Tutor System to develop algorithmic abilities. The architecture integrates student, domain, tutor and interface modules, now integrates the gamification aspect based on the ChildProgramming model. As a result, integration of aspects of gamification in relation to the other modules was obtained. It is hoped to help motivate students during their learning process in architecture, observing their cognitive and practical progress in the process of rewards and feedback to solve the problems. It is intended that the system determine its previous knowledge, motivate and accompany the student of higher level to develop their algorithmic thinking skills. This is a previous part about the creation of an Intelligent Tutor System (STI) to help students learn in the programming area.

**Key words:** algorithmic thinking skills, Intelligent Tutors, gamification, student, upper level, bachelor's degree.

## 1. Introducción

Estudios realizados muestran que las habilidades de los estudiantes universitarios que ingresan a carreras del área de ciencias de la Computación son todavía deficientes. Además, existe ausencia del dominio de los aspectos cognitivos e instrumentales relacionados con éstas habilidades, tanto para su enseñanza y evaluación. Por supuesto, sus habilidades de pensamiento algorítmico no están exentas [Moroni y Señas 05, Robins et al. 03, Rosanigo y Paur 06].

Hoy en día es importante integrar en la educación las tecnologías de la información y comunicación (TIC) que se van generando. El área de ciencias se ha caracterizado por tener poca demanda y a su vez mayor dificultad en el aprendizaje; en particular, en el área de ciencias de la computación los estudiantes tienen dificultades en los cursos iniciales de programación esto puede dar lugar a altas tasas de reprobación [Moroni y Señas 05, Rosanigo y Paur 06].

La programación es una disciplina en Ciencias de la Computación con diversas áreas de aplicación, que implica la resolución de problemas que llevan al estudiante a un reto intelectual, es importante integrar nuevas herramientas que ayuden a potencializar el proceso de aprendizaje. Debido a que los Sistemas Tutores Inteligentes (STI), permiten la emulación de un tutor humano para determinar qué enseñar, cómo enseñar y a quién enseñar [Cataldi y Lage 09].

El presente trabajo describe la arquitectura de un Sistema Tutor Inteligente para desarrollar habilidades algorítmicas, que integra los módulos de estudiante, dominio y tutor, integrando el aspecto de gamificación. Se pretende que el sistema determine su conocimiento previo y acompañe al estudiante en su progreso a través de estrategias de aprendizaje lúdicas que les

motiven y ayuden a desarrollar sus habilidades de pensamiento algorítmico.

En la figura 1, se muestra cuales son las teorías a incluir en cada módulo a fin de identificar como se integrará la arquitectura. Como se puede observar los elementos de entrada para el sistema son las características del estudiante a partir de tres dominios, el razonamiento, estilo de aprendizaje [Felder y Silverman 88] y sus habilidades de pensamiento computacional [Román-González et al. 15]. Integrando las características en los procesos de aprendizaje en la diseño de actividades basados en la parte lúdica del modelo Childprogramming y de acuerdo a las habilidades requeridas para la programación.

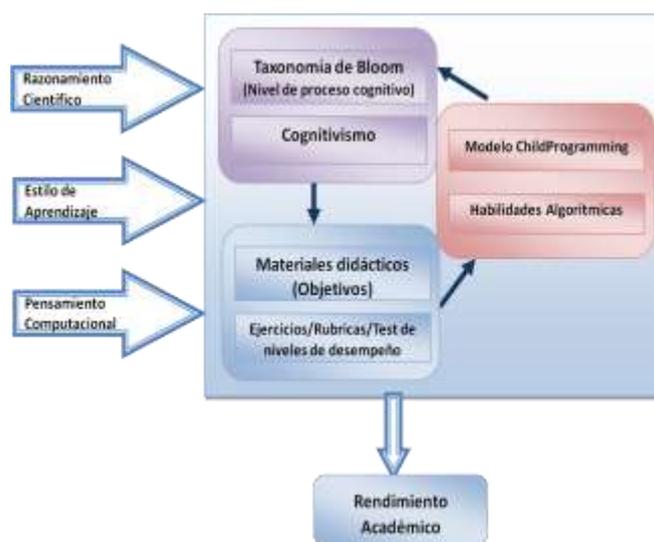


Figura 1: Teorías para el modelo del STI (propio)

La arquitectura propuesta para el STI pretende ayudar a desarrollar las habilidades algorítmicas que tienen los estudiantes de computación. El objetivo es que la integración de estrategias de aprendizaje lúdica integrada en el diseño de un STI influirá en el desarrollo de habilidades algorítmicas para la resolución de problemas en estudiantes de nivel superior.

## 2. Metodología

Para la integración de estrategias de enseñanza-aprendizaje se ha considerado la estrategia lúdica para fomentar el aprendizaje y la motivación. El modelo ChildProgram (figura 2) integra tres dimensiones: el aspecto lúdico, la colaboración y técnicas de agilidad. De este modelo tomaremos la parte de la lúdica que integra mecanismos de gamificación como forma de aprendizaje. La gamificación es definida como un proceso de utilizar el pensamiento y mecánicas de juego para resolver problemas y a su vez atrae e incrementa su motivación y compromiso [Robins et al. 03, Rosanigo y Paur 06].

La metodología para el desarrollo de habilidades de programación básica integra el apoyo de un Sistema Tutor Inteligente (STI) basado en técnicas de Inteligencia Artificial. A partir de la inferencia de los perfiles de estudiantes cada alumno estará en una categoría según su estilo cognitivo propio y el conocimiento previo [1].



**Figura 2:** Modelo ChildPrograming [Orejuela et al. 13]

El STI generará un conjunto de estrategias pedagógicas y didácticas en el módulo tutor que personalizarán el proceso de aprendizaje de acuerdo a las necesidades particulares de cada alumno. Se espera mejorar su aprendizaje y por

supuesto, sus habilidades algorítmicas para la resolución de problemas, evitando la deserción escolar [Rosanigo y Paur, 2006].

Un STI debe ser capaz de comportarse como un experto, tanto en el dominio de conocimiento que enseña (mostrando al alumno cómo aplicar dicho conocimiento), como en el dominio pedagógico (donde es capaz de diagnosticar la situación en la que se encuentra el estudiante y de acuerdo a ello ofrecer una acción o solución que le permita progresar en el aprendizaje) [Carbonell 70].

En la figura 3, se muestra cuales son las tareas a incluir en cada módulo a fin de identificar como se integrará la arquitectura. Se tiene contemplado integrar la metodología propuesta en el diseño de los módulos de un STI para permitir dar seguimiento individual y gradual, mejorando las habilidades del estudiante en la resolución de problemas generando un aprendizaje significativo para que pueda resolver problemas cada vez más complejos de acuerdo a su nivel de avance. Para la elección de los ejercicios de acuerdo al nivel o avance se genera una categorización para que el sistema aprenda de los diferentes comportamientos de entrada y que se trabajarán con técnicas de clasificación ya definidos en Inteligencia Artificial (IA).

El módulo de estudiante debe de integrar los tres aspectos del perfil de estudiante (después de la aplicación de test de razonamiento, estilo de aprendizaje y pensamiento computacional), una vez diagnosticado el perfil se hace una clasificación del estudiante de acuerdo a la definición de clusters con base a los datos obtenidos. Se definirá el material que debe aprender el estudiante.

El módulo de dominio tendrá los objetivos de aprendizaje de acuerdo a la materia del área de computación, los contenidos temáticos, las actividades ya clasificados por niveles y los ejercicios correctivos o ejemplos. Se debe establecer la base del conocimiento de acuerdo a un tema del área de dominio.

En el módulo de tutor se debe de tener la integración de las técnicas de Inteligencia Artificial que analicen el desempeño y cree el plan instruccional personalizado, la forma en que se da la retroalimentación y la modificación del perfil de acuerdo al avance el estudiante. El módulo de tutor se relaciona con el módulo de estudiante para la actualización del perfil del estudiante.

En el módulo de Interfaz de usuario se espera incluir las indicaciones a seguir y las actividades diseñadas con las mecánicas de gamificación al usuario, indicando el trazado del juego, los tipos de jugadores y las recompensas que motiven al estudiante. Por lo tanto, la gamificación se debe de interrelacionar con los módulos de estudiante y tutor a través del módulo de interfaz.

Las últimas investigaciones tratan de integrar los STI en Sistemas de Gestión de Aprendizaje o en plataformas Web y colaborativamente; por tanto, una arquitectura que integra gamificación podría ayudar a utilizar las herramientas de autor para considerar éste proceso de creación de STI en éstos ambientes. El sistema tutor provee a los estudiantes de un curso de control la posibilidad de acceder a servicios que les permite el aprendizaje individualizado, el cual ocurre gracias a la clasificación que se realiza de cada estudiante mediante los estilos de aprendizaje propuestos por Felder y Silverman [Felder y Silverman 88].

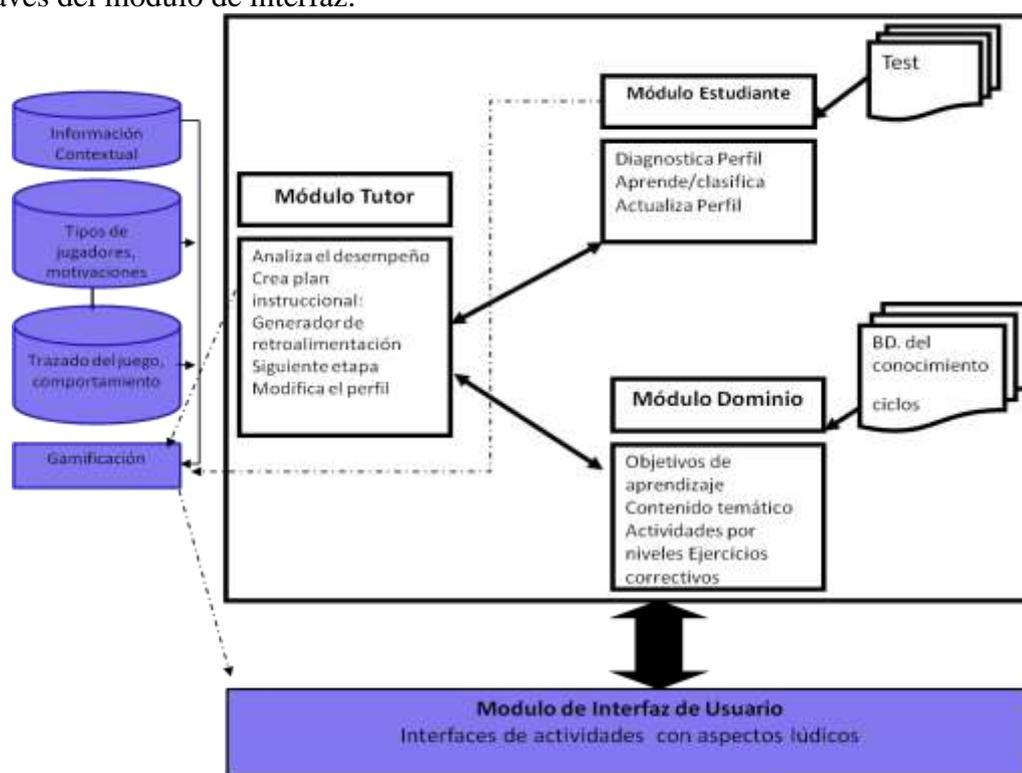


Figura 3: Arquitectura del STI (adaptado de [Carbonell 70, González 16])

### 3. Resultados

Los sistemas Tutores Inteligentes imitan a los

profesores pero con la ventaja de enseñar en la modalidad uno a uno y llevar su proceso de aprendizaje personalizado. La arquitectura previa sólo contemplaba los módulos de dominio,

estudiante y tutor. Sin embargo se considera la didáctica lúdica para el aprendizaje de los estudiantes. De acuerdo a modelos ya propuestos[3], son pocos los que han podido integrar la gamificación en la arquitectura. Se considera que integrar las técnicas de gamificación [Orejuela et al. 13] ayudará a motivar al estudiante durante su proceso de aprendizaje [4], observando su avance cognitivo y práctico en el proceso de recompensas y retroalimentación para dar solución a los problemas.

Se ha integrado en la arquitectura el módulo de interfaz que forma parte del STI para dar una interacción efectiva que aporte en el aprendizaje de los estudiantes a través de la gamificación. En éste módulo se integrarán principios del diseño, implementación y evaluación de sistemas computacionales interactivos para su utilización (HCI: Human Computer Interaction), y los aspectos de gamificación [Orejuela et al. 13] que implican criterios de usabilidad, es decir con alto grado de facilidad en el uso del sistema interactivo [Orejuela et al. 13].

Dada la arquitectura completa (figura 2), se pretende diseñar las actividades que involucren las técnicas de gamificación con base al perfil del estudiante, se pretende que al integrar la gamificación en la arquitectura pueda dar un modelo en el que se prediseñen las actividades lúdicas con base al perfil de cada estudiante.

La orientación de la investigación se centra en proveer una alternativa al estudiante para aprender de forma autónoma, debido a que el docente regularmente no tiene tiempo para atender a los estudiantes en el momento que lo necesitan. Además de la modularidad e independencia, se busca modelar el STI centrado en las necesidades y características reales de los estudiantes. Esto significa integrar los diferentes protocolos pedagógicos que hay para las diferentes áreas y también, las características del estudiante, tanto cognitivas, afectivas y psicológicas, entre otras.

#### **4. Conclusiones**

Con el análisis de las teorías de aprendizaje se puede entender que el aprendizaje surge como: el proceso de adquirir conocimientos y habilidades generando contextos (uso de la tecnología) de interacción a partir de conocimientos previos para llegar a un objetivo. Por lo tanto, la teoría cognitivista soporta la investigación a partir de dos supuestos: conocimientos previos y resolución de problemas. Se pretende saber como el estudiante genera el proceso cognitivo para la resolución del problema, establecer la forma de medir el desarrollo de habilidades algorítmicas y proponer el diseño del tutor para que podamos aportar un prototipo para el desarrollo de las habilidades algorítmicas.

La propuesta de arquitectura define que elementos se deben de integrar en cada módulo, así mismo como se relaciona los aspectos de gamificación en los módulos de interfaz, tutor y de estudiante. La arquitectura da un panorama para continuar con el siguiente proceso que es el prototipo, además de proponer la técnica de Inteligencia Artificial, que por el momento se está proponiendo la clusterización en los perfiles de estudiante y las redes bayesianas para el definir el dominio y estrategias de aprendizaje.

## Agradecimientos

Los autores de este artículo agradecen a la Benemérita Universidad Autónoma de Puebla-México y la Universidad del Cauca-Colombia por el apoyo para realizar el presente trabajo y también se extienden los agradecimientos Consejo Nacional de Ciencia y Tecnología (CONACYT) por el apoyo a la formación de Investigadores.

## Referencias

- [Aparicio y Gutierrez 12] F. A. Aparicio, V. L. F. Gutierérrez, J. L. González, and M. J. L. Isla, “Método de análisis y aplicación de la gamificación,” 2012.
- [Carbonell 70] J. . Carbonell, “AI in CAI: An Artificial-Intelligence Approach to Computer-Assisted Instruction,” *Man-Machine Systems*, IEEE Transactions, vol. 11, no. 4, pp. 190–202, 1970.
- [Cataldi et al. 09] Z. Cataldi, C. Donnamaria, and F. Lage, “Línea de investigación: Sistemas Tutores Inteligentes orientados a la enseñanza para la comprensión,” in *XI Workshop de Investigadores en Ciencias de la Computación*, 2009, pp. 650–654.
- [Cataldi y Lage 09] Z. Cataldi and J. F. Lage, “Sistemas Tutores Inteligentes Orientados a la Enseñanza para la comprensión,” in *EduTec Revista Electrónica de Tecnología*, 2009, no. 28.
- [Felder y Silverman 88] R. M. Felder and L. K. Silverman, “Learning and Teaching Styles,” *Engr. Educ.*, vol. 78, no. 7, pp. 674–681, 1988.
- [González 16] C. S. G. González, “Personalización de la gamificación: Aplicación a sistemas tutores inteligentes,” 2016.
- [Moroni y Señas 05] N. Moroni and P. Señas, “Estrategias para la enseñanza de la programación,” *Primeras Jornadas de Educación en Informática y TICS en Argentina*, Bahía Blanca, Argentina, pp. 254–258, 2005.
- [Orejuela et al. 13] H. F. Orejuela, A. A. García, J. A. Hurtado, and C. Collazos, “Analizando y Aplicando la Gamificación en el Proceso ChildProgramming,” pp. 7–23, 2013.
- [Robins t al. 03] A. Robins, J. Rountree, and N. Rountree, “Learning and Teaching Programming: A Review and Discussion,” *Comput. Sci. Educ.*, vol. 13, no. 2, pp. 137–172, 2003.
- [Román-González et al. 15] M. Román-González, J. C. Pérez-González, and C. Jiménez-Fernández, “Test de Pensamiento Computacional: diseño y psicometría general [ Computational Thinking Test : design & general psychometry ] 2015.
- [Rosanigo y Paur 06] Z. Rosanigo and A. Paur, “Estrategias para la enseñanza de Algorítmica y Programación,” *TE&ET’06*, pp. 117–124, 2006.