

LEVEL OF UTILIZATION OF THE VIRTUAL CARAMBA PLATFORM VS. ACADEMIC PERFORMANCE IN PROGRAMMING FUNDAMENTALS: A QUANTITATIVE ANALYSIS

NIVEL DE UTILIZACIÓN DE LA PLATAFORMA VIRTUAL CARAMBA VS. RENDIMIENTO ACADÉMICO EN FUNDAMENTOS DE PROGRAMACIÓN: UN ANÁLISIS CUANTITATIVO

Pavel Novoa-Hernández¹ Milvio A. Novoa Pérez²

¹Ph.D., Universidad Técnica Estatal de Quevedo, Quevedo, Los Ríos, Ecuador. pnovoa@uteq.edu.ec

²Ph.D. Director General Adjunto para Posgrado e Investigación, Instituto Superior Politécnico Metropolitano De Angola, Angola.

Abstract

Learning computer programming is a skill hard to achieve in college students. One of the current approaches to dealing with this difficulty is to introduce Visual Programming Languages such as Scratch in the initial courses. The goal is to soften the learning curve and focus the student on acquiring algorithmic and computational thinking. However, there is evidence that the motivation of more prepared students may be affected since they usually expect more professional environments to work with. One recent alternative to solve this dilemma is CARAMBA, a Web platform that involves an exercise recommendation system along with Scratch. In this way, students can build their learning paths according to their interests and level of knowledge. Although the success of CARAMBA is remarkable, some questions related to its impact still remain to be answered. In this sense, the present investigation aims to analyze the correlation existing between two important variables: the level of utilization of CARAMBA and the improvement rate on a computer programming course. The results obtained from an Ecuadorian university show a positive, significant correlation between these two variables, which is an additional evidence of the benefits of CARAMBA.

Keywords: *computer programming learning, Scratch, CARAMBA, correlation analysis.*

Resumen

Aprender a programar en una computadora es una habilidad difícil de lograr en los estudiantes universitarios. Uno de los enfoques actuales para lidiar con esta dificultad es introducir lenguajes de programación visuales como Scratch, en los cursos iniciales de computación. La idea es suavizar la curva de aprendizaje y enfocar al estudiante en adquirir un pensamiento algorítmico y computacional. No obstante, en el caso de Scratch, existen evidencias de que puede afectar la motivación de los estudiantes más preparados, pues éstos esperan programar en entornos más profesionales. Una herramienta que busca resolver este dilema es CARAMBA, una plataforma Web que integra un sistema de recomendación de ejercicios con el propio Scratch. De esta forma, los estudiantes pueden construir sus caminos de aprendizaje de acuerdo a sus intereses y nivel de conocimiento. Aunque el éxito de CARAMBA es notable, aún quedan por responder algunas interrogantes relacionadas con su impacto. En este sentido, la presente investigación tiene por objetivo analizar la correlación del nivel de utilización de CARAMBA por parte de los estudiantes y su

efecto en el aprendizaje de programación. Los resultados, obtenidos en una universidad ecuatoriana, muestran que existe una correlación positiva y significativa entre estas variables, lo cual es una evidencia adicional de los beneficios de CARAMBA.

Palabras clave: *aprendizaje de programación, Scratch, CARAMBA, análisis de correlación.*

I. INTRODUCCIÓN

El aprendizaje de la programación es una de las tareas más complejas para el estudiante universitario de carreras tecnológicas como la Ingeniería Informática, Ciencias de la Computación, entre otras. Varias son las causas que derivan en este problema. Una muy común es la introducción de lenguajes de programación profesionales desde los primeros cursos. Esto provoca que el estudiante se enfoque significativamente en detalles tales como la sintaxis del programa, la cual (aunque importante) no es relevante para desarrollar un pensamiento algorítmico y computacional. Es por eso que algunas estrategias educativas emplean los denominados lenguajes de programación visuales. Ejemplo de este tipo de lenguaje son Scratch (Maloney, Resnick, & Rusk, 2010) y Alice (Costa & Miranda, 2017).

En el caso específico de Scratch, la literatura refleja más experiencias positivas que negativas en el ámbito de la educación superior. Por ejemplo (Bittencourt et al., 2015; Cardenas, Benavides, D'Armas, Vinueza, & Rodas, 2017; Erol & Kurt, 2017; Jancheski, 2017; Mishra, Balan, Iyer, & Murthy, 2014; Quan, 2017; Wolz et al., 2009) reportan beneficios en cursos introductorios de computación a nivel de universidad. Sin embargo, a diferencia de estas experiencias positivas, en el caso de (Martínéz-Valdés, Ángel Vélazquez-Iturbidé, & Hijon-Néira, 2017; Tanrikulu & Schaefer, 2011) los autores resaltaron algunas dificultades durante su aplicación.

Tal vez el principal problema con Scratch en la universidad sea el de la falta de motivación por parte de los estudiantes que tienen aspiraciones más altas sobre la asignatura de programación. Por ejemplo, en (Tanrikulu & Schaefer, 2011) se ha podido constatar que los estudiantes que poseen una mayor preparación en programación se sienten desmotivados al usar Scratch (incluso en etapas tempranas del curso), pues tienen expectativas de emplear un entorno de desarrollo más avanzado y profesional. Para resolver este dilema es preciso buscar una solución que se ajuste a los intereses personales de cada estudiante. Por tales motivos, en una investigación anterior (Cárdenas-Cobo, Novoa-Hernández, Puris, & Benavides, 2018) se propuso extender a Scratch mediante un sistema de recomendación de ejercicios. La herramienta resultante, denominada CARAMBA, posibilita que cada estudiante construya su propio camino de aprendizaje, esto es, a través de una práctica personalizada con Scratch. Específicamente, los estudiantes pueden acceder a los ejercicios de dos formas, o bien seleccionándolos del conjunto de todos los ejercicios disponibles, o a partir de los recomendados por la plataforma. Para recomendar convenientemente los ejercicios CARAMBA se basa en dos criterios con los que el estudiante evalúa a cada ejercicio que enfrenta: gusto y nivel de complejidad. Aplicando entonces un enfoque de filtrado colaborativo (Manouselis, Drachsler, Verbert, & Duval, 2013), CARAMBA es capaz de recomendar ejercicios que coinciden con las aspiraciones de cada usuario.

Hasta donde se tiene conocimiento, CARAMBA ha sido evaluada desde el punto de vista de expectativas de los estudiantes. Véase por ejemplo (Cárdenas-Cobo et al., 2018; Cárdenas-Cobos, Novoa-Hernández, Trujillo-Reyes, & Puris-Cáceres, 2018; Cardenas et al., 2017). Sin embargo, no se tiene una medida del efecto que causa CARAMBA en el aprendizaje de los estudiantes. Para intentar responder esta cuestión, en la presente investigación se realiza un análisis correlacional de dos

Level of Utilization of the Virtual Caramba Platform vs. Academic Performance in Programming Fundamentals: A Quantitative Analysis

variables esenciales vinculadas a esta estrategia educativa: nivel de utilización del sistema y el rendimiento académico del estudiante. El estudio fue realizado en la Universidad Estatal de Milagro, del Ecuador, donde surgió CARAMBA y donde ha existido históricamente un índice de reprobación alto (superior al 50%) en la materia Fundamentos de Programación, que se imparte en las carreras de Ingeniería en Sistemas e Ingeniería Industrial. Los resultados muestran que los estudiantes que más practican con CARAMBA poseen un buen rendimiento académico en programación.

II. METODOLOGÍA

Como se mencionó anteriormente, la investigación se desarrolló con estudiantes del primer curso de la carrera Ingeniería en Sistemas de la Universidad Estatal de Milagro. Estos estudiantes reciben la materia Fundamentos de Programación en el primer semestre de la carrera. Particularmente se analizó las interacciones con CARAMBA de un total de 56 estudiantes.

Se empleó el método estadístico para analizar cuantitativamente la relación existente entre las variables nivel de utilización de CARAMBA y el rendimiento académico. Esta última variable fue analizada en términos de la nota (en base a 10 puntos, donde una nota superior o igual a 7 puntos significa que el estudiante aprueba).

Es importante mencionar que los ejercicios presentes en CARAMBA (un total de 109) fueron creados para lograr un nivel de práctica en 4 temas de la programación. Estos son: *Variables e inicialización*, *Ciclos*, *Condicionales*, y *Concurrencia*. En la Figura 1 se muestra la distribución de dichos ejercicios por cada concepto de aprendizaje. Como puede verse, existe una diversidad apreciable entre los tipos de ejercicios, principalmente para los tres primeros conceptos.

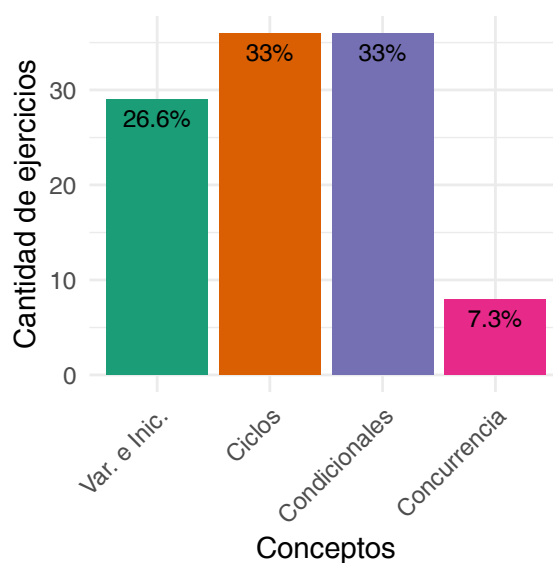


Figura 1. Distribución de los ejercicios incluidos en CARAMBA de acuerdo al tipo de concepto de aprendizaje con el que se relacionan.

Los datos relacionados con las interacciones de los estudiantes en CARAMBA, fueron obtenidos de la propia herramienta. Éstos permitieron realizar el estudio cuantitativo que se describirá en la próxima sección.

III. RESULTADOS

Concretamente, estamos interesados en responder las siguientes interrogantes:

1. ¿Los estudiantes usaron realmente a CARAMBA?
2. ¿Practicaron realmente los estudiantes con los ejercicios recomendados por CARAMBA?
3. ¿En qué medida la solución de ejercicios de un determinado concepto de aprendizaje influye en el éxito/fracaso de un estudiante en la asignatura?
4. ¿En qué medida el uso general de CARAMBA contribuyó al éxito/fracaso del estudiante en la asignatura?

En relación a la pregunta 1, la Figura 2 muestra la cantidad promedio de ejercicios realizados por los estudiantes a lo largo de las 16 semanas que duró el curso. Como se observa, de manera general, la tasa de ejercicios por estudiante-semana se mantuvo superior a 2 durante todo el curso. Es de notar que la tendencia (línea continua) de dicha tasa fue prácticamente constante y tuvo un margen de error aproximadamente de un ejercicio (zona gris alrededor de la línea continua).

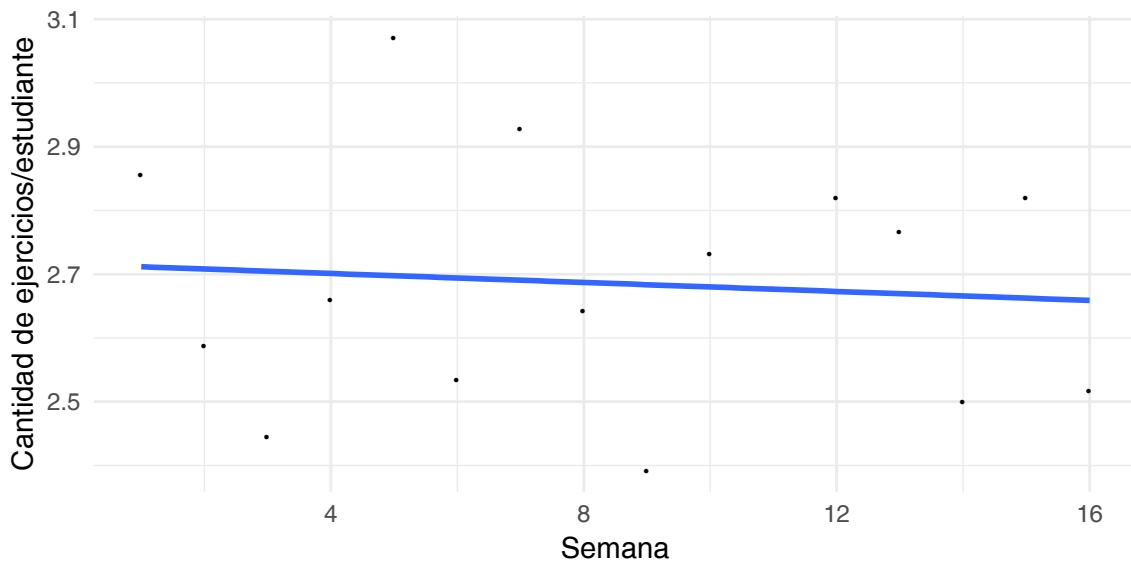


Figura 2. Nivel de ejercicios enfrentados por los estudiantes a lo largo de las 16 semanas del curso. La línea azul representa la tendencia empleando un modelo lineal a partir de la cantidad de ejercicios por estudiantes en función de las semanas. La zona gris representa el error estándar del modelo.

Comparado con la tasa ideal, esto es 6.81 (calculada a partir del total de ejercicios disponibles (109) por estudiante en las 16 semanas (asumiendo que de manera ideal cada estudiante haría una misma cantidad de ejercicios por semana.), los valores obtenidos por los estudiantes representan el 38% de dicha tasa aproximadamente. Lo anterior responde afirmativamente a la pregunta 1, a pesar del bajo valor en general de interacción con el sistema.

Para responder la pregunta 2, la Figura 3 muestra la distribución de las interacciones de los estudiantes con los ejercicios incluidos en CARAMBA, teniendo en cuenta el origen del ejercicio (Conjunto de ejercicios, Sistema de recomendación). Es importante recordar que un ejercicio puede ser seleccionado por el estudiante de dos formas en CARAMBA: a partir del conjunto de todos los

Level of Utilization of the Virtual Caramba Platform vs. Academic Performance in Programming Fundamentals: A Quantitative Analysis

ejercicios disponibles (esto es, sin recomendación) o a partir del conjunto creado por el sistema de recomendación. Como se aprecia en la Figura 3, la mayoría de las interacciones (cerca del 70%) de los estudiantes con el sistema se debieron al interés por los ejercicios recomendados por CARAMBA. Por el contrario, solo un 30% corresponde a interacciones que surgieron a partir de ejercicios no recomendados. Esto significa que los estudiantes sí practicaron con CARAMBA a través del sistema de recomendación.

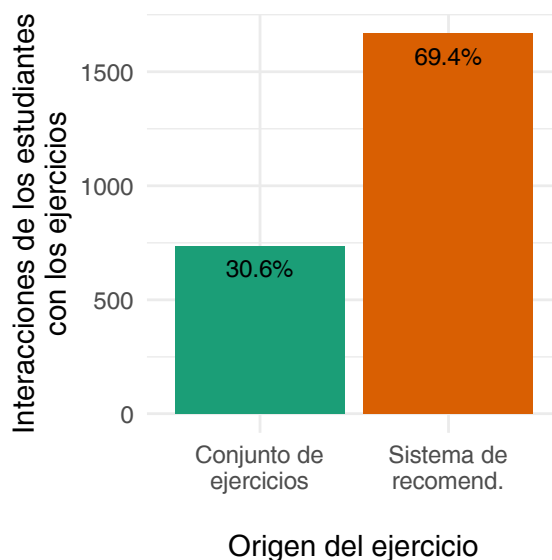


Figura 3. Distribución de las interacciones de los estudiantes de acuerdo al origen de los ejercicios.

Las preguntas 3 y 4 están estrechamente relacionadas. Ambas dependen de lo que se defina como nivel de utilización del sistema. En este caso, hemos considerado que la utilización de CARAMBA por un estudiante puede definirse como el porcentaje de ejercicios realizados a partir del total disponible (sin importar el origen: recomendado o no). Específicamente, para la pregunta 3 este porcentaje estaría calculado tomando como base la cantidad de ejercicios disponibles por cada concepto de aprendizaje. De esta forma, cada estudiante tendrá asociado 5 niveles de utilización: uno por cada concepto de aprendizaje y uno global (teniendo en cuenta todos los ejercicios disponibles). Un valor alto del nivel de utilización indicará que el estudiante interactuó con gran cantidad de ejercicios, mientras que un valor bajo, lo contrario.

En la Figura 4 se resume por cada nivel cognitivo, el nivel de utilización del sistema (eje de las x) y la nota final de los estudiantes (eje y). Recuerde que la nota está en un rango de 0 a 10, donde 0 es la peor nota posible y 10 lo contrario. Se considera que un estudiante aprobó la asignatura cuando la nota es mayor o igual a 7. Para facilitar la comparación visual en este gráfico, hemos identificado a cada estudiante con una categoría de acuerdo a su nota: estudiantes que no aprueban (Reprobado con nota inferior a 7), estudiantes promedio (Promedio con notas entre 7 y 8.5), y finalmente, estudiantes sobresalientes (Excelente con nota de 8.5 en adelante).

Las gráficas de la Figura 4 permiten sacar algunas conclusiones importantes. Inicialmente, se puede ver que la mayoría de los estudiantes tuvieron niveles de utilización superiores al 20% en todos los niveles cognitivos. Resulta interesante ver que en el caso del concepto *Concurrencia*, hubo estudiantes que mostraron un alto nivel de práctica. Sin embargo, quizás la conclusión más

importante aquí sea que, al menos para tres de los niveles cognitivos (*Variables e Inicialización, Ciclos, y Condicionales*) existe una correlación aparente entre el nivel de utilización del sistema y el éxito/fracaso del estudiante. Por ejemplo, se puede ver que los estudiantes *Excelentes* están asociados a niveles de utilización altos. Algo similar sucede con los *Promedio*. En el caso de los *Reprobados*, la gran dispersión entre los niveles de utilización de los estudiantes no permite identificar claramente esta relación.

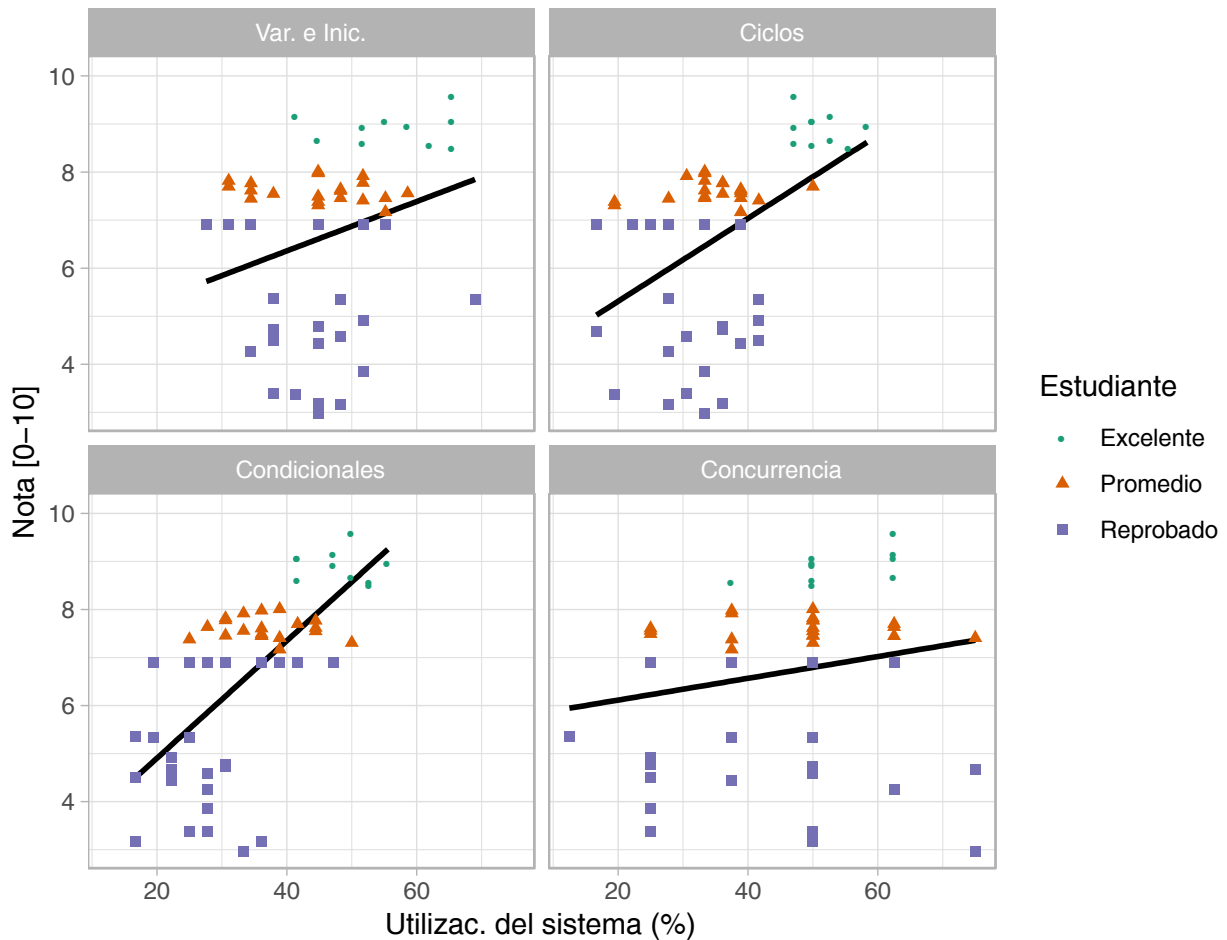


Figura 4. Relación entre el nivel de utilización del sistema y el rendimiento del estudiante de acuerdo al concepto de aprendizaje. La línea continua está derivada a partir de un modelo lineal y la zona gris es el error estándar.

En el caso de la pregunta 4, hemos procedido de manera similar. La gráfica de la Figura 5 muestra el mismo análisis, pero considerando el nivel de utilización global de cada estudiante. Aquí, el patrón de correlación (positiva) se aprecia mucho mejor.

Level of Utilization of the Virtual Caramba Platform vs. Academic Performance in Programming Fundamentals: A Quantitative Analysis

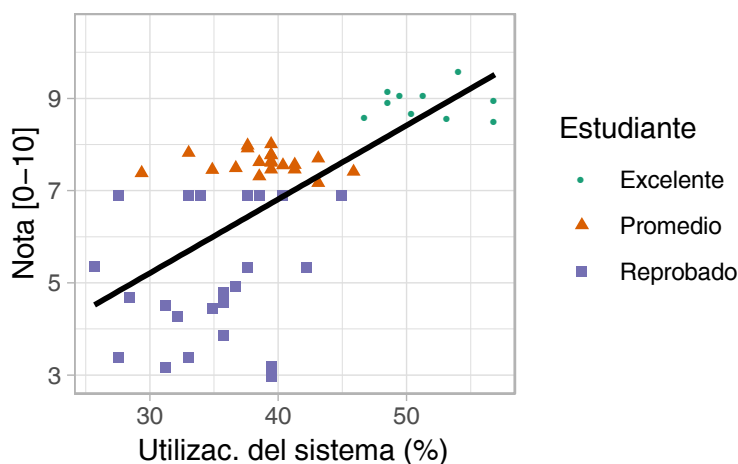


Figura 5. Relación entre el nivel de utilización del sistema vs. el rendimiento de los estudiantes teniendo en cuenta la nota final. La línea continua corresponde a un modelo lineal con un error estándar representado por la zona gris.

Para confirmar nuestra percepción de que el nivel de utilización de CARAMBA influye positiva/negativamente en el éxito/fracaso del estudiante en la asignatura, hemos procedido con una prueba de correlación. Concretamente, se aplicó la prueba de correlación de Pearson tomando como variables el nivel de utilización del sistema por cada estudiante y la nota correspondiente. Los resultados se muestran en la Tabla 1 considerando cada concepto de aprendizaje y todos los ejercicios (General). Nótese que se han incluido el p-valor de la prueba, el coeficiente de correlación y los valores mínimo y máximo del intervalo de confianza (para un 95%).

Tabla 1. Análisis de correlación por cada concepto de aprendizaje y considerando todos conceptos (General).

Concepto	p-valor	Coeficiente de Correlación	Mín.	Máx.
Var. e Inic.	3.79E-02	2.78E-01	1.65E-02	5.04E-01
Ciclos	2.46E-04	4.71E-01	2.38E-01	6.53E-01
Condicionales	4.99E-09	6.87E-01	5.18E-01	8.05E-01
Concurrencia	1.80E-01	1.82E-01	-8.53E-02	4.24E-01
General	2.56E-07	3.36E-01	2.15E-01	4.48E-01

A partir de la Tabla 1 se puede concluir que existe una correlación positiva entre el nivel de utilización y la nota del estudiante. Sin embargo, en el caso del concepto *Concurrencia*, esta correlación no es significativa ($p\text{-valor} > 0.05$). Se puede responder afirmativamente por tanto las preguntas 3 y 4.

IV. CONCLUSIONES

Los resultados obtenidos hasta aquí solo permiten concluir que, un alto nivel de práctica con CARAMBA, está asociado con un buen desempeño académico. Sin embargo, una interrogante aún sin responder es qué causa que los estudiantes exhiban ese alto nivel de práctica con CARAMBA. En nuestra opinión, esto se debe principalmente a que los estudiantes se encuentran muy motivados con CARAMBA. Hemos observado que los estudiantes de hoy, al igual que la mayoría de la sociedad moderna, se encuentran muy ligados a las tecnologías de la información y la comunicación. De manera que suponemos que al usar una herramienta inteligente como CARAMBA, que les permite aprender de forma autónoma con libertad, los estudiantes se sienten más motivados. En consecuencia, esta motivación conduce a un nivel mayor de práctica, y este nivel de práctica a una mayor probabilidad de éxito en la asignatura. Reconocemos que tal suposición implica un estudio formal en profundidad que explique estas y otras cuestiones. Nuestros trabajos futuros estarán relacionados en esa dirección.

V. AGRADECIMIENTOS

P. Novoa-Hernández agradece el apoyo de la dirección de investigación de la Universidad Técnica Estatal de Quevedo que, a través del Proyecto “Aplicaciones de la Soft Computing en escenarios de decisión de la educación superior” (FOCICYT 5ta Convocatoria) financia a la presente investigación.

VI. REFERENCIAS BIBLIOGRÁFICAS

- Bittencourt, R. A., dos Santos, D. M. B., Rodrigues, C. A., Batista, W. P., Chalegre, H. S., Santos, D. M. B. dos, ... Chalegre, H. S. (2015). Learning programming with peer support, games, challenges and scratch. In *Proceedings of the 45th Annual Frontiers In Education Conference* (pp. 1–9). IEEE. <https://doi.org/10.1109/FIE.2015.7344222>
- Cárdenas-Cobo, J., Novoa-Hernández, P., Puris, A., & Benavides, D. (2018). *Recommending exercises in scratch: An integrated approach for enhancing the learning of computer programming. Advances in Intelligent Systems and Computing* (Vol. 627). https://doi.org/10.1007/978-3-319-60937-9_20
- Cárdenas-Cobos, J., Novoa-Hernández, P., Trujillo-Reyes, Y., & Puris-Cáceres, A. (2018). RECOMENDACIÓN PERSONALIZADA DE EJERCICIOS EN SCRATCH: UNA ESTRATEGIA PARA EL APRENDIZAJE DE LA PROGRAMACIÓN EN LA UNIVERSIDAD. *Universidad Ciencia y Tecnología*, 1(1).
- Cardenas, J., Benavides, D., D'Armas, M., Vinuesa, M., & Rodas, J. (2017). Programación con la herramienta SCRATCH +CARAMBA. Una experiencia de aprendizaje significativo. In *Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology* (Vol. 2017–July). <https://doi.org/10.18687/LACCEI2017.1.1.268>
- Costa, J. M., & Miranda, G. L. (2017). Relation between Alice software and programming learning: A systematic review of the literature and meta-analysis. *British Journal of Educational Technology*, 48(6), 1464–1474. <https://doi.org/10.1111/bjet.12496>
- Erol, O., & Kurt, A. A. (2017). The effects of teaching programming with scratch on pre-service information technology teachers' motivation and achievement. *Computers in Human Behavior*, 77, 11–18. <https://doi.org/10.1016/j.chb.2017.08.017>

Level of Utilization of the Virtual Caramba Platform vs. Academic Performance in Programming Fundamentals: A Quantitative Analysis

- Jancheski, M. (2017). Improving teaching and learning computer programming in schools through educational software. In *Olympiads in Informatics* (Vol. 11, pp. 55–75). <https://doi.org/10.15388/ioi.2017.05>
- Maloney, J., Resnick, M., & Rusk, N. (2010). The Scratch programming language and environment. *ACM Transactions on Computing Education*, 10(4), 1–15. <https://doi.org/10.1145/1868358.1868363>.http
- Manouselis, N., Drachsler, H., Verbert, K., & Duval, E. (2013). Survey and Analysis of TEL Recommender Systems. In *Recommender Systems for Learning* (pp. 37–61). New York, NY: Springer New York. https://doi.org/10.1007/978-1-4614-4361-2_3
- Martínez-Valdés, J. A., Angél Vélazquez-Iturbidé, J., & Hijon-Néira, R. (2017). A (relatively) unsatisfactory experience of use of Scratch in CS1. In *ACM International Conference Proceeding Series* (Vol. Part F1322). <https://doi.org/10.1145/3144826.3145356>
- Mishra, S., Balan, S., Iyer, S., & Murthy, S. (2014). Effect of a 2-week Scratch Intervention in CS1 on Learners with Varying Prior Knowledge. In *Proceedings of the 2014 Conference on Innovation & Technology in Computer Science Education* (pp. 45–50). New York, NY, USA: ACM. <https://doi.org/10.1145/2591708.2591733>
- Quan, C. G. (2017). The pros and cons of requiring SCRATCH in the applied linguistics classroom: Student and student-teachers' reactions and recommendations, instructor's observations for EISTA 2017. In *IMSCI 2017 - 11th International Multi-Conference on Society, Cybernetics and Informatics, Proceedings* (pp. 87–92). Retrieved from <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85034239731&partnerID=40&md5=1709aeb7f034106f955c3599c0847151>
- Tanrikulu, E., & Schaefer, B. C. (2011). The users who touched the ceiling of scratch. In H. İ. Yalin, F. Adiloglu, H. Boz, S. Karataş, & F. Ozdamli (Eds.), *World Conference on Educational Technology Researches - 2011* (Vol. 28, pp. 764–769).
- Wolz, U., Leitner, H. H., Malan, D. J., Maloney, J., Wolz, U., Leitner, H. H., ... Maloney, J. (2009). Starting with scratch in CS 1. In *Proceedings of the 40th ACM technical symposium on Computer science education - SIGCSE '09* (Vol. 41, p. 2). New York, New York, USA: ACM Press. <https://doi.org/10.1145/1508865.1508869>