

# Evaluando afecto en un entorno de aprendizaje para Java

Ramón Zatarain-Cabada<sup>1</sup>, María Lucía Barrón-Estrada<sup>1</sup>,  
Francisco González-Hernández<sup>1</sup>, Giner Alor-Hernandez<sup>2</sup>

<sup>1</sup> Instituto Tecnológico de Culiacán, Culiacán, Sinaloa,  
México

<sup>2</sup> Instituto Tecnológico de Orizaba, Orizaba, Veracruz,  
México

rzatarain@itculiacan.edu.mx, galor@itorizaba.edu.mx

**Resumen.** Los cursos de programación habitualmente tienen los índices más altos de abandono. En este artículo se muestra que además de los factores cognitivos como el razonamiento y la memorización, el factor motivacional juega un papel crucial para el aprendizaje del primer lenguaje de programación. También, presentamos el desarrollo y evaluación de un entorno de aprendizaje afectivo e inteligente para Java. El entorno fue probado con estudiantes de universidad y preparatoria donde el reconocimiento afectivo fue usado solamente con los estudiantes de universidad. Se presentan los resultados y las discusiones de las pruebas.

**Palabras claves:** Inteligencia artificial, entorno inteligente de aprendizaje, entorno colaborativo, detección de afecto multimodal.

## Evaluating Affect in a Learning Environment for Java

**Abstract.** Programming courses usually have the highest dropout indices. In this paper we show that, as cognitive factors such as reasoning and memorization play a crucial role for learning of a first programming language, motivation is also essential in this process. Also, we present the developing and evaluation of an affective and intelligent learning environment for Java. The environment was tested with different university and high-school students where the affective recognition was used only with university students. We present results and discussions about the tests.

**Keywords:** Artificial intelligence, intelligent learning environment, collaborative environment; affective multimodal detection.

## 1. Introducción

Históricamente, aprender un lenguaje de programación suele ser un reto complejo para los estudiantes sin experiencia en el área de las ciencias computacionales, debido a que ellos necesitan entender nuevos conceptos de programación como estatutos, expresiones, estructuras de datos y estructuras de control. Además, los estudiantes necesitan desarrollar nuevas habilidades como la resolución de problemas, el desarrollo

de software o la depuración de código. Otros procesos cognitivos se vuelven esenciales para alcanzar el éxito en este reto como la abstracción o el pensamiento matemático. Además, también es importante considerar la obtención de habilidades como aprender las mejores prácticas de programación o el correcto uso de los paradigmas de los lenguajes de programación (orientado a objetos por ejemplo) [1-3].

Anteriormente, se presentó un Entorno Inteligente de Aprendizaje (EIA) llamado Java Sensei [4], el cual detecta el afecto del estudiante a través de sus características faciales. Además, Java Sensei utiliza estrategias pedagógicas utilizando el afecto detectado. En este trabajo, se presentan dos características nuevas añadidas al EIA. La primera característica agregada es la colaboración social entre los estudiantes que lo utilizan, para esto se analizó la literatura del área conocida como “*Computer-supported Collaborative Learning (CSCL)*” [5].

La segunda característica agregada es el reconocimiento de afecto multimodal, el cual detecta las emociones de los estudiantes a partir de diferentes dispositivos que se encuentran interactuando con ellos. De este modo, se mejora la precisión y exactitud para detectar el estado afectivo del estudiante [6].

Este trabajo está estructurada de la siguiente manera: La sección 2 muestra los trabajos relacionados en CSCL y reconocimiento de afecto multimodal; La sección 3 describe los cambios realizados en Java Sensei; La sección 4 describe los experimentos realizados con el EIA y sus resultados. La sección 5 presenta las discusiones de los resultados, conclusiones y el trabajo a futuro.

## **2. Trabajos relacionados**

### **2.1. CSCL en los EIA**

Un ejemplo de los efectos de la colaboración social es dada en [7] donde los estudiantes usan un chat después de la lectura de material académico y la realización de un examen sobre sus conocimientos. El contenido del texto generado en el chat fue analizado y en él encontraron que los estudiantes compartieron información por medio de sus opiniones y debates. El trabajo reportado en [8] analiza las actividades que los estudiantes llevan a cabo durante la colaboración y las relaciones entre esas actividades, todo mientras utilizan un software en línea. El software fue utilizado en un salón de clases donde los estudiantes podrían hablar cara a cara, además, los maestros podían entrar al software para responder las preguntas del estudiante. Los resultados probaron que la colaboración ayudo a los estudiantes a realizar tareas más complejas y de larga duración aplicando métodos de planeación y estrategia decididas en conjunto. Järvelä [9] analizo varios diseños de herramientas CSCL para encontrar características comunes entre ellas. El análisis identifico tres diseños principales: 1) Incrementar la conciencia acerca del conocimiento propio, 2) Soportar la expresión de ideas en el proceso de aprendizaje de los estudiantes, 3) Monitorear la adquisición y activación de procesos regulatorios.

### **2.2. Reconocimiento de afecto multimodal**

El Sistema Java Tutor [10] es un sistema tutor afectivo el cual usa diálogos en lenguaje natural aplicándolo en una interfaz humano a computadora o humano a

humano. El sistema aplica la teoría cognitiva ACT-R para la representación del conocimiento. Diferentes sensores en el tutor miden las señales eléctricas del cerebro, además, una cámara detecta las expresiones faciales. En ABE [15] se presenta un framework basado en agentes. Los agentes son componentes que administran el procesamiento y el flujo de la información obtenida desde diferentes dispositivos los cuales funcionan a un bajo nivel. ABE busca una solución estándar para configurar y agregar diferentes dispositivos detectores de afecto en cualquier software. FILTWAM [11] es otro framework que realiza reconocimiento de afecto en tiempo real y no intrusivo dentro de un entorno de juegos educativos. El framework está dividido en 4 capas, cada una con un rol y componentes específicos. FILTWAM presenta algunas técnicas para permitirle al usuario clasificar las fotos que le son tomadas.

Uno de las principales ventajas de Java Sensei sobre otros sistemas similares es que este sistema está enfocado en mejorar las buenas prácticas de programación. Por ejemplo, cuando el estudiante encuentra una solución a un problema, el sistema recomienda una solución mejor u óptima. Otra ventaja de Java Sensei es su sistema de recomendación que usa un modelo del estudiante para personalizar la selección de ejercicios y las ayudas y consejos dependiendo de las preferencias del estudiante. Respecto al lado tecnológico, la principal ventaja de Java Sensei es que puede ser usado en cualquier tipo de computadora (PC, laptop, móvil) que contenga un navegador web y tenga instalado una cámara y un micrófono, lo cual es común hoy en día. A diferencia de otros sistemas similares, Java Sensei no necesita sensores especiales para el reconocimiento de emociones.

### **3. Integración de características en Java Sensei**

#### **3.1. Colaboración social**

A continuación, se describirán los nuevos elementos añadidos a Java Sensei. Estos elementos no tratan de convertir al EIA en un entorno CSCL. Sin embargo, se considera que las características colaborativas ayudaran al EIA a tener más herramientas para ayudar a los estudiantes en sus procesos de aprendizajes. La literatura sobre CSCL se resumió en cuatro componentes sociales: sala de chat, debates, consejos y roles sociales.

**Sala de Chat:** Se agregó una sala de chat donde los estudiantes pueden hablar entre ellos mientras realizan un ejercicio. Cuando un estudiante inicia un ejercicio, el sistema lo asigna dentro de un equipo; posteriormente, el sistema elige un líder basándose en las habilidades globales de cada estudiante. Todos los miembros necesitan trabajar al mismo tiempo.

**Debates:** Java Sensei ya usaba un enfoque de resolución de problemas el cual es usado típicamente en los entornos CSCL también. Los entornos CSCL tratan de persuadir a los estudiantes de realizar discusiones y debates sobre el tema que están resolviendo para así llegar a una solución en común. Basándose en eso, se realizó un cambio en la parte donde se resolvían problemas en el EIA. Cuando se elige una respuesta, todos los miembros deben elegir la misma respuesta para poder continuar, en caso de que no lleguen a un acuerdo, los miembros deberán ejercer los roles sociales que les fueron asignados.

**Consejos:** El EIA trabaja como un miembro consejero dentro el equipo. Propone consejos acorde a los estados afectivos y a los estados cognitivos de los miembros del equipo. Los consejos para los estudiantes están almacenados dentro del dominio del tutor.

**Social Roles:** Cuando un ejercicio es iniciado en Java Sensei, el sistema asigna a dos miembros del equipo dos roles:

1. Líder, el líder trata de llegar a un acuerdo entre todos los miembros del equipo cuando se presenta un debate. En caso de que no haya acuerdo, él decide la respuesta final.
2. Comunicador, este miembro puede establecer contactos con otros equipos para solicitar ayudas o consejos.

A continuación, se describe el proceso que siguen los estudiantes para realizar un ejercicio dentro de Java Sensei. Cada punto del proceso es mostrado en la Fig 1.

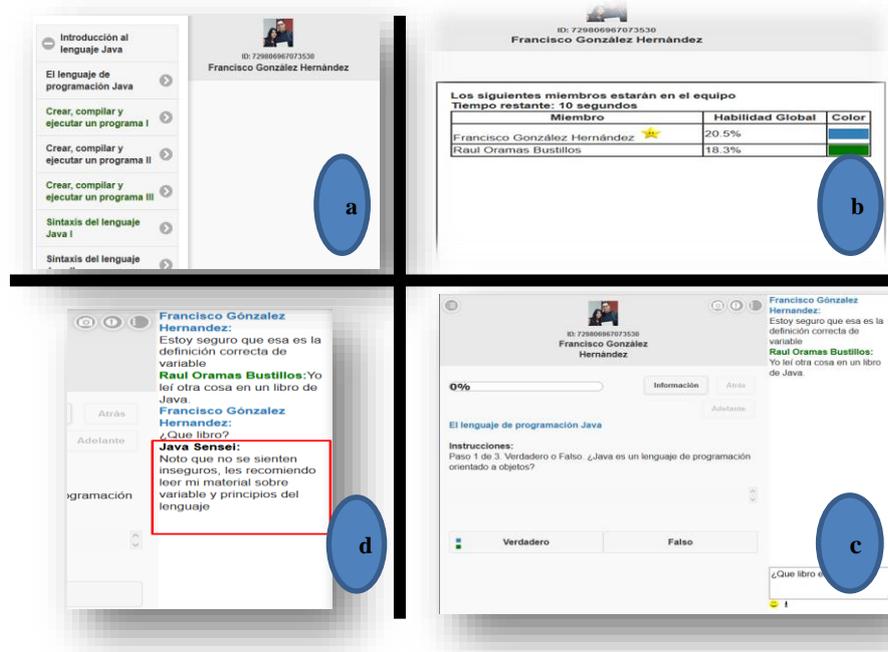


Fig. 1. Colaboración dentro de Java Sensei.

- (a) El estudiante escoge un ejercicio en el menú.
- (b) El sistema trata de emparejar al estudiante con otros estudiantes para poder crear un equipo conformado de dos a cinco miembros. El tiempo de espera para encontrar estudiantes con un conocimiento similar es de 30 segundos.
- (c) Durante la realización de los ejercicios, el tutor del sistema invita a los estudiantes a que se comuniquen entre ellos. En el momento que los estudiantes escogen una respuesta, se les da la posibilidad de compartir mensajes de voz y de texto.

- (d) En caso de que el sistema decida realizar una intervención, se mostrará como un mensaje la sala de chat.

### 3.2. Detección de afecto a través de diálogos

Java Sensei implementa un algoritmo semántico (ASEM) el cual es utiliza etiquetas semánticas para identificar la emoción del estudiante a partir de sus diálogos de texto [12], el algoritmo trabaja de la siguiente manera:

- (a) Un estudiante ingresa un texto en la sala de chat.
- (b) El texto es normalizado, esto quiere decir que a las palabras se les remueven los acentos, números y caracteres especiales. Además, las letras en mayúsculas son convertidas a minúsculas.
- (c) Las palabras que no representan una emoción como “*ella, el, eso*” son removidas por otro corpus nombrado como *StopWords*.
- (d) Las palabras semánticas son buscadas en dos corpus uno *Semántico* y el otro llamado *Palabras Impropias*. Si la palabra semántica no es encontrada en alguno de los corpus entonces es agregada a un corpus de *Nuevas Palabras*.
- (e) Si la palabra semántica es encontrada en el corpus, entonces dos características de la palabra son extraídas, el *Probability Factor of Affect* (PFA) y la emoción.
- (f) La emoción es clasificada acorde a las características obtenidas de la palabra y se utiliza un algoritmo potenciador/inhibidor (explicado más adelante).
- (g) La emoción con la intensidad más alta es producida.

El clasificador ejecuta un algoritmo potenciador/inhibidor para el valor encontrado en cada palabra, el proceso es el siguiente:

- (a) La palabra semántica es obtenida.
- (b) Esta palabra es buscada en el corpus *Semántico*. Si es encontrada, la emoción y el valor PFA son extraídos.
- (c) Si la palabra es un adverbio negativo, entonces la emoción de la siguiente palabra es invertido (Por ejemplo, de feliz a triste).
- (d) Si es la palabra es un adverbio superlativo, se identifica si la palabra es un pre-adverbio o un post-adverbio. En caso de ser un pre-adverbio se incrementa el PFA de la siguiente palabra semántica. En caso de ser un post-adverbio se incrementa el valor PFA de la palabra previa.
- (e) Finalmente, se suma el valor PFA en cada emoción encontrada en el texto de entrada y se determina que la emoción escogida es aquella que tenga el valor PFA más alto.

### 3.3. Detección de afecto a través de la voz

El algoritmo de clasificación de las emociones utiliza un *Support Vector Machine* (SVM). Para poder obtener las características principales de la voz y para entrenar el SVM, el sistema extrae la intensidad y el ritmo del audio, el cual se obtiene por medio de la sala de chat [13, 14]. Para implementar el SVM se usó la librería Java-VM [15] el cual utiliza el *Nearest Mean Classifier*. Antes de la clasificación, se tuvo que construir un corpus propio de audios. Para lograr esto, se invitó a un grupo de 20 personas (12

hombres y 8 mujeres), de 22 a 30 años de edad a ver una serie de videos, los cuales provocarían las emociones que se buscaban obtener. Después de ver los video se grabó un audio con una duración de al menos de 10 segundos en el cual expresaban sus sentimientos y emociones acerca del video visto. Al último, se les solicito evaluar la prueba self-assessment manikin (SAM) para categorizar la emoción con su correspondiente audio. El algoritmo para extraer el audio y clasificar el afecto de la voz es el siguiente:

- (a) El estudiante ingresa un mensaje de audio por medio de la sala de chat.
- (b) El Sistema extrae del audio un espectro de 1024 *frames*.
- (c) Cada *frame* es procesado obteniendo así el valor más grande y más pequeño de la intensidad y el ritmo.
- (d) Esos cuatro valores son normalizados donde cada valor tiene su propio rango de normalización
- (e) Los cuatros valores son ingresados en el SVM para la clasificación de las emociones.
- (f) El SVM predice la emoción.

### 3.4. Sistema lógico difuso para el reconocimiento multimodal

Java Sensei usa un conjunto de reglas difusas para la evaluación pedagógica de los estudiantes. Las reglas cuentan con 5 variables de entrada y 3 variables de salida. Las variables de entrada son: 1) emoción del rostro, 2) emoción del habla, 3) emoción del dialogo, 4) habilidad global (en referencia a su dominio sobre el tema) y 5) calidad de las respuestas (en referencia al ejercicio actual). Las variables de salida son: 1) retroalimentación (parte cognitiva), 2) expresión (apoyo afectivo) e 3) intervención (mensajes de parte del agente pedagógica). El EIA ejecuta el siguiente procedimiento cuando los estudiantes realizan sus ejercicios en cualquier navegador web:

- (a) El navegador web del estudiante guarda la información generada por el usuario (imágenes del rostro, mensajes de voz y mensajes de texto).
- (b) Después de un intervalo de 20 segundos, el navegador web envía los datos al servidor. Estos datos están administrados por el sistema.
- (c) El sistema valida que todos los estudiantes envíen sus datos; después de eso, el sistema evalúa cada emoción.
- (d) Si la mayoría de los estudiantes necesitan una intervención, el sistema enviara un mensaje el cual se construye a partir de las variables de salidas de expresión y retroalimentación.
- (e) El sistema mostrará la información en el navegador del estudiante.

La Fig. 2 muestra la interconexión entre las diferentes fuentes de datos afectivos y el procedimiento que realiza el EIA.

## 4. Experimentos, resultados y discusiones

El experimento de este trabajo se comparó con el experimento reportado previamente en [16] con una versión no multimodal ni colaborativa del sistema, esto con el objetivo de poder comparar si hubo mejoras después de los cambios. El primer

experimento fue llevado a cabo con un grupo de 20 estudiantes universitarios de ingeniería en sistemas del Instituto Tecnológico de Culiacán (México). El tema evaluado fue “herencia” y se consideraron diferentes puntos como el conocimiento sobre herencia y los buenos hábitos de programación. Java Sensei detectaba emociones mientras los estudiantes usaban el entorno. Se realizó una evaluación pre-test y otra post-test. Los resultados del experimento son mostrados en la Fig. 3.

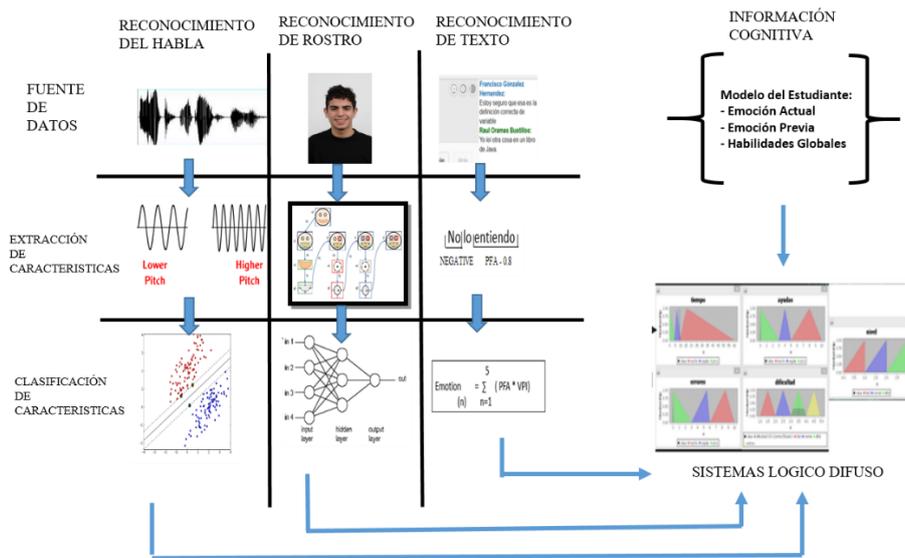


Fig. 2. Unificación entre el reconocimiento multimodal y los aspectos cognitivos.

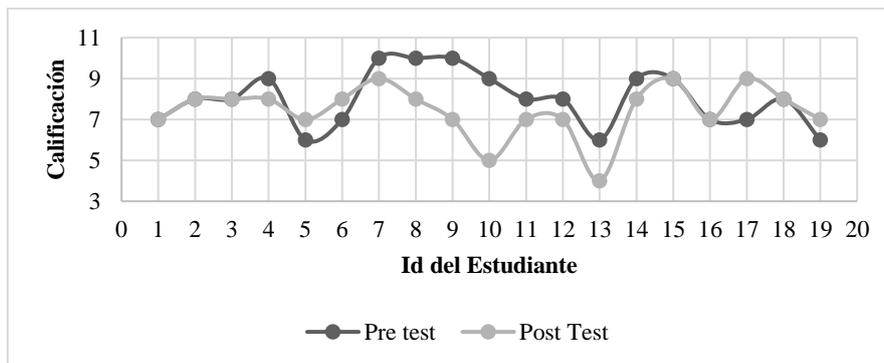


Fig. 3. Resultados con estudiantes de nivel universitario.

En el experimento anterior la muestra fue dividida en dos grupos: control y experimental. Los resultados reportan que los 19 estudiantes del grupo experimental mejoraron un 44.4% sus calificaciones en comparación al grupo control, 35% se mantuvieron sin cambios y 21% bajo sus calificaciones.

El segundo experimento fue llevado a cabo con alumnos de preparatoria en el Instituto Chapultepec (Culiacán, Sinaloa, México). La muestra consistía en treinta y dos estudiantes de dos grupos con una edad promedio de 15.125 años y una desviación estándar de 0.33. Antes de realizar los experimentos, los estudiantes se encontraban tomando un curso introductorio en el lenguaje de programación JavaScript. Se eligieron cuatro temas: Introducción a Java, variables y cálculos, estatutos de selección e estatutos de iteración. Todos esos temas son cubiertos por el dominio de Java Sensei. Se realizaron tres sesiones de 45 minutos, para manejar los mismos tiempos que se utilizaron en el primer experimento. En la primera sesión, los estudiantes resolvieron el pre-test en papel. En la siguiente sesión, los estudiantes usaron el EIA. En la última sesión, los estudiantes resolvieron el post-test en papel. Los maestros tuvieron instrucciones en como poder usar el EIA.

Ambas pruebas contuvieron 15 preguntas de tres tipos: el tipo 1 el cual evalúa conceptos teóricos con opciones “verdadero y falso”; el tipo 2 el cual también evalúa conceptos teóricos, pero con opciones múltiples; el tipo 3 el cual evaluaba la habilidad para entender una pieza de código o un programa completo. La tabla 1 muestra la distribución del tipo de pregunta en ambos exámenes.

**Tabla 1.** Distribución del tipo de preguntas en ambos exámenes.

Tipo de pregunta	Pre-test	Post-test
Uno	2	2
Dos	6	3
Tres	7	10

La puntuación de la prueba es un número entre 0 y 100. Se describen los análisis estadísticos descriptivos de los resultados de ambas pruebas. La tabla 2 muestra la media, la mediana y sus correspondientes medidas de dispersión la cuales son la desviación estándar (DE), el primer cuartil (PC) y el tercer cuartil (TC).

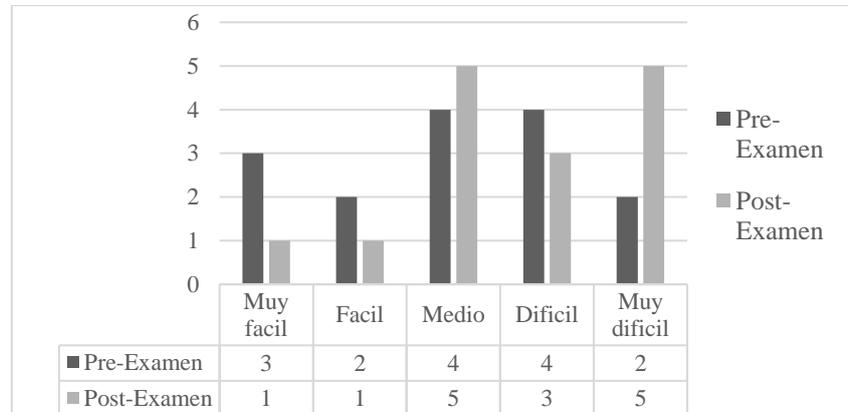
**Tabla 2.** Resultados de las pruebas.

Prueba	Media	DE	Mediana	PC	TC
Pre-test	46.7	12.36	46.7	40	53
Post-test	37.29	10.78	40	26.67	40

Además, se evaluó la confiabilidad de ambas pruebas del segundo experimento (pre y post tests). Para evaluar ambas pruebas, se usó la metodología descrita en [17]. La metodología mide la complejidad de las preguntas asignándole una categoría a cada una. La ecuación 1 es usada para obtener el Índice de Confiabilidad (IC):

$$IC = \text{respuestas correctas} / \text{total de preguntas.} \quad (1)$$

El IC es comparado y clasificado en una escala mostrada en la metodología. La escala tiene las categorías: muy fácil, fácil, medio, difícil y muy difícil. Los resultados de la medida son mostrados en la Fig. 4.



**Fig. 4.** Comparación del nivel de dificultad en ambas pruebas.

También se midió la media del puntaje obtenido de los cuatro tópicos evaluados en las pruebas. La tabla 3 muestra los resultados.

**Tabla 3.** Resultados de cada tema.

Tema	Pre-test	Post-test
<b>Introducción a Java</b>	45.57	39.25
<b>Variables y cálculos</b>	29	23
<b>Estatutos de selección</b>	48.25	45.25
<b>Estatutos de iteración</b>	55.33	43.66

Los resultados en la tabla indican que aparentemente los estudiantes en el segundo experimentos no tuvieron un cambio importante en sus resultados, en contraste a los estudiantes de ingeniería del primer experimento mostrados en la Fig. 3. Este resultado está generalizado para todos los estudiantes; la desviación estándar en la Tabla 2 demuestra que el puntaje de los estudiantes casi no excede los 50 puntos de los 100 puntos posibles. Además, los cuartiles muestran que los puntajes son homogéneos. Estos resultados llevan a pensar que las características emocionales tiene un impacto importante en los estudiantes de modo que se estudiaría el impacto de agregar más detectores emocionales. Es importante considerar todas esas características. La importancia de entender la influencia de las emociones en los estudiantes es algo que merece una considerable atención.

Una observación que se averiguo en el análisis de confiabilidad de los exámenes es que los estudiantes tuvieron problemas para entender los temas introductorios. El aprender nuevos conceptos es una tarea siempre complicada. Al contrario, los temas relacionados con razonamiento de programación o de codificación tuvieron pocos problemas. Todo esto se refleja en la Tabla 3. Además, otro importante factor (y conclusión) es que los estudiantes de preparatoria están menos comprometidos para aprender un tema como la programación a comparación de los estudiantes de ingeniería. Esto ayuda a entender que temas necesitan más atención en el diseño de las pruebas con respecto a los estudiantes con un menor nivel académico.

Por otra parte, se observa que el post-test fue diseñado con una complejidad más grande en los temas de “Introducción a Java” y “Variables y cálculos”. Por ejemplo, en la Fig. 4 se observa que ambos exámenes tienen 8 preguntas con un nivel más duro. Esto vuelve necesario la redefinición de las pruebas. Posiblemente, se tenga que modificar la distribución de las preguntas mostradas en la Tabla 1.

## 5. Conclusiones y trabajo a futuro

En este trabajo se ha aprendido la complejidad de unir los conceptos usados en CSCL con detección multimodal para enseñarles a los estudiantes nuevos conceptos de un lenguaje de programación. Integrar todo esto es un reto que ayudará a entender que aspectos son necesarios para ayudar a los estudiantes. Es complicado considerar que solamente una prueba pueda evaluar todos los tipos de grupos. En el futuro, se realizarán más pruebas para validar la confiabilidad de la integración multimodal. Otra importante meta es trabajar más para mejorar la clasificación y la integración de diferentes reconocedores en Java Sensei. Finalmente, se quiere agregar más técnicas con respecto al aprendizaje colaborativo.

## Referencias

1. McCracken, M., Almstrum, V., Diaz, D., Guzdial, M., Hagan, D., Kolikant, Y.B.-D., Laxer, C., Thomas, L., Utting, I., Willusz, T.: A multi-national, multi-institutional study of assessment of programming skills of first-year CS students. *ACM SIGCSE Bulletin*, Vol. 33, No. 4, p. 125–180 (2001)
2. Matthíasdóttir, Á.: How to teach programming languages to novice students? Lecturing or not. *International Conference on Computer Systems and Technologies-CompSysTech*. (2006)
3. Robins, A., Rountree, J., Rountree, N.: Learning and teaching programming: A review and discussion. *Computer Science Education*, Vol. 13, No. 2, pp. 137–172 (2003)
4. Zatarain Cabada, R., Barron Estrada, M.L., Gonzalez Hernández, F.: An Affective Learning Environment for Java. *Advanced Learning Technologies (ICALT), IEEE 15th International Conference on, IEEE* (2015)
5. Dillenbourg, P., Järvelä, S., Fischer, F.: The evolution of research on computer-supported collaborative learning. *Technology-enhanced learning*, Springer, pp. 3–19 (2009)
6. D'mello, S.K., Kory, J.: A Review and Meta-Analysis of Multimodal Affect Detection Systems. *ACM Computing Surveys (CSUR)*, Vol 47, No. 3, p. 43 (2015)
7. Dowell, N.M., Cade, W.L., Tausczik, Y., Pennebaker, J., Graesser, A.: What works: Creating adaptive and intelligent systems for collaborative learning support. *Intelligent Tutoring Systems*, Springer (2014)
8. Janssen, J., Erkens, J., Kirschner, P., Kanselaar, G.: Task-related and social regulation during online collaborative learning. *Metacognition and Learning*, Vol. 7, No. 1, pp. 25–43 (2012)
9. Järvelä, S., Kirschner, P., Panadero, E., Malmberg, J., Phielix, C., Jaspers, J., Koivuniemi, M., Järvenoja, H.: Enhancing socially shared regulation in collaborative learning groups: designing for CSCL regulation tools. *Educational Technology Research and Development*, Vol. 63, No. 1, pp. 125–142 (2014)
10. Wiggins, J.B., Boyer, K.E., Baikadi, A., Ezen-Can, A., Grafsgaard, J.F., Young Ha, E., Lester, J.C., Mitchel, C.M., Wiebe, E.N.: JavaTutor: An Intelligent Tutoring System that

- Adapts to Cognitive and Affective States during Computer Programming. Proceedings of the 46th ACM Technical Symposium on Computer Science Education, ACM (2015)
11. Bahreini, K., Nadolski, R., Westera, W.: Towards multimodal emotion recognition in e-learning environments. *Interactive Learning Environments*, (ahead-of-print), pp. 1–16 (2014)
  12. Wu, C.-H., Chuang, Z.-J., Lin, Y.-C.: Emotion recognition from text using semantic labels and separable mixture models. *ACM transactions on Asian language information processing (TALIP)*, Vol. 5, No. 2, pp. 165–183 (2006)
  13. Krüger, S.E.: Speech recognition with support vector machines in a hybrid system. *INTERSPEECH*, Citeseer, (2005)
  14. Pan, Y., Shen, P., Shen, L.: Speech emotion recognition using support vector machine. *International Journal of Smart Home*, Vol. 6, No. 2, pp. 101–107 (2012)
  15. Abeel, T., Van de Peer, Y., Saeys, Y.: Java-ML: A machine learning library. *The Journal of Machine Learning Research*, Vol. 10, pp. 931–934 (2009)
  16. Barrón-Estrada, M.L., Zatarain-Cabada, R., González Hernández, F., Oramas Bustillos, R., Reyes-García, C.A.: An Affective and Cognitive Tutoring System for Learning Programming. *Advances in Artificial Intelligence and Its Applications*, Springer, pp. 171–182 (2015)
  17. Escudero, E.B., Reyna, N.L., Morales, M.R.: The level of difficulty and discrimination power of the Basic Knowledge and Skills Examination (EXHCOBA). *Revista Electrónica de Investigación Educativa*, Vol. 2, No. 1, p. 2 (2002)