# Sistema tutor afectivo para la enseñanza de lógica algorítmica y programación

Ramón Zatarain-Cabada[1], María Lucia Barrón-Estrada[1], José Mario Ríos-Félix[1], Giner Alor-Hernandez[2]

[1]Instituto Tecnológico de Culiacán, Culiacán Sinaloa, México

[2] Instituto Tecnológico de Orizaba, División de Estudios de Posgrado e Investigación, Orizaba, Veracruz, México

{rzatarain, lbarron, mario_rios}@itculiacan.edu.mx, galor@itorizaba.edu.mx

**Resumen.** La creciente demanda de herramientas de software que motiven y apoyen a los estudiantes en el aprendizaje de diseño e implementación de algoritmos y programas, ha motivado la creación de este tipo de sistemas de software. En este artículo presentamos un nuevo e innovador sistema tutor afectivo de lógica algorítmica y programación, basado en la técnica de bloques. Nuestro enfoque combina la interfaz de Google Blockly con técnicas de gamificación y ejercicios que son monitoreados para obtener el estado afectivo del estudiante. Dependiendo de la emoción manifestada (aburrido, enganchado, frustrado y neutral), el sistema evalúa una serie de variables, para determinar si el estudiante requiere asistencia. En base a las pruebas preliminares con varios estudiantes, hemos obtenido resultados satisfactorios, que indican que nuestro sistema contribuye de forma adecuada al proceso de enseñanza de lógica algorítmica.

**Palabras clave:** Computación afectiva, sistemas tutores inteligentes, lógica algorítmica, lenguajes de programación.

# Affective Tutoring System for Teaching Algorithmic Logic and Programming

**Abstract.** The growing demand for software tools that encourage and support students in learning design and algorithm implementation, has allowed the creation of such software systems . In this paper we present a new and innovative affective tutoring system, for logic and algorithmic programming, based on block techniques. Our approach combines the Google Blockly's interface with gamification techniques and exercises that are monitored according to the emotional state of the student. Depending on the expressed emotion (boring, engagement, frustration and neutral), the system evaluates a number of variables to determine whether the student requires assistance. Based on preliminary tests

with several students, we have obtained satisfactory results, which indicate that our system properly contributes to the process of learning algorithm logic.

**Keywords:** Affective computing, intelligent tutoring systems, algorithmic logic, programming languages.

## 1. Introducción

Aprender a programar sigue siendo una de las tareas más demandantes para la mayoría de los estudiantes. Cada año se imparten cursos universitarios sobre programación básica a nuevas generaciones de estudiantes. Incluso carreras ajenas a las ciencias de la computación incluyen en su plan de estudios materias de programación básica.

En la actualidad existen muchas herramientas, que pretenden apoyar a los estudiantes, en su ardua tarea de aprender temas relacionados con la computación, debido a que la ciencia de la computación no es sólo acerca de la programación, sino toda una forma de pensar [1]. Podemos clasificar estas herramientas según su enfoque. Algunas de estas herramientas de software son los sistemas llamados e-learning o CAI (Computer Assisted Instruction), los sistemas tutores inteligentes (STI) de programación y los juegos serios que instruyen los temas de programación.

Las herramientas o sistemas e-learning de apoyo para la programación pretenden facilitar el aprendizaje, haciendo uso de marcos de trabajo pre-establecidos, que les facilitan la implementación de los algoritmos y programas a los estudiantes. Dentro de esta clasificación, existen las herramientas de autor, las cuales permiten a los usuarios diseñar y construir sus propias historias interactivas, animaciones, simulaciones o incluso juegos. Dentro de las herramientas de autor, destacan aquellas que pretenden instruir a los estudiantes en lo que se conoce como "Pensamiento Computacional". El Pensamiento Computacional es una serie de habilidades que integran la resolución de problemas, el diseño de sistemas y la comprensión de la conducta humana que se basa en los conceptos fundamentales de la computación [2].

Por otra parte, los STI son ambientes computarizados de aprendizaje que incorporan los modelos computacionales de las ciencias cognitivas, ciencias de aprendizaje, lingüística computacional, inteligencia artificial entre otros campos. Un STI brinda seguimiento de los estados psicológicos de los alumnos en los detalles finos, un proceso llamado modelado de los estudiantes [3]. Dentro de la familia de los STI, existen los Sistemas Tutores Afectivos (STA), quienes son sistemas inteligentes que incorporan la capacidad de reconocer el estado emocional de los estudiantes, con lo que permiten al usuario interactuar con ejercicios que estimulen su estado emocional.

Finalmente, los juegos serios son aquellos juegos que más que entretener, pretenden enseñar a través de un juego. De esta forma atraen y retienen a los estudiantes más fácilmente que otras herramientas de enseñanza [4]. De la idea de los juegos, surge lo que se conoce como gamificación. Básicamente, la gamificación es un término que se utiliza para describir aquellas características de un sistema interactivo que tienen como objetivo motivar y comprometer a los usuarios finales a través del uso y la mecánica de estímulos comúnmente incluidos en los juegos, como por ejemplo, los trofeos, niveles y las tablas de posiciones [5].

EasyLogic es una herramienta que combina las tecnologías previamente descritas (herramientas e-learning para enseñanza de programación, sistemas tutores inteligentes y afectivos y gamificación) que permite a un estudiante diseñar e implementar algoritmos de una forma simple pero a su vez poderosa. La principal contribución de la herramienta y de este trabajo, es que incluye aspectos afectivos y motivacionales por primera vez dentro de una herramienta que enseña a un estudiante el difícil proceso de construcción de algoritmos y programas.

Este artículo está organizado de la siguiente forma: la Sección 2 describe los trabajos relacionados. La Sección 3 presenta la estructura y funcionamiento de la herramienta (EasyLogic). En la Sección 4 se muestra una sesión de trabajo con EasyLogic. La Sección 5 contiene algunos de los experimentos realizados y los resultados obtenidos. Por ultimo en la Sección 6 se describen las conclusiones y trabajos futuros.

## 2. Trabajos relacionados

En esta sección se describen investigaciones y trabajos en áreas relacionadas con nuestra investigación.

### 2.1. Herramientas e-learning de programación

Dentro de las herramientas de programación más importantes podemos mencionar a GreenFoot [6], un ambiente de desarrollo integrado, que combina la programación en Java con una interfaz gráfica interactiva. Greenfoot se enfoca a facilitar la enseñanza de temas de programación orientada a objetos. Otra herramienta importante es Alice [7], en donde los estudiantes novatos pueden programar juegos en 3D utilizando modelos de objetos ya predefinidos.

Por otra parte, en los últimos años se han desarrollado herramientas que permiten aprender programación para móviles, como es el caso de MIT App Inventor [8], que permite diseñar y construir aplicaciones móviles para Android totalmente funcionales. Con esta herramienta los desarrolladores se enfocan en la lógica y el algoritmo de solución, en lugar de la sintaxis del lenguaje de programación.

### 2.2. Herramientas de autor

Algunas de las herramientas de autor que abordan conceptos de pensamiento algorítmico y programación básica son: Scratch [9], [10], una comunidad de aprendizaje creativo en línea, donde los usuarios pueden interactuar y subir sus proyectos para favorecer el reúso y permitir la remezcla de ellos. Para crear un proyecto se emplean bloques gráficos, donde cada bloque representa un elemento del lenguaje de programación, como son: estructura de control, operadores, variables, funciones, etc.

Otro ejemplo es Scalable Game Design Arcade (AgentSheets) [1], [11], que permite crear juegos de manera fácil, para lo cual utiliza agentes pedagógicos, empleando técnicas de gamificación. También, es importante mencionar a Code Studio [12], el cual se presenta mediante una página principal con los cursos en línea creados por

Code.org, y que representa la comunidad donde un estudiante puede aprender diferentes temas de ciencias de la computación.

Por último, PseInt [13]es una herramienta para asistir a un estudiante mediante un simple e intuitivo pseudo-lenguaje en español, complementado con un editor de diagramas de flujo que permite centrar la atención en los conceptos fundamentales de lógica de programación y al uso de estructuras de control.

### 2.3. Sistemas tutores inteligentes y afectivos

Nuestro principal interés son aquellos tutores inteligentes que realizan una intervención dependiendo del estado afectivo del estudiante.

Entre los STI afectivos más novedosos se encuentran Gaze tutor, un tutor inteligente para el área de biología que mediante un rastreo de los ojos de los estudiantes detecta si ellos se encuentran aburridos, enganchados/motivados o distraídos. El tutor utiliza diálogos textuales a través de un agente pedagógico para tratar de reorientar la atención del estudiante [14]. Por otra parte Gerda es un ITS para temas de sistemas operativos, el cual mediante diálogos (preguntas y respuestas), analiza el estado afectivo del estudiante utilizando diccionarios de palabras con anotaciones afectivas [15].

Tal vez el más importante de los STI afectivos es Affective AutoTutor, una extensión de AutoTutor (un ITS que ayuda a los estudiantes a aprender contenido técnico complejo en diferentes áreas del conocimiento), que agrega capacidades de reconocimiento y manejo de emociones mediante la detección del estado afectivo del estudiante, utilizando expresiones faciales, sensores de movimiento del cuerpo y conversación textual [16].

### 2.4. Juegos serios

Dos de los juegos más interesantes son Program your robot [17] y Cargo-bot [18]. El primero se enfoca a la enseñanza de pensamiento algorítmico y el segundo plantea una nueva metodología de enseñanza de la técnica de recursión a través de un juego. Ambos son juegos muy bien logrados, que atrapan a los usuarios, quienes no se dan cuenta que están aprendiendo mientras juegan.

### 2.5. Panorama actual

Una de las principales debilidades de esta clase de herramientas enfocadas al aprendizaje de temas relacionados con el pensamiento computacional, y en específico lógica algorítmica, es que solo se orientan a estudiantes de nivel pre-universitario (primaria y secundaria principalmente), y no contemplan el estado afectivo de los estudiantes cuando estos las usan. Immordino-Yang and Damasio establecieron que los procesos emocionales actúan como un "timón" en la transferencia de conocimientos y habilidades, y que las emociones juegan un papel importante y necesario en la toma de decisiones, ya que es la interfaz entre la cognición que conduce a desarrollar potencialmente la creatividad [19].

Además, como establecen J. Maloney y M. Kölling, acerca de Scratch y Greenfoot, la filosofía que utilizan para enseñar es que al estudiante primero debes dejarlo jugar,

permitirle crear algo, motivarlo a ser creativo y luego explicarle lo que realmente está pasando. Ellos concluyen que una persona que utilice Scratch puede continuar su enriquecimiento de habilidades por medio de Greenfoot y luego migrar hacia algún IDE (Scratch → Greenfoot → IDE) [20].

Hasta la fecha, no existe un sistema integral, con el que el estudiante pueda aprender y dominar los elementos de lógica algorítmica y programación de forma progresiva, y que además tome en cuenta el estado afectivo del mismo estudiante. Para poder manejar este gran reto, la herramienta deberá manejar 3 etapas:

1. Fungir como tutor para desarrollar un pensamiento computacional.
2. Permitir practicar los conocimientos de lógica algorítmica.
3. Guiar al estudiante hacia una transición a un lenguaje de programación.

La contribución principal de EasyLogic, es contar con un STI para la enseñanza de lógica algorítmica a nivel universitario que además de utilizar técnicas de gamificación, permita reconocer y manejar el estado afectivo de los estudiantes, como apoyo a elementos cognitivos usados en la individualización del proceso de enseñanza/aprendizaje de los estudiantes.

## 3. EasyLogic

EasyLogic es un sistema web, que actúa como STI afectivo en el proceso de aprendizaje de programación de un lenguaje. Nuestro sistema se conforma de 3 secciones principales: a) Aprende. Se proveen diversos cursos, que incluyen una serie de ejercicios, para dar lección a cada una de las distintas estructuras de control que existen. Estos ejercicios son supervisados para obtener el estado afectivo que el estudiante está presentando. b) Imagina y crea. Esta sección permite a los estudiantes diseñar sus propios algoritmos y posteriormente ejecutarlos. Asimismo puede generar código en un lenguaje de programación asociado al algoritmo elaborado. c) Codifica. En esta sección se puede programar directamente en JavaScript en lugar de usar los bloques gráficos (aún en desarrollo).

Al igual que muchas herramientas de autor y de programación modernas, EasyLogic utiliza bloques gráficos. Nuestra interfaz se logra utilizando la librería de Google Blockly [21]. Blockly es una librería de JavaScript que además de ser libre, es personalizable y extensible, por lo que se crearon nuevos bloques adecuados para los diversos ejercicios.

### 3.1. PREMOC

Para el reconocimiento emocional se utilizó una Plataforma de Reconocimiento Multimodal de Emociones (PREMOC), la cual se está desarrollando en el laboratorio de Maestría en Ciencias de la Computación del Instituto Tecnológico de Culiacán. PREMOC Consiste en un servicio web que puede utilizar: imágenes faciales, audio, texto y señales electroencefalografías (EEG), las cuales procesa para brindar como resultado una o varias emociones, dependiendo del modo de integración a utilizar.

PREMOC cuenta con 3 modos de integración: Simple, Multi y Student. En el modo Simple, la respuesta enviada, es la emoción detectada por cada reconocedor

individualmente, sin integrarlos. Por ejemplo, en caso de enviar datos de una imagen facial, y datos de señales EEG, la respuesta serán 2 emociones independientes, una para la imagen facial y otra para la señal EEG. En el modo Multi, la respuesta al usuario es multimodal, es decir, envía una sola emoción al usuario, integrando todas las emociones detectadas por cada reconocedor. Esto se logra utilizando un integrador que se implementó en lógica difusa. En el modo Student, a diferencia de los modos de integración anteriores, que utilizan las emociones de Ekman, el usuario obtendrá una respuesta multimodal orientada a la educación (enganchado, aburrido, frustrado y neutral). La Figura 1 muestra la estructura y funcionamiento de PREMOC.

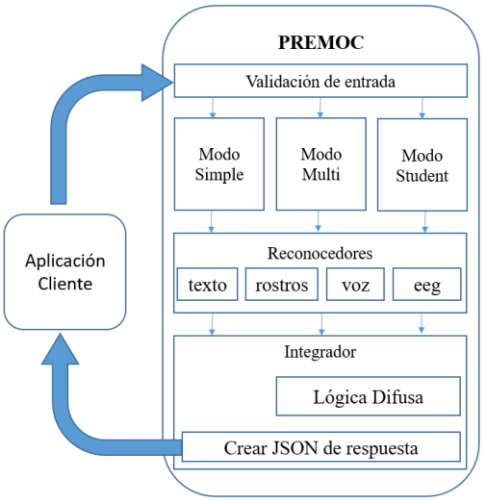
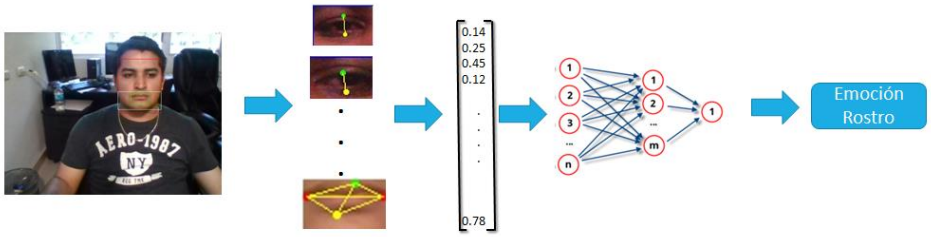



**Fig. 1.** Estructura de PREMOC.

Para la versión actual del sistema, se utiliza el modo Student, y se envía información de imágenes faciales. El reconocedor de imágenes de PREMOC consiste en dos fases: el Entrenamiento y la Ejecución. La fase de entrenamiento consistió en entrenar la red neuronal encargada de la clasificación de imágenes. Para esto se tomaron como datos de entrenamiento las imágenes del corpus RaFD [22], del cual se utilizaron 955 imágenes. De cada una de las imágenes se extrajo un vector de 10 características, donde cada característica es una distancia euclidiana entre diferentes puntos del rostro. Con estos datos se entrenó una red neuronal de propagación hacia atrás, empleando la




librería Weka para Java. La fase de ejecución se describe en la Figura 2.

**Fig. 2**. Algoritmo reconocedor de emociones, por medio de imagen facial.

### 3.2. Ayudas

Una vez que el usuario procede a realizar un ejercicio, el sistema comienza a obtener imágenes del usuario, para ser enviadas de forma continua a PREMOC. Luego de ser analizadas las imágenes, se obtiene una emoción (aburrido, enganchado, confundido, neutral); posteriormente se almacena en el sistema la emoción presentada por el usuario. Cada 30 segundos, EasyLogic evalúa diversas variables como el tiempo invertido en el ejercicio, cantidad de ejecuciones y la emoción que más se ha detectado recientemente por el usuario, para determinar si el estudiante necesita de algún tipo de ayuda.

Las ayudas consisten en ventanas emergentes que se muestran automáticamente cuando se detecta que el usuario las necesita. Las ayudas se dividen en tres tipos: iniciales, informativas y motivacionales. En la Figura 3 se describen cada una de las ayudas utilizadas por EasyLogic.

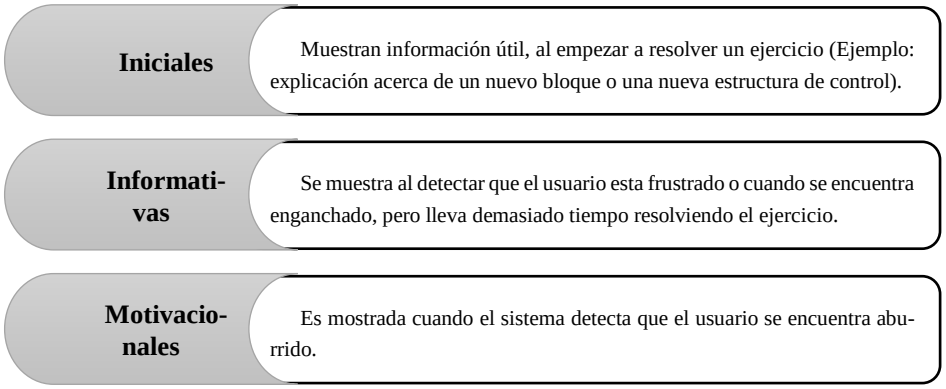



**Fig. 3.** Tipos de ayudas utilizadas en EasyLogic.

EasyLogic implementa técnicas de gamificación para motivar a los estudiantes, otorgándoles puntos y trofeos conforme van avanzando en su curso. Debido a que la herramienta pretende usarse en cursos reales, los estudiantes deberán seleccionar el grupo al que pertenecen al momento de registrarse. Cada estudiante podrá visualizar el puntaje que cada uno de sus compañeros tiene hasta el momento y los trofeos que han logrado obtener. Se pretende con esto estimular a que sean competitivos y se enganchen en la herramienta.

## 4. Una sesión de trabajo con EasyLogic

El sistema inicia en una pantalla de bienvenida, donde se muestran información sobre los cursos, acceso al área libre para la creación de algoritmos y acceso a la sección de programación. Para acceder a los cursos, los usuarios deberán iniciar sesión utilizando su correo electrónico y contraseña. Una vez seleccionado el curso el sistema cargará automáticamente el siguiente ejercicio aún sin resolver por el estudiante. Algunos ejercicios están asociados a ayudas iniciales; una vez cargado el ejercicio se

muestra la ayuda inicial en caso de requerirse. Para no aburrir al estudiante con ayudas que ya ha leído anteriormente, se guarda información de todas aquellas ayudas que se le han mostrado al estudiante y se ha confirmado que la información ha sido leída.

Una vez cargado el ejercicio, el sistema deja trabajar al estudiante por un tiempo específico. En caso de resolver el ejercicio se calculan los puntos obtenidos en base al tiempo invertido en el ejercicio y a la cantidad de ejecuciones requeridas. En caso de obtener un nuevo trofeo se muestra al estudiante. En caso de no haber resuelto el ejercicio, se analizan los datos del usuario (tiempo transcurrido, estado emocional, cantidad de ejecuciones) para verificar si el usuario necesita algún tipo de ayuda extra (informativa o motivacional). De ser así se obtiene de una base de datos una ayuda para el ejercicio actual. En la Figura 4 se muestra el proceso descrito anteriormente.

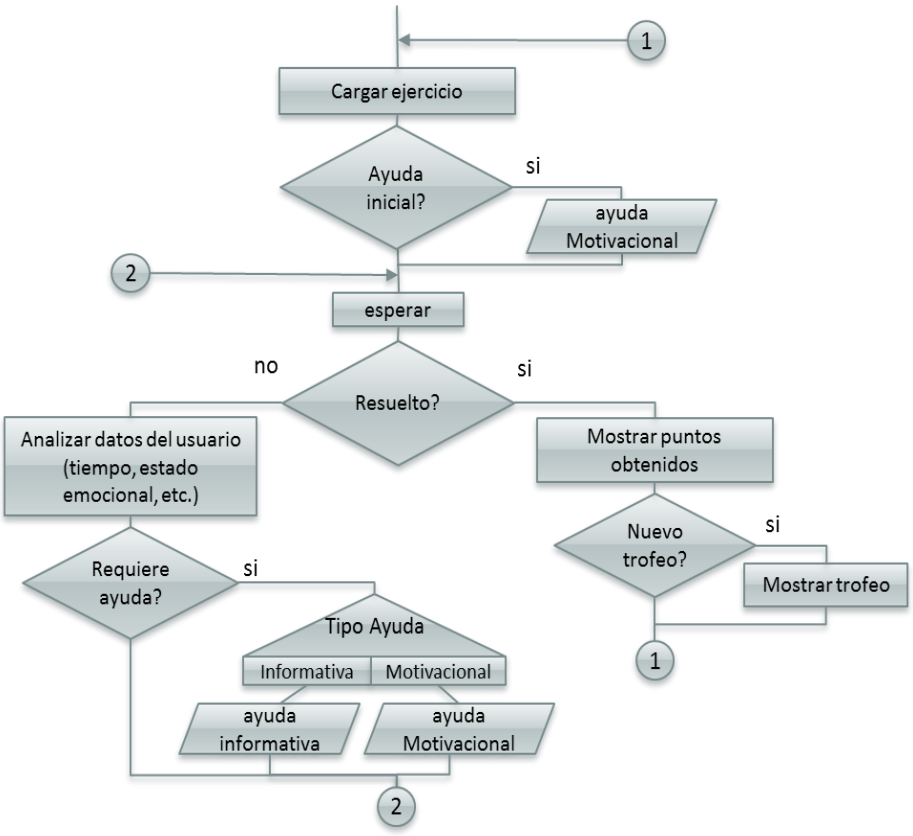



**Fig. 4**. Diagrama de flujo que muestra el proceso que siguen los cursos de EasyLogic.

La interfaz principal para resolver los ejercicios se muestra en la Figura 5, la cual consiste en una barra de opciones que re direccionan hacia otras pantallas (1), el nombre del curso que se está realizando (2), el nivel del ejercicio actual (3), la clasificación de los bloques gráficos (4), un área para crear el algoritmo por medio de los bloques (5), el juego que se animará una vez ejecutado el algoritmo (6), la cantidad de bloques

disponibles (7), un botón para ejecutar el algoritmo y otro para ver el código asociado al algoritmo creado (8).

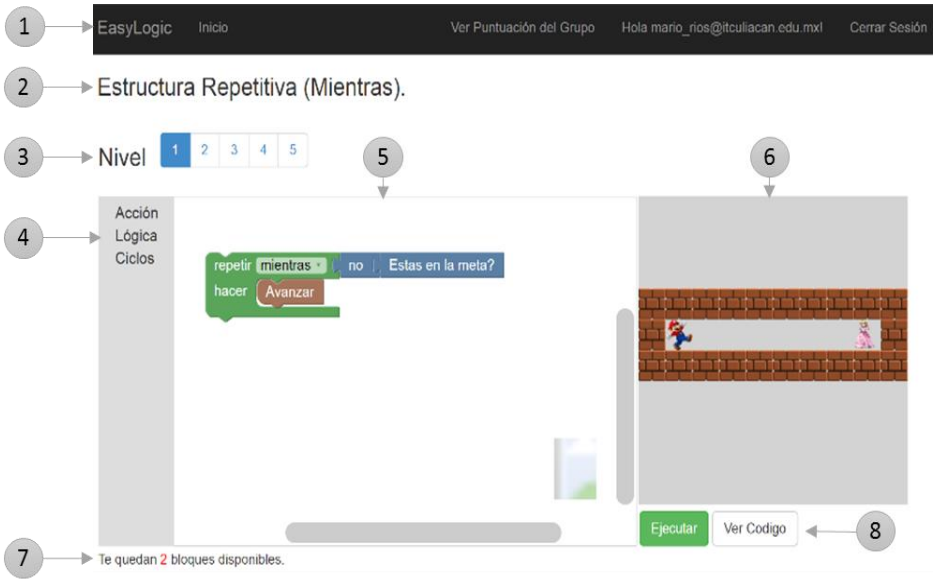



**Fig. 5.** Interfaz para la solución de ejercicios.

La Figura 6 muestra un ejemplo de una ayuda informativa acerca de la estructura repetitiva Desde/Hasta (For).

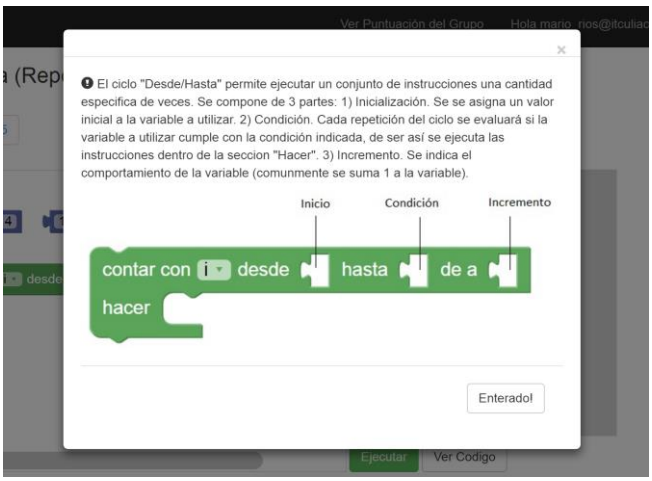



**Fig. 6.** Ejemplo de ayuda Informativa.

En la sección libre se pueden crear algoritmos mediante bloques. Está enfocado principalmente para resolver ejercicios comúnmente utilizados por maestros en las aulas de clase. Una vez elaborado el algoritmo, los estudiantes pueden visualizar el código asociado en JavaScript. En la Figura 7 se muestra un ejemplo de un algoritmo,

donde al ejecutarlo, se tendrá que introducir un número por teclado y el programa mostrara un mensaje de texto que indicará si eres mayor o menor de edad.

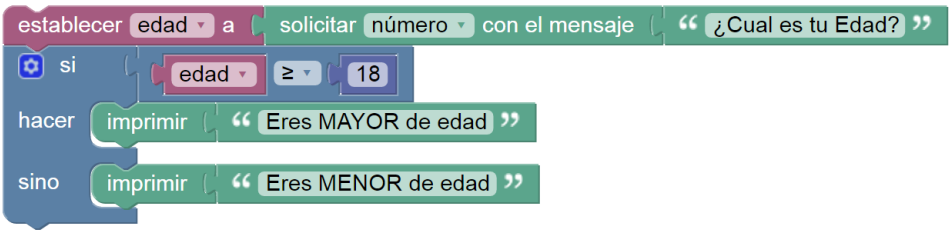



**Fig. 7** - Ejemplo de algoritmo creado en EasyLogic.

## 5. Experimentos y resultados

Hasta la fecha hemos realizado una serie de pruebas preliminares con 10 estudiantes del Instituto Tecnológico de Culiacán. Los sujetos de prueba realizaron 2 algoritmos en la sección libre y completaron todos los cursos que actualmente están disponibles en EasyLogic (Estructuras: Secuencial, Condicional, Repetitiva While y Repetitiva For). Al finalizar los ejercicios, se encuestaron a los sujetos de prueba utilizando la escala de Likert. La Figura 8 muestra los resultados de dos de las preguntas más importantes de la encuesta (acerca de la usabilidad y de las ayudas de la herramienta):

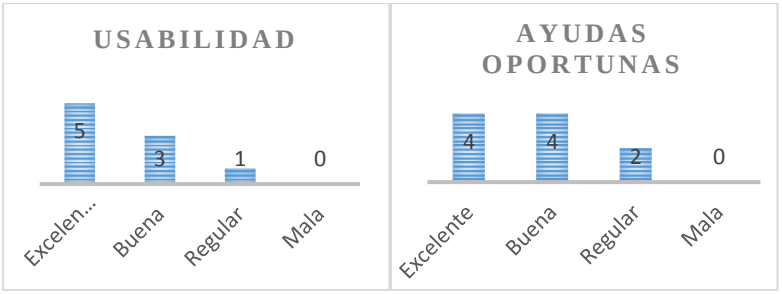



**Fig. 8.** Resultados de 2 preguntas de las encuestas.

La mayoría de los sujetos de prueba concordaron en que las ayudas mostradas por el sistema han sido oportunas y contribuyen de forma adecuada a la realización de los ejercicios. Otro dato interesante es que los usuarios piensan que el sistema es muy amigable y fácil de usar, incluso sin conocimientos previos del sistema.

Para la realización de los ejercicios, EasyLogic combina 3 aspectos importantes: Afecto, la Herramienta y la Gamificación. La Figura 9 muestra los resultados generados por un estudiante, al momento de realizar el ejercicio 5 del curso "Estructura Selectiva". La parte Afectiva (A) evaluó el estado emocional del estudiante utilizando PREMOC, al cual se le envía una imagen cada 15 segundos. Se registró que el estudiante sobre todo se encontraba aburrido, por lo que procedió a mostrar una Ayuda Motivacional, debido a que el usuario empleo mucho tiempo para resolver el ejercicio y se había detectado 3 veces la emoción de enganchado, a los 120 segundos se mostró una Ayuda

Informativa. La Herramienta (B), mostró el juego a resolver y proporcionó los bloques adecuados para dar solución al problema. La parte de Gamificacion (C), calculó un total de 73 puntos de un máximo de 100, debido a que el usuario tardó bastante tiempo (168 segundos) y además requirió de 2 ejecuciones del algoritmo. Para este ejercicio no se obtuvo ningún trofeo nuevo, debido a que necesitaba 500 puntos para obtener el siguiente trofeo. Opcionalmente el estudiante tenía acceso a la tabla de posición donde se muestra el puntaje actual de todos los integrantes del experimento.

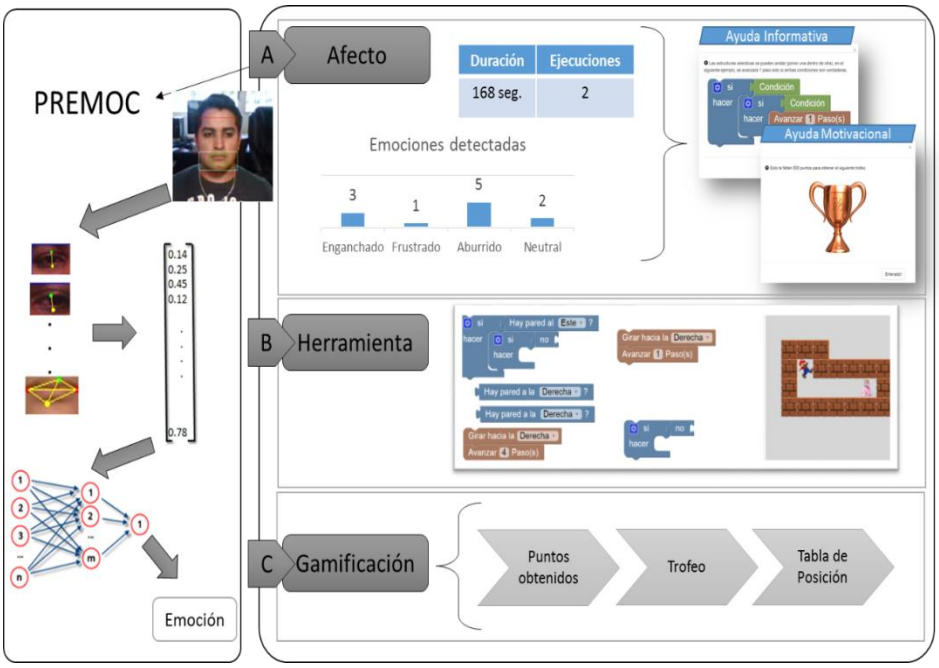



**Fig. 9**. Ejemplo de experimento Ejercicio 5 del curso "Estructura Selectiva".

## 6. Conclusiones y trabajos futuros

Aprender a programar sigue siendo una tarea demandante para los estudiantes. En este artículo, presentamos EasyLogic, un STI Afectivo para el aprendizaje de lógica algorítmica y programación. Las pruebas preliminares realizadas, han demostrado que la detección del estado emocional del estudiante, influye favorablemente al ser utilizadas para realizar una intervención, en caso de que el estudiante lo necesite.

Como trabajo futuro se planea realizar experimentos con dos grupos de estudiantes que actualmente cursan la materia de Algoritmos y Lenguajes de Programación, en el Instituto Tecnológico de Culiacán. Se planea modificar la herramienta para desactivar el reconocimiento emocional y experimentar con esta versión, con uno de los grupos de prueba. De este modo los estudiantes no recibirán ayudas informativas, ni motivacionales por parte de la herramienta, solo se mostrarán ayudas iniciales. De tal manera, se permitirá corroborar si la parte afectiva del tutor, implica realmente una mejora en el aprendizaje de los estudiantes.

## Referencias


1. Ioannidou, A., Bennett, V., Repenning, A., Koh, K.H., Basawapatna, A.: Computational Thinking Patterns. Online Submiss, Vol. 2 (2011)
2. Wing, J.: Computational thinking. J. Comput. Sci. Coll., Vol. 24, No. 6, pp. 6–7 (2011)
3. Graesser, A.C., Conley, M.W., Olney, A.: Intelligent tutoring systems. APA educational psychology handbook. Vol 3: Application to learning and teaching., Washington: American Psychological Association, pp. 451–473 (2012)
4. Charsky, D.: From Edutainment to Serious Games: A Change in the Use of Game Characteristics. Games Cult., Vol. 5, No. 2, pp. 177–198, Apr. (2010)
5. Seaborn, K., Fels, D.I.: Gamification in theory and action: A survey. Int. J. Hum. Comput. Stud., Vol. 74, pp. 14–31 (2014)
6. Kölling, M.: The Greenfoot Programming Environment. Vol. 10, No. 4, pp. 1–21 (2010)
7. Werner, L., Campe, S., Denner, J.: Children Learning Computer Science Concepts via Alice Game-programming. Proc. 43rd ACM Tech. Symp. Comput. Sci. Educ., pp. 427–432 (2012)
8. Crawford, S.C.P., Dominguez, J.J.D.V.: MIT App Inventor: Enabling Personal Mobile Computing, p. 3 (2013)
9. Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J.Y., Silverman, B., Kafai, Y.: Scratch: Programming for All. Commun. ACM, Vol. 52, pp. 60–67 (2009)
10. Brennan, K.,Resnick, M.: New frameworks for studying and assessing the development of computational thinking. Annu. Am. Educ. Res. Assoc. Meet. Vancouver, BC, Canada, pp. 1–25 (2012)
11. Repenning, A., Webb, D.: Scalable Game Design and the Development of a Checklist for Getting Computational Thinking into Public Schools (2010)
12. Code Studio.: https://studio.code.org. Accesed April (2016)
13. Loyarte, H., Novara, P.: Desarrollo e implementación de un Intérprete de Pseudocódigo para la Enseñanza de Algorítmica Computacional. (2006)
14. D'Mello, S., Olney, A., Williams, C., Hays, P.: Gaze tutor: A gaze-reactive intelligent tutoring system. Int. J. Hum. Comput. Stud., Vol. 70, No. 5, pp. 377–398 (2012)
15. Landowska, A.: Affective computing and affective learning – methods , tools and prospects. Vol. 1, No. 1, pp. 16–31 (2013)
16. D'Mello, S., Graesser, A.: AutoTutor and Affective AutoTutor: Learning by Talking with Cognitively and Emotionally Intelligent Computers that Talk Back. Vol. 1, No. 212, p. 30 (2012)
17. Kazimoglu, C., Kiernan, M., Bacon, L., MacKinnon, L.: A serious game for developing computational thinking and learning introductory computer programming. Procedia - Soc. Behav. Sci., Vol. 47, pp. 1991–1999 (2012)
18. Tessler, J., Beth, B., Lin, C.: Using cargo-bot to provide contextualized learning of recursion. Proc. ninth Annu. Int. ACM Conf. Int. Comput. Educ. Res. - ICER '13, p. 161 (2013)
19. Butler-Kisber, L.: Mind, Brain, and Education: Implications for Educators. Learn. Landscapes, Vol. 5, No. 1, pp. 1–266 (2011)
20. Utting, I., Cooper, S., Kölling, M.: Alice, greenfoot, and scratch--a discussion. ACM Trans., Vol. 10, No. 4, pp. 1–11 (2010)
21. Fraser, N.: Google blockly-a visual programming editor. URL: http://code. google. com/p/blockly. Accessed Aug (2014)
22. Langner, O., Dotsch, R., Bijlstra, G.: Presentation and validation of the Radboud Faces Database. Cogn. (2010)