Tutores Cognitivos en ACT-R

Angie C. Mogollón Mendoza
amogollonmendoza56@correo.unicordoba.edu.co

Adán Alberto Gómez
aagomez@correo.unicordoba.edu.co

Manuel Fernando Caro
manuelcaro@correo.unicordoba.edu.co

RESUMEN

Los Tutores Cognitivos son un tipo de Sistemas Tutores Inteligentes que se basan en teorías de psicología cognitiva. Los Tutores Cognitivos utilizan arquitecturas cognitivas para analizar cada acción realizada por los estudiantes y calcular las competencias de aprendizaje que necesitan ser medidas. La programación de computadores, es una metodología de resolución de problemas que a menudo causa un conflicto en el estudiante para encontrar la solución adecuada a dicho problema.
En este artículo se presentara estudio general sobre la utilización de Arquitecturas Cognitivas dentro de Tutores cognitivos y su aplicación en el campo de la educación, se hizo una revisión bibliográfica acerca de las Arquitecturas Cognitivas basándose en la memoria y el aprendizaje, y Tutores Cognitivos creados para la enseñanza de la programación y su aplicación en la educación.
Palabras Clave— Tutores Cognitivos, Arquitecturas Cognitivas, Programación, Enseñanza.

ABSTRACT

Cognitive Tutors are a type Intelligent Tutors Systems which are based on cognitive psychology theories. The Cognitive Tutors use cognitive architectures in order to analize each action made by students and to calculate the learning competences that need to be measured. The computers programming, is a methodology of problem resolution that usually causes conficts to the students to find out adecuate answers to this problem.
In this study it is presented a general study about the utilization of Cognitives Architectures inside Cognitives Tutors and its application in the education setting, it was carried out a bibliographic review about the cognitive architecture based on the memory and learning, and Cognitive Tutors created for programming teaching and its application in the education.
Keywords

Cognitive Tutors, Cognitive Architectures, programming, teaching, Cognitive Models.

INTRODUCTION

LOS SISTEMAS tutores inteligentes (STI) permiten que una computadora funcione como un sustituto, dinámicamente adaptativo de un profesor; generando diagnósticos cognitivos que permiten realizar inferencias sobre la base de un modelo de estudiante constantemente reajustado (Reusser, 1993). Los STI se caracterizan por poseer la capacidad de emular a un tutor humano y determinar qué enseñar, cómo enseñar y a quién enseñar (Cataldi & Lage, 2010). Las arquitecturas de los STI se dividen en cuatro módulos que conforman su estructura y permiten su desarrollo: (i) el módulo tutor, cuya función es tomar decisiones acerca de las estrategias de enseñanza que se van a utilizar y las acciones que se deben realizar a través de la interfaz para la comunicación con el aprendiz (Cala, 2014); (ii) el módulo estudiante, el cual categoriza al alumno dentro de los estilos de aprendizaje disponibles dentro del STI (Salgueiro, Costa, Cataldi, García Martinez, & Lage, 2005); (iii) el módulo dominio, que representa el conocimiento del experto en la temática que será enseñada en el sistema y las características de la solución de problemas asociadas a ella (Constanza Raquel Huapaya & Lic. Laura Lanzarini, 2009); y (iv) la interface, que permite la interacción entre el estudiante y el STI de manera eficiente(Cataldi & Lage, 2010).

La incorporación de STI en ambientes educativos permite la adaptación de contenidos y contextos al estilo de aprendizaje de cada estudiante, convirtiéndose en poderosas herramientas adaptativas que apoyan el proceso de enseñanza(Al-Nakhal & Abu Naser, 2017).

Carnegie Learning (Jiménez et al., 2011) y el Departamento de Educación de Estados Unidos, realizaron investigaciones que determinaron la efectividad de los STI con relación a otros enfoques de instrucción, logrando una mejora en el rendimiento académico de más del 85% en el área de matemáticas (Garay & Castro, 2010).

Los Tutores Cognitivos (TC) son un tipo específico de STI que se basan en teorías de psicología cognitiva y que permiten la creación de modelos cognitivos que representan la forma como los estudiantes abordan la solución de problemas (Cala, 2014). Los TC utilizan Arquitecturas Cognitivas para analizar cada una de las acciones realizadas por los estudiantes con el fin de calcular las competencias de aprendizaje que necesitan ser medidas o aprendidas(John R Anderson, 1995).

Una Arquitectura Cognitiva es un modelo computacional cognitivo de dominio genérico que captura la estructura esencial de los procesos de la mente(Sun, 2007). Una de las Arquitecturas cognitivas más conocidas se encuentran: ACT-R(John R. Anderson & Lebiere, 1998)(John R Anderson et al., 2004).

Por otro lado, la programación de computadores, es una metodología de resolución de problemas(Scherer, 2016) que a menudo causa en el estudiante un conflicto cognitivo para encontrar la solución adecuada a dicho problema, ya que por lo general no dispone de estrategias que le permitan responder de manera satisfactoria(Arellano, Fernandez, Rosas, & Zuñiga, 2014). La programación orientada a objetos (POO) es una concepción acerca del proceso de solución de problemas en programación, que considera el programa como una colección de agentes con niveles de autonomía, llamados objetos y por tal razón se constituye en una de las mejores alternativas para implementar sistemas inteligentes(Abdón & Aldana, 2002). El aprendizaje de la POO es difícil, sobre todo para estudiantes universitarios en los primeros semestres(Arévalo & Solano, 2013). A los estudiantes se les hace más complicado aprender fundamentos de POO y desarrollar las habilidades para resolver problemas (Ruiz, Hernández López, & Loaiza Brito, 2015) (Coppo, Iparraguirre, Feres, Ursua, & Cavallo, 2011)(Arellano et al., 2014).

En la Licenciatura en Informática de la Universidad de Córdoba (Colombia), se han venido presentando una serie de dificultades en cuanto al aprendizaje de la programación (Olier, Gómez, & Caro, 2017); los problemas más comunes se están presentando en los estudiantes que están cursando la materia de fundamentos de POO (Barroso & Ortiz, 2007), específicamente en la comprensión de la sintaxis del lenguaje de programación empleado, debido al poco tiempo utilizado en el análisis de los enunciados y a generar posibles soluciones (Schoenfeld, 1985).

El objetivo de este artículo es presentar una revisión general de los tutores cognitivos que se han desarrollado en el campo de la enseñanza de los conceptos básicos de la programación.
Debido a que se han desarrollado pocos TC para la enseñanza de la programación(John R Anderson, Conrad, & Corbett, 1989)(Mercado, Argelio, Reynoso, & Manuel, 2010) (pese a que se han usado ampliamente), el aporte de este artículo es presentar un panorama general sobre los mismos, incluyendo sus características esenciales y las habilidades y componentes del conocimiento que se buscan comprender a través de su uso.

II. ARQUITECTURAS COGNITIVAS

Una Arquitectura Cognitiva es un modelo computacional cognitivo de dominio genérico que captura la estructura esencial de los procesos de la mente (Sun, 2007), especificando la infraestructura subyacente para un sistema inteligente. En pocas

palabras, una AC incluye aquellos aspectos de un agente cognitivo que son constantes en el tiempo y a través de diferentes dominios de aplicación (Langley, Laird, & Rogers, 2006). Es por esto que una AC se constituye en un marco amplio en el que tienen cabida distintos tipos de organización de información y donde esa información está sujeta a modificaciones en su contenido y en su estructuración.(Barros, 2000). Además, este tipo de arquitecturas enmarcan la conducta inteligente de los sistemas dentro de un modelo computacional de espacios - problema. Así, cuando un sistema se enfrenta con una determinada tarea, ésta se convierte en un espacio -problema que se identifica como una meta a resolver (Ruiz Sánchez de León & Fernández Blázquez, 2011).

Al diseñar toda arquitectura cognitiva se tiene en cuenta dos aspectos primordiales para su desarrollo: la memoria y el aprendizaje. La función de la memoria es la de actuar como un repositorio del conocimiento del mundo y de uno mismo, y el aprendizaje cumple el rol de moldear este conocimiento (Duch, Oentaryo, & Pasquier, 2008).Las AC se pueden clasificar de acuerdo a estas características en tres grupos principales (Fig.1): Simbólicos, emergentes e híbrido.

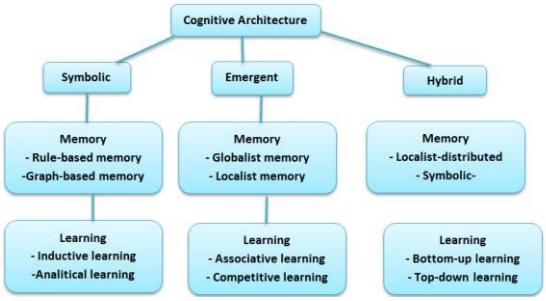


Fig.1 Taxonomía simplificada de las arquitecturas cognitivas. Tomado de(Duch et al., 2008)

Esta sección se centra en las funcionalidades de razonamiento objetivo implementadas en AC populares, para ello se seleccionó la más conocida, la siguiente: ACT-R, organizada de acuerdo a la clasificación anteriormente mencionada. A continuación, la descripción de la AC y una breve explicación acerca de los procesos que la constituyen.

Arquitecturas Cognitivas Híbridas

Las arquitecturas cognitivas híbridas, pueden ser divididas en dos clases basadas en el tipo de memoria de los módulos constituyentes: localista-distribuido y simbólico-conexionista (Sun & Alexandre, 1997). La primera clase de arquitecturas híbridas comprende una combinación de módulos localistas (con cada concepto especificado por un nodo PE) y módulos distribuidos (con cada concepto representado por un conjunto de nodos superpuestos). En comparación, la segunda clase implica una mezcla de módulos simbólicos (es decir, memoria basada en reglas o gráficos) y módulos conexionistas (de tipo localista o distribuido). De manera correspondiente, las arquitecturas híbridas pueden clasificarse en dos clases principales de acuerdo con su dirección de aprendizaje: aprendizaje de arriba hacia abajo y de abajo arriba (Sun & Zhang, 2004).

ACT-R

ACT-R (The Adaptive Control of Thought-Rational) (John R Anderson, Bothell, Lebiere, & Matessa, 1998) es un modelo pensado para explicar la conducta humana, este se basa en la adquisición, almacenamiento y recuperación de información que provee un buen entorno para integrar investigaciones sobre memoria y aprendizaje. La teoría cognitiva en la q se basa ACT-R tiene sus orígenes en la teoría de la memoria humana asociativa (HAM) (John R. Anderson & Bower, 2014), que intenta representar los recuerdos almacenados y cómo esas representaciones medían el comportamiento que se observaba en experimentos de memoria(John R Anderson, 1995). ACT-R cuenta con 5 módulos que tienen como objetivo procesar un tipo de información diferente, los cuales deben ser integrados para producir una cognición coherente.(Oscar Javier Romero Lopez, 2011) En la Fig.5 se muestra una perspectiva general de ACT-R. Basado en Anderson. Las cajas son módulos, controlados por un módulo procesal central a través de búferes de capacidad limitada.



Fig.5 Una perspectiva general de ACT-R. Basado en (Anderson, 2007). Las cajas son módulos, controlados por un módulo procesal central a través de búferes de capacidad limitada.

ACT-R propone un módulo de visión que determina la identidad y la posición de los objetos en el campo visual, un módulo manual que controla las manos, un módulo declarativo que recupera la información de largo plazo, un módulo de metas que actualiza el estado interno del agente, y un sistema basado en reglas de producción que coordina la operación de los otros cuatro módulos.
En ACT-R se producen dos tipos de transferencia de datos: el primero se limita a un tipo de unidad de conocimiento declarativa llamada CHUNK; y el segundo solo puede disparar una única regla en cada ciclo cognitivo; este se compara con las arquitecturas SOAR y EPIC las cuales permiten una activación simultánea de reglas.

III. TUTORES COGNITIVOS

Los departamentos de psicología e informática de la Carnegie Mellon University, iniciaron sus trabajos en tutores cognitivos, basándose en las investigaciones que Anderson estaba desarrollando sobre ACT-R, la cual busca explicar el funcionamiento de la cognición humana(John R Anderson, 2013). La teoría de ACT-R se ha utilizado para construir modelos cognitivos(J. Anderson, Boyle, Corbett, & Lewis, 1990). Un modelo cognitivo es una descripción simplificada de la resolución de problemas humanos en tareas educativas estandarizadas, que ayuda a caracterizar el conocimiento y las habilidades de los estudiantes en diferentes niveles de aprendizaje y facilitar la explicación y predicción del rendimiento académico de los estudiantes(Roberts et al., 2012).

Los STI que utiliza modelos cognitivos se les conoce como Tutores Cognitivos (TC). Los TC utilizan los modelos cognitivos para descomponer la métodos de resolución de problemas en componentes de conocimiento o habilidades, para luego trazar el conocimiento que el estudiante tiene de esa habilidad(Fancsali, Nixon, Vuong, & Ritter, 2013). Los primeros TC creados fueron dirigidos a la enseñanza de la programación y las matemáticas. (John R Anderson et al., 1989)(J R Anderson, Boyle, Farrell, & Reiser, 1984)(J. Anderson et al., 1990).

Los Tutores Cognitivos en Programación han sido ampliamente usados en el mundo académicos(A. Corbett, 2000). Los TC permiten desarrollar ambientes de aprendizaje efectivo. Los estudiantes que trabajan con tutores cognitivos completan actividades de resolución de problemas en un tercio del tiempo que necesitan otros estudiantes que resuelven este mismo tipo de actividades en ambientes convencionales (John R Anderson, Corbett, Koedinger, & Pelletier, 1995). sin embargo han sido pocos los que han sido creados para la enseñanza de la programación (John R Anderson et al., 1989)(Mercado et al., 2010); a continuación describiremos dos tutores cognitivos que son los ampliamente utilizados para la enseñanza de la programación(A. T. Corbett & Anderson, 1995):

APT Programming Tutor

El Tutor de Programación de ACT (APT) es un STI que emplea un modelo cognitivo detallado de conocimientos de programación en un intento por lograr un aprendizaje experto (A. T. Corbett & Anderson, 2008). El objetivo de este tutor es supervisar el conocimiento procedimental del estudiante en la resolución de problemas del área de programación, proveyendo suficientes oportunidades de aprendizaje en el dominio temático del tutor y realizando predicciones con precisión del desempeño del examen de los estudiantes(A. Corbett & Mclaughlin, 2000). Además utiliza el modelado cognitivo del estudiante para incorporar una medida

independiente de su conocimiento declarativo previo afinando así la validez predictiva del proceso de modelado(A. T. Corbett & Bhatnagar, 1997).

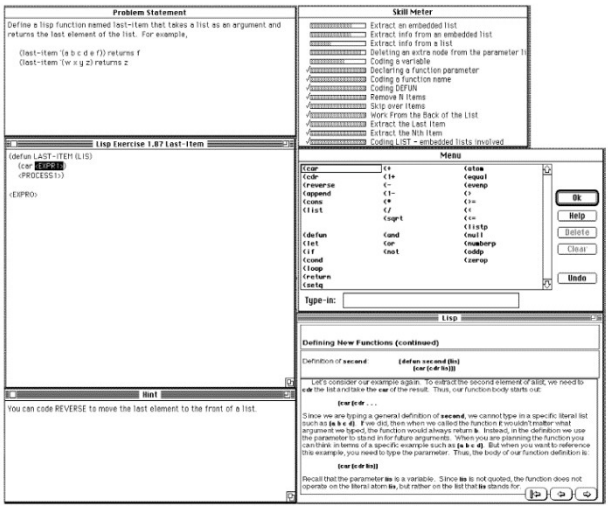


Figura 8. La interfaz APT LISP Tutor.

Así mismo, el tutor realiza un seguimiento del conocimiento procedimental del estudiante a través de los problemas que va resolviendo en un proceso que se denomina rastreo del conocimiento. En este rastreo, el estudiante se representa como una superposición del modelo ideal que posee el tutor (Goldstein, 1979). El medidor de habilidades en la esquina superior derecha de la figura 8 muestra el modelo que posee el tutor del estado de conocimiento del estudiante. Cada entrada en el medidor de habilidades representa una regla de producción en el modelo cognitivo de conocimientos de programación. El sombreado representa la probabilidad de que el estudiante conozca la regla y una marca de verificación indique que el estudiante ha dominado la regla(A. Corbett & Mclaughlin, 2000).

APT programming Tutor ha sido usado ampliamente utilizado para enseñar cursos de programación cada semestre en The Carnegie Mellon University desde 1984.

Mini-Tutores Cognitivos en Plataformas LMS

Se presenta la aplicación de un estudio piloto para la enseñanza de conceptos básico de programación con dos grupos de estudiantes de nivel medio del estado de Aguascalientes, usando prototipos de tutores cognitivos. Se aplicó una evaluación para medir el grado de retención así como pruebas estadísticas para analizar los datos. Los resultados preliminares no muestran diferencias estadísticamente significativas en el rendimiento de ambos grupos. Se argumenta que la ausencia de un instructor humano y la falta de una capacitación presencial previa, entre otras variables, pudo influir en los resultados obtenidos(Mercado et al., 2010).

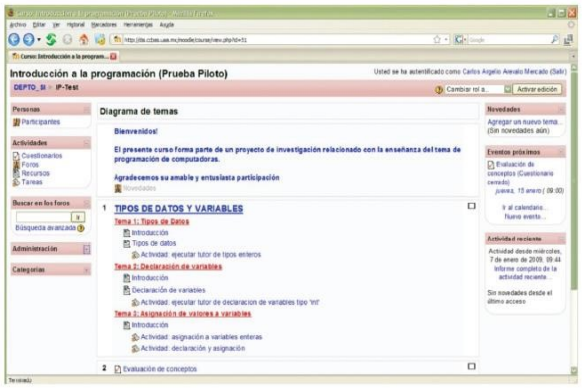


Figura 9. Curso Piloto de la Introducción a la Programación.

En la figura 9 se muestra el despliegue de los mini-tutores que utilizan una plataforma LMS (Learning Management System), en donde se diseñó un curso piloto, simulando el estudio de un tema, en este caso, los tipos de datos y variables enteras en lenguaje C. Se desarrollaron lecciones textuales a manera de explicación teórica, presentadas a los participantes en forma de páginas web estáticas. Los mini-tutores cognitivos, se pusieron a disposición de los alumnos participantes al final de cada lección.

Los Mini-Tutores cognitivos en Plataformas LMS se utilizaron en estudiantes de 5º semestre de la preparatoria Licenciado Jesús Reyes Heroles de Aguascalientes (México) y en estudiantes que cursan entre 1º y 5º semestre en la preparatoria de la Universidad Autónoma de Aguascalientes.

En el área de la programación, los TC que se han desarrollado han sido pocos, específicamente los referenciados en este artículo; sin embargo han sido ampliamente utilizados, ya que ellos permiten la creación de modelos cognitivos que representan la

forma como los estudiantes abordan la solución de problemas(Cala, 2014) desarrollar ambientes de aprendizaje efectivo(A. Corbett & Mclaughlin, 2000).

IV. DISCUSIÓN

El artículo muestra que los TC mencionados están basados en la arquitectura cognitiva ACT-R. APT Programming Tutor esta creado en LISP en concordancia con la Arquitectura Cognitiva ACT-R, esto facilita la realización de ejercicios y la programación de tutores, mientras que los Mini-tutores Cognitivos están creados en PHP e insertados en la plataforma Moodle, esto hace más difícil su comunicación con ACT-R debido a que tiene que usar MIDDLE WARE para la comunicación entre Moodle y ACT-R.
Se observa que los tutores no están hechos para una plataforma específicamente, sino que se puede correr en cualquier sistema operativo.
En cuanto a su utilización, se observó que los TC son aplicados a nivel universitario y a nivel básico, no se encontró la aplicación de estos a nivel de primaria, lo cual puede ser una oportunidad de investigación.

V. CONCLUSIONES

Este trabajo presentó un estudio general sobre la utilización de la Arquitectura Cognitiva ACT-R dentro de Tutores cognitivos y su aplicación en el campo de la educación, se hizo una revisión bibliográfica acerca de las Arquitectura Cognitiva basándose en la memoria y el aprendizaje, y Tutores Cognitivos creados para la enseñanza de la programación y su aplicación en la educación.
La revisión bibliográfica permitió encontrar una de las Arquitecturas Cognitivas más importantes como lo son ACT-R; y los Tutores Cognitivos que han sido utilizados ampliamente para la enseñanza y el aprendizaje y sus características y componentes más esenciales.
Se encontraron 2 tutores cognitivos los cuales se han utilizado a lo largo de los años desde que se desarrollaron; estos tutores se enfocan específicamente en la enseñanza de la programación, ya que se ha evidenciado que pocos son los TC que se crean para esta área.
Se espera seguir trabajando con estos temas, haciendo una profundización acerca de estos y su aplicación no solo en programación, sino también en otras áreas del conocimiento.

VI. REFERENCIAS BIBLIOGRAFICAS

Abdón, I., & Aldana, M. (2002). PREGUNTAS COGNITIVAS Y METACOGNITIVAS EN EL PROCESO DE APRENDIZAJE Influencia de preguntas cognitivas y metacognitivas en comprensi{ó}n conceptual y en habilidad para resolver problemas en ciencias.

Al-Nakhal, M. A. M., & Abu Naser, S. S. (2017). Adaptive Intelligent Tutoring System for Learning Computer TheoryAdaptive Intelligent Tutoring System for Learning Computer Theory. *B+) EUROPEAN ACADEMIC RESEARCH*, *9*(10).

Anderson, J., Boyle, F., Corbett, A., & Lewis, M. (1990). Cognitive Modeling and Intelligent Tutoring. Retrieved from https://www.researchgate.net/profile/Matthew_Lewis23/publication/220545766_Cognitive_Modeling_and_Intelligent_Tutoring/links/56b11a4e08ae56d7b069e9a3.pdf

Anderson, J. R. (1995). A Simple Theory of Complex Cognition.

Anderson, J. R. (2013). *The architecture of cognition*. Psychology Press.

Anderson, J. R., Bothell, D., Byrne, M. D., Douglass, S., Lebiere, C., & Qin, Y. (2004). An integrated theory of the mind. *Psychological Review*, *111*(4), 1036.

Anderson, J. R., Bothell, D., Lebiere, C., & Matessa, M. (1998). An Integrated Theory of List Memory-ent aspects of cognition—memory for serial. *JOURNAL OF MEMORY AND LANGUAGE*, *38*, 341–380. Retrieved from

http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.330.9219&rep=rep1&type=pdf

Anderson, J. R., & Bower, G. H. (2014). *Human Associative Memory.* Taylor and Francis. Retrieved from https://books.google.es/books?hl=es&lr=&id=R0XrAgAAQBAJ&oi=fnd&pg=PP1&dq=human+associative+memory&ots=xXHaz8t65P&sig=hyI-we0CH_QY3a2hgXkaFzwH5og#v=onepage&q=human associative memory&f=false

Anderson, J. R., Boyle, C. F., Farrell, R., & Reiser, B. J. (1984). Cognitive Principles in the Design of Computer Tutors. Retrieved from http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA144825

Anderson, J. R., Conrad, F. G., & Corbett, A. T. (1989). Skill Acquisition and the LISP Tutor. *Cognitive Science*, *13*(4), 467–505. https://doi.org/10.1207/s15516709cog1304_1

Anderson, J. R., Corbett, A. T., Koedinger, K. R., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences*, *4*(2), 167–207.

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Erlbaum.

Arellano, N., Fernandez, J., Rosas, M. V., & Zuñiga, M. E. (2014). Estrategia metodológica de la enseñanza de la programación para la permanencia de los alumnos de primer año de Ingeniería Electrónica, 55–60. Retrieved from http://sedici.unlp.edu.ar/bitstream/handle/10915/39993/Documento_completo.pdf?sequence=1

Arévalo, C., & Solano, L. (2013). Patrones de Comportamiento de Estudiantes de Programación al Utilizar una Herramienta de Visualización de Protocolos Verbales. Retrieved from https://www.researchgate.net/profile/Carlos_Mercado2/publication/303365283_Patrones_de_Comportamiento_de_Estudiantes_de_Programacion_al_Utilizar_una_Herramienta_de_Visualizacion_de_Protocolos_Verbales/links/573e26e208ae298602e6e041/Patrones-de-Comportamie

Barros, G. T. (2000). Un modelo de Arquitectura Cognitiva Aplicación en Robótica Móvil Tesis Doctoral. Retrieved from http://oa.upm.es/641/1/10200011.pdf

Barroso, J. J., & Ortiz, I. R. R. (2007). Dificultades de aprendizaje e intervención psicopedagógica en la resolución de problemas matemáticos, *342*, 257–286. Retrieved from http://www.revistaeducacion.mec.es/re342/re342_13.pdf

Cala, O. (2014). *Estudio comparativo del funcionamiento de Sistemas tutores inteligentes orientados a la enseñanza de los fundamentos de Control Automático*. Universidad Nacional de Colombia.

Cataldi, Z., & Lage, F. J. (2010). Modelado del estudiante en sistemas tutores inteligentes. *TE {&} ET*.

Constanza Raquel Huapaya, & Lic. Laura Lanzarini. (2009). *Sistemas Tutoriales Inteligentes Un analisis crıtico*. Retrieved from http://sedici.unlp.edu.ar/bitstream/handle/10915/4169/Documento_completo.pdf?sequence=1

Coppo, R., Iparraguirre, J., Feres, G., Ursua, G., & Cavallo, A. (2011). Sistema didáctico para la enseñanza de la programación con metodologías de aprendizaje basado en problemas. Retrieved from http://sedici.unlp.edu.ar/bitstream/handle/10915/19923/Documento_completo.pdf?sequence=1

Corbett, A. (2000). Cognitive Mastery Learning in the ACT Programming Tutor. Retrieved from http://www.aaai.org/Papers/Symposia/Spring/2000/SS-00-01/SS00-01-007.pdf

Corbett, A., & Mclaughlin, M. (2000). Modeling

Student Knowledge: Cognitive Tutors in High School and College.

Corbett, A. T., & Anderson, J. R. (1995). Knowledge Tracing: Modeling Acquisition of Procedural Knowledge.

Corbett, A. T., & Anderson, J. R. (2008). Knowledge decomposition and subgoal reification in the ACT programming tutor. *Department of Psychology*, 81.

Corbett, A. T., & Bhatnagar, A. (1997). Student modeling in the ACT programming tutor: Adjusting a procedural learning model with declarative knowledge. In *User modeling* (pp. 243–254).

Duch, W., Oentaryo, R. J., & Pasquier, M. (2008). Cognitive Architectures: Where do we go from here? In *AGI* (Vol. 171, pp. 122–136).

Fancsali, S., Nixon, T., Vuong, A., & Ritter, S. (2013). Simulated Students, Mastery Learning, and Improved Learning Curves for Real-World Cognitive Tutors. Retrieved from https://www.researchgate.net/profile/John_Champaign/publication/260722477_Simulated_Learners/links/02e7e5390d2a7d2fbf000000.pdf#page=15

Garay, I. M., & Castro, J. (2010). *TUTORES INTELIGENTES EN LA ENSEÑANZA DE LA MATEMÁTICA EN SECUNDARIA*. Retrieved from http://funes.uniandes.edu.co/4799/1/MoralesTutoresALME2010.pdf

Goldstein, I. P. (1979). The genetic graph: a representation for the evolution of procedural knowledge. *International Journal of Man-Machine Studies*, *11*(1), 51–77. https://doi.org/10.1016/S0020-7373(79)80005-X

Jiménez, M., Universidad De Costa, C., Rica, R. C., Salas, E., Ogan, A., & Baker, R. S. J. D. (2011). *Tutor Cognitivo y el incremento de aprendizaje en matemática*.

Langley, P., Laird, J. E., & Rogers, S. (2006). Cognitive architectures: Research issues and challenges. *COGNITIVE SYSTEMS RESEARCH*. https://doi.org/10.1016/j.cogsys.2006.07.004

Mercado, A., Argelio, C., Reynoso, G., & Manuel, J. (2010). Un estudio piloto sobre el efecto de los tutores cognitivos para la enseñanza de conceptos básicos de programación. In *Investigación y Ciencia* (Vol. 18, pp. 17–29). Retrieved from http://redalyc.uaemex.mx/src/inicio/ArtPdfRed.jsp?iCve=67413203004

Olier, A. ., Gómez, A. A., & Caro, M. F. (2017). Design and Implementation of a Teaching Tool for Introduction to Object Oriented Programming, *15*. Retrieved from http://www.ewh.ieee.org/reg/9/etrans/ieee/issues/vol15/vol15issue01Jan.2017/15TLA1_13OlierQuiceno.pdf

Oscar Javier Romero Lopez. (2011). Arquitectura Hıbrida para Sistemas Cognitivos Artificiales con Comportamiento Emergente, Adaptativo y Auto-organizado. Retrieved from http://dw.crackmypdf.com/0059959001489075171/romero_oscar_tesis.pdf

Reusser, K. (1993). Tutoring systems and pedagogical theory: Representational tools for understanding, planning, and reflection in problem solving. *Computers as Cognitive Tools*, *1*, 143–177.

Roberts, M. R., Alves, C. B., Chu, M.-W., Thompson, M., Bahry, L. M., & Gotzmann, A. (2012). Testing Expert-Based vs. Student-Based Cognitive Models for a Grade 3 Diagnostic Mathematics Assessment.

Ruiz, J. G., Hernández López, M., & Loaiza Brito, J. A. (2015). REVISTA ELECTRÓNICA ANFEI DIGITAL PENSAMIENTO SISTÉMICO Y DESARROLLO DE COMPETENCIAS, EN EL APRENDIZAJE DE LOS LENGUAJES DE PROGRAMACIÓN. *Año 1*, (2). Retrieved from http://www.anfei.org.mx/revista/index.php/revista/article/viewFile/108/443

Ruiz Sánchez de León, J. M., & Fernández Blázquez, M. A. (2011). Arquitecturas cognitivas y cerebro: hacia una teoría unificada de la cognición. *International Journal of Psychological Research, ISSN-E 2011-2079, Vol. 4, Nº. 2, 2011, Págs. 38-47*, *4*(2), 38–47. Retrieved from https://dialnet.unirioja.es/servlet/articulo?codigo=3904273

Salgueiro, F. A., Costa, G., Cataldi, Z., García Martinez, R., & Lage, F. J. (2005). Sistemas inteligentes para el modelado del tutor. *Proc. GCETE, Brazil*, 63.

Scherer, R. (2016). Learning from the Past-The Need for Empirical Evidence on the Transfer Effects of Computer Programming Skills. *Frontiers in Psychology*, *7*, 1390. https://doi.org/10.3389/fpsyg.2016.01390

Schoenfeld, A. H. (1985). *Mathematical problem solving*. Academic Press.

Sun, R. (2007). The importance of cognitive architectures: An analysis based on CLARION. *Journal of Experimental {&} Theoretical Artificial Intelligence*, *19*(2), 159–193.

Sun, R., & Alexandre, F. (1997). *Connectionist-symbolic integration : from unified to hybrid approaches*. Lawrence Erlbaum Associates. Retrieved from https://books.google.es/books?hl=es&lr=&id=FBlga7dU-IsC&oi=fnd&pg=PP1&dq=Connectionist+symbolic+integration&ots=QBgNH_siNh&sig=7AzOqvXj8RH1AlhZ-Dz_I29H9rE#v=onepage&q=Connectionist symbolic integration&f=false

Sun, R., & Zhang, X. (2004). Top-down versus bottom-up learning in cognitive skill acquisition. *Cognitive Systems Research*, *5*(1), 63–89.

Para citar este artículo / to cite this article: Mogollón, A., Gómez, A., Caro, M. (2017). Motores Cognitivos en ACT-R. Acta Scientiæ Informaticæ 1(1), 14-22.