

META-ANÁLISIS

SOBRE EL IMPACTO DE SCRATCH EN EL APRENDIZAJE DE LA PROGRAMACIÓN POR COMPUTADORA

META-ANALYSIS OF SCRATCH'S IMPACT ON COMPUTER PROGRAMMING LEARNING

Pavel Novoa Hernández¹

E-mail: pnovoa@uteq.edu.ec

ORCID: <https://orcid.org/0000-0003-3267-6753>

Ricardo Hablich Proaño¹

E-mail: juan.hablich2013@uteq.edu.ec

ORCID: <https://orcid.org/0000-0003-0529-8195>

Milvio Novoa Pérez²

E-mail: dapi@unimetroangola.co.ao

ORCID: <https://orcid.org/0000-0002-4542-3619>

¹ Universidad Técnica Estatal de Quevedo. Ecuador.

² Instituto Superior Politécnico Metropolitano de Angola. Angola.

Cita sugerida (APA, séptima edición)

Novoa Hernández, P., Hablich Proaño, R., & Novoa Pérez, M. (2020). Meta-análisis sobre el impacto de Scratch en el aprendizaje de la programación por computadora. *Revista Universidad y Sociedad*, 12(S1), 302-307.

RESUMEN

El presente trabajo se mide el impacto del lenguaje de programación visual Scratch sobre el aprendizaje de la programación por computadora. Se desarrolló un meta-análisis basado en los trabajos más relevantes publicados, los cuales fueron extraídos a partir de las bases de datos: Scopus y Web of Science. Un total de 10 resultados obtenidos a partir de 8 trabajos publicados en el periodo 2008 a 2019, fueron procesados estadísticamente. Como resultado, se pudo confirmar que efectivamente Scratch posee un impacto positivo en el aprendizaje de la programación. El tamaño del efecto global fue significativo e igual a 0.56 con un intervalo de confianza [0,19; 0,92] al 95%. Este tamaño del efecto no es sensible a las variables moderadoras definidas, excepto por la duración de la intervención. El test de asimetría de Egger mostró que no existe de sesgo de publicación.

Palabras clave: Scratch, Meta-analysis, Lenguaje de Programación Visual, Aprendizaje de programación por computadora.

ABSTRACT

In this work the impact of the visual programming language Scratch on the learning of computer programming is measured. A meta-analysis was developed based on the most relevant works published, which were extracted from Scopus and Web of Science. A total of 10 results obtained from 8 papers published in the period 2008 to 2019 were processed statistically according to a meta-analysis approach. As a result, it was confirmed that Scratch has a positive impact on learning programming. The effect size was significant and equal to 0.56 with a 95% confidence interval of [0.19; 0.92]. It was found that this effect size is not sensitive to the moderating variables defined, except for the duration of the intervention. Egger's asymmetry test showed that no publication bias was observed.

Keywords: Scratch, Meta-analysis, Visual Programming Language, Computer programming learning.

INTRODUCCIÓN

En los enfoques de educación tradicional, los profesores desempeñan un papel importante porque guían el proceso de aprendizaje según la estructura planificada en la materia. En otras palabras, el conocimiento se revela al estudiante de manera organizada. Con frecuencia, esto se denomina aprendizaje formal. Independientemente de los beneficios de este modelo educativo, en los últimos 20 años el llamado Aprendizaje mejorado de tecnología ha planteado modelos alternativos. TEL ha sido definido como un dominio de aplicación que cubre tecnologías para apoyar todas las formas de actividades de enseñanza y aprendizaje. Un ejemplo importante en el contexto de la educación de programación es Scratch un lenguaje de programación visual, que busca reducir la curva de aprendizaje en niños y principiantes en la programación por computadora.

Curiosamente, Scratch ha sido empleado en cursos introductorios de Ciencias de la Computación en carreras universitarias. En este contexto, Scratch logró que los estudiantes inexpertos se familiarizaran exitosamente con los fundamentos de la programación. Malan & Leitner (2007), explican este resultado debido a que Scratch ayuda a los estudiantes a ver a la programación desde una perspectiva diferente de los entornos tradicionales. La transición de los estudiantes a los lenguajes de programación Java o C se vuelve por tanto más fácil después de aprender Scratch.

El uso de lenguajes de programación visuales como Scratch tiene la ventaja de eliminar los problemas de sintaxis, por lo que los estudiantes se pueden enfocar en la semántica del lenguaje. De esta forma, puede proporcionar experiencias tanto concretas como abstractas y, en consecuencia, puede utilizarse como un medio para la transición hacia la programación “real”, esto es, lenguajes de alto nivel profesionales.

Independientemente del éxito de esta herramienta, a día de hoy no resulta claro cuál es su verdadero impacto en el aprendizaje de la programación. Resulta claro que conocer la magnitud de este impacto, apoyaría el proceso de decisión de docentes y/o instituciones que se planteen adoptar Scratch como asistente para la enseñanza de la programación.

- Investigaciones similares existen, pero ninguna cuantifica dicho impacto basado en las evidencias reportadas en la literatura. Por ejemplo, Costa & Miranda (2017), aunque revisan y meta-analizan evidencias, se enfocaron en Alice, un lenguaje visual diferente a Scratch. Por otro lado, Moreno-Leon & Robles (2016), realizan una revisión sistemática de la literatura sobre investigaciones que utilizan la programación con Scratch en temas no relacionados con la informática y las comunicaciones, así como estudios que analizan el tipo de habilidades que desarrollan los estudiantes mientras aprenden a codificar en este entorno. Dentro de este estudio se observan las diferentes materias que se pueden aprender mediante el uso de Scratch. Hay artículos que estudian el impacto de la programación en materias STEM como Matemáticas (Armería Zavala, Herández Gallardo & García-Ruiz, 2013; Ke, 2014). Dentro de los diferentes estudios analizados en este artículo se encontró que la programación puede ser una herramienta para mejorar el aprendizaje. asignaturas como matemáticas, ciencias, artes, escritura o inglés como segundo idioma. Sin embargo, estos estudios no siguieron, en una mayoría, las recomendaciones básicas para desarrollar la investigación por lo que una mayor cantidad de estudios con datos cuantitativos son necesarios para dar conclusiones más claras. En este estudio también se demuestra que al aprender a programar los estudiantes mejoraron sus habilidades de resolución de problemas, razonamiento, pensamiento lógico y creatividad.

DESARROLLO

Los estudios fueron seleccionados para el meta-análisis de acuerdo a los criterios de inclusión y exclusión que se muestran en la Tabla 1.

Tabla 1. Criterios de inclusión y exclusión.

Criterios de inclusión	Criterios de exclusión
Publicaciones a partir de 2001 hasta 2019.	Estudios duplicados
Accesible en la fuente que lo alberga.	No accesibles
La enseñanza con Scratch fue comparada contra otro método de enseñanza	No es evaluado el aprendizaje con Scratch.
	Estudios sin datos numéricos

Para asegurar la rigurosidad en la búsqueda de la bibliografía asociada, se han seguido los estándares de calidad de la declaración PRISMA (Moher, et al., 2009), en especial, a los criterios de selección de los estudios. Como consecuencia, se procedió a recopilar los trabajos

relacionados con el tema, a través de la consulta a dos bases de datos bibliográficas de alto impacto científico: Web of Science y Scopus.

Los términos incluidos en la consulta fueron seleccionados a partir de las siguientes palabras claves:

Scopus:

TITLE (scratch AND (learn* OR comput* OR programm* OR "systematic review" OR meta-anal*))

Web of Science:

TITLE (scratch AND (learn* OR comput* OR programm* OR "systematic review" OR meta-anal*))

Las búsquedas realizadas con las consultas descritas anteriormente arrojaron un total de 233 trabajos, con los que se procedió a simplificar los estudios duplicados (trabajos en ambas bases de datos). Dos autores procedieron con una revisión preliminar teniendo en cuenta solo los títulos y resúmenes. Como resultado, quedaron un total de 83 trabajos. En esta revisión independiente, la concordancia observada correspondiente y la concordancia debido al azar, fueron y (p-value), respectivamente.

Finalmente, aplicando los criterios de inclusión descritos en el apartado anterior, y teniendo en cuenta las preguntas de investigación, resultaron 10 trabajos. Para asegurarnos de que el proceso de exclusión era el adecuado y existía similitud en los resultados de exclusión, se realizó un análisis doble. En este caso la concordancia observada fue de y la concordancia atribuible al azar de (p-value).

Para cada estudio se definieron varias características que fueron codificadas como variables moderadoras. En la Tabla 2, se enumeran las variables definidas incluyendo su significado y dominio. Se debe tener en cuenta que estas variables corresponden a las preguntas de investigación que guían la investigación.

Tabla 2. Descripción de las variables.

Nombre de la variable	Significado y dominio
Comparado.Con	Con quien es comparado Scratch Nivel: {Método tradicional, Otros}
Participantes.Universitarios	Se refiere al nivel de estudio de los participantes presentes en el estudio. Nivel {No, SI}
Duracion.Semestral	Tiempo en el cual fue realizado el estudio Nivel: {No, Si}

Región	Regiones en la que se realizaron los estudios Nivel {Eurasia, América}
--------	---

El tamaño del efecto a calcular se ha definido como el impacto de Scratch en el aprendizaje de programación. Más formalmente, como una media estandarizada. La tecnología empleada para el procesamiento estadístico fue el lenguaje de programación R.

Una vez revisados en profundidad los 10 resultados (Tabla 3) y teniendo en cuenta las variables definidas anteriormente, se observa que el interés por evaluar a Scratch desde un punto de vista de aprendizaje ha crecido a lo largo del tiempo. Por supuesto, la Tabla 3 solo muestra una cantidad muy reducida de los trabajos disponibles sobre este tema, dado que se han dejado fuera investigaciones que no cumplieron con nuestros criterios de inclusión y exclusión.

En la Tabla 3 se resumen las principales características de los 10 trabajos que fueron seleccionados a través del protocolo de búsqueda adoptado. Como se aprecia, los trabajos encontrados han sido publicados en los años comprendidos en el periodo 2008 a 2019, y poseen una distribución diversa en cuanto a los niveles de las restantes variables.

Tabla 3. Características principales de los estudios seleccionados.

Autor.Year	Comparado.Con	Participantes.Universitarios	Duracion.Semestral	Región
(Yukselturk & Altok, 2017)	Tradicional	Sí	Sí	Eurasia
(Cetin, 2016)	Otros	Sí	No	Eurasia
(Zaranis, et al., 2016)a	Otros	No	Sí	Eurasia
(Zaranis, et al., 2016)b	Tradicional	No	Sí	Eurasia
(Cardenas-Cobo, et al., 2019)a	Tradicional	Sí	Sí	América
(Cardenas-Cobo et al., 2019)b	Otros	Sí	Sí	América
(Erol & Kurt, 2017)	Tradicional	No	Sí	Eurasia
(Wang, Huang & Hwang, 2014)	Tradicional	No	No	Eurasia
(De Kereki, 2008)	Tradicional	Sí	No	América
(Nam, Kim & Lee, 2010)	Otros	No	No	Eurasia

En las Figuras 1, 2 y 3 se puede observar las diferentes distribuciones que siguen los resultados recopilados de acuerdo a las variables moderadoras.

Una vez descritas las principales características de los estudios en función de las variables definidas, a continuación, se buscarán respuestas a las preguntas que guían a la investigación a través de un meta-análisis sobre diferencias de medias estandarizadas. Esto supone una comparación entre los niveles de aprendizaje del grupo de control y el grupo experimental (Scratch).

Es importante tener en cuenta que en el meta-análisis basado en la diferencia de medias estandarizadas, el tamaño del efecto está definido entorno al 0. En este caso, este intervalo tendría la siguiente interpretación: un valor superior a 0 (cero) indicaría que el impacto de Scratch en el aprendizaje de programación es alto, mientras que un valor inferior a 0, lo contrario.

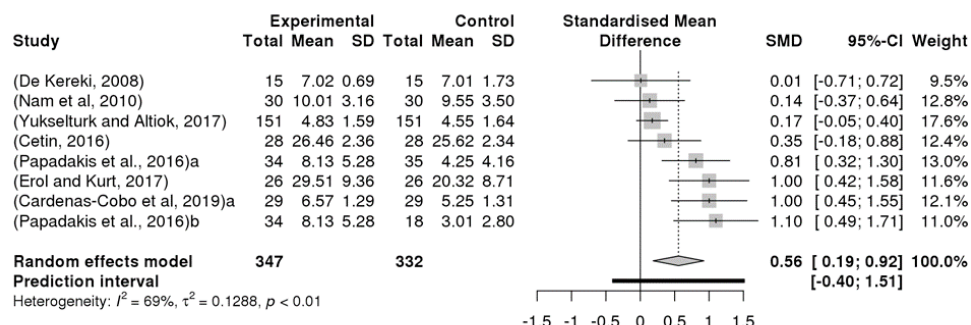


Figura 1. Gráfico de bosque impacto de Scratch en el aprendizaje de la programación.

Al ajustar un modelo de efectos aleatorios teniendo en cuenta los estudios descritos en la Tabla 3, la diferencia de medias estandarizada (impacto de Scratch en el aprendizaje de programación) fue de 0.32, con un intervalo de confianza de [-0.14; 0.78] al 95%. Esto indica que, de manera general, el impacto de Scratch sobre el aprendizaje de la programación es positivo. No obstante, el grado de heterogeneidad asociado a este resultado fue, de acuerdo al estadígrafo ($I^2 = 83\%$). El p-valor asociado a la prueba de heterogeneidad Q fue inferior a 0.01, lo cual indica que la heterogeneidad no solo es alta, sino también significativa. Como consecuencia, el efecto global calculado anteriormente no puede considerarse homogéneo y, por tanto, se debe proceder con un análisis que explique el origen de tal heterogeneidad.

Un análisis *leave one out* mostró que los estudios con mayor valor absoluto del *residuo estudentizado* fueron Wang, et al. (2014); y Cardenas-Cobo, et al. (2019). Estos trabajos son los que más influyen en la heterogeneidad y en el resumen del tamaño del efecto. Si no se tienen en cuenta a estos trabajos, un nuevo cálculo del tamaño del efecto global da como resultado 0,56, con un intervalo de confianza [0,19; 0,92] al 95%, y una heterogeneidad menor ($I^2 = 69\%$).

Como elemento adicional, se analizó la influencia de las variables moderadoras en el efecto global. En la Tabla 4 se muestra un resumen de estos análisis por cada una de las variables. En la columna Q_M se reporta el estadígrafo relacionado con la prueba de los moderadores realizada a cada variable en cuestión. La columna siguiente contiene el p-valor correspondiente a la prueba de moderadores. Finalmente, en la última columna se reporta la heterogeneidad residual que puede explicarse debido a la partición del conjunto de estudios según los niveles de la variable en cuestión.

Tabla 4. Análisis de las variables moderadoras.

Variable	Q_M	p-valor	I^2 (%)
Comparado.Com	0,31	0,58	45,00%
Participantes.Universitarios	0,62	0,43	69.34%
Duracion.Semestral	4,50	0,03	53.88%
Region	0,31	0,58	68.16%

Como se aprecia en la Tabla 4, solo la variable *Duracion.Semestral* resulta significativa como moderadora (p -valor = 0.03), siendo la heterogeneidad residual mucho menor que en el resto (53.88%). El gráfico de bosque de la Figura 2 ilustra cómo quedaría particionado el conjunto de estudios según esta variable.

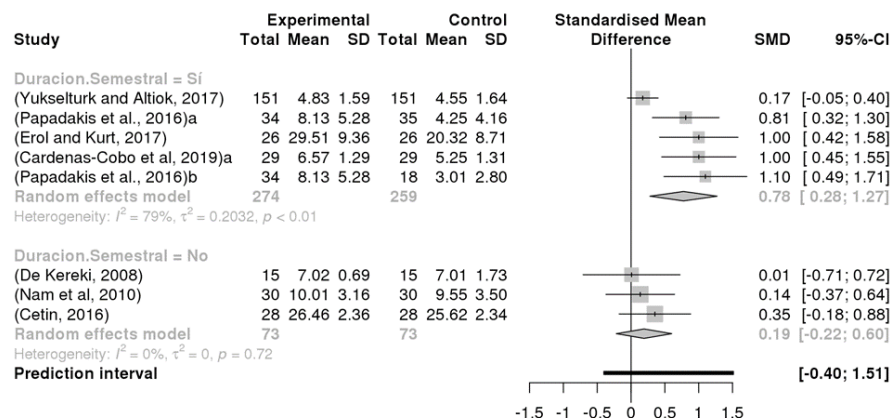


Figura 2. Gráfico de bosque correspondiente a la variable *Duracion.Semestral*.

Con el objetivo de identificar posibles sesgos de publicación en la muestra de estudios recolectados, estos fueron representados en un gráfico de embudo (Figura 3). En dicho gráfico se ilustra cómo se distribuyen de acuerdo a la precisión (error estándar) y su tamaño del efecto estimado. La correspondiente prueba de asimetría de Egger dio como resultado un valor z igual a 2.53, con un p -valor asociado de 0.12488. Lo cual indica que no existe asimetría en el modelo y que, por tanto, no se aprecia sesgo de publicación en el conjunto de estudios seleccionados.

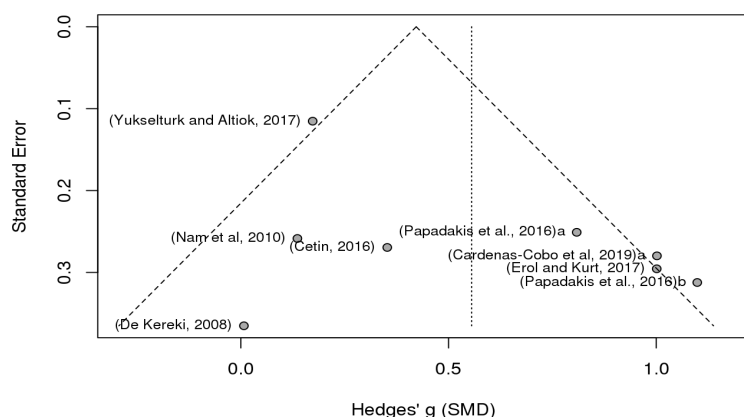


Figura 3. Test de Egger sin estudios con mayor valor absoluto.

CONCLUSIONES

El impacto global resumido de Scratch en el aprendizaje de programación manteniendo todos los estudios es de 0,32 con un intervalo de confianza de [-0,14; 0,78] al 95%. Este valor indica que dicho impacto no es muy alto y no se considera homogéneo. Por esta razón se realizó un análisis *leave one out* el cual permitió identificar dos estudios atípicos. En el cálculo sin estos estudios se obtuvo un SMD de 0,56, con un intervalo de confianza del 95%, [0,19; 0,92]. Como se puede apreciar, el valor del SMD aumentó con este resultado y se puede inferir que el impacto que tiene Scratch en el aprendizaje de la programación es positivo.

No se puede afirmar que exista una diferencia significativa estadísticamente entre los niveles educativos. No obstante, es importante destacar que el resultado para el caso universitario fue SMD = 0.38. Mientras que los resultados que se

obtuvieron para el resto de los niveles educativos fueron superiores ($SMD = 0.74$). Estos resultados permiten inferir que Scratch tendría un mayor impacto en los niveles educativos inferiores al universitario. Esto es consistente con la finalidad con la que fue concebido Scratch: potenciar el aprendizaje de la programación en niños y adolescentes.

Por medio del test de Egger se pudo apreciar que no existe una asimetría en la distribución de los trabajos. En otras palabras, existe una gran heterogeneidad de resultados tanto en valores positivos como negativos.

Finalmente, es importante indicar que, independientemente de las respuestas y conclusiones derivadas de la presente investigación, es importante advertir que las mismas deben interpretarse con precaución debido a la baja cantidad de estudios considerados en el meta-análisis.

Los resultados alcanzados hasta aquí poseen limitaciones, y, por tanto, deben interpretarse con precaución. La principal limitación tiene que ver con la baja cantidad de trabajos que fueron considerados para el meta-análisis. No obstante, los resultados brindan una panorámica general del impacto de Scratch en el aprendizaje de programación, y sugieren que puede ser implementado con éxito en entornos educativos inferiores al universitario y con una duración semestral.

REFERENCIAS BIBLIOGRÁFICAS

- Armería Zavala, L., Hernández Gallardo, S. C., & García-Ruiz, M. Á. (2013). Designing interactive activities within Scratch 2.0 for improving abilities to identify numerical sequences. IDC '13: Proceedings of the 12th International Conference on Interaction Design and Children, 423–426.
- Cardenas-Cobo, J., Puris, A., Novoa-Hernández, P., Galindo, J. A., & Benavides, D. (2019). Recommender Systems and Scratch: An integrated approach for enhancing computer programming learning. *IEEE Transactions on Learning Technologies*, 1–1.
- Cetin, I. (2016). Preservice Teachers' Introduction to Computing. *Journal of Educational Computing Research*, 54(7), 997–1021.
- 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.
- De Kereki, I. F. (2008). Scratch: Applications in Computer Science 1. In *2008 38th Annual Frontiers in Education Conference* (pp. T3B-7-T3B-11). IEEE.
- 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.
- Ke, F. (2014). An implementation of design-based learning through creating educational computer games: A case study on mathematics learning during design and computing. *Computers and Education*, 73, 26–39.
- Malan, D. J., & Leitner, H. H. (2007). Scratch for budding computer scientists. En, K. Y. Covington (Ed.), *Proceedings of the 38th SIGCSE technical symposium on Computer science education - SIGCSE '07* (Vol. 39, p. 223). ACM Press.
- Moher, D., Liberati, A., Tetzlaff, J., & Altman, D. G. (2009). Preferred Reporting Items for
- Moreno-Leon, J., & Robles, G. (2016). Code to learn with Scratch? A systematic literature review. *IEEE Global Engineering Education Conference, EDUCON*, 150–156.
- Nam, D., Kim, Y., & Lee, T. (2010). The Effects of Scaffolding-Based Courseware for the Scratch Programming Learning on Student Problem Solving Skill. In *ICCE2010*, 723–727.
- Wang, H. Y., Huang, I., & Hwang, G. J. (2014). Effects of an Integrated Scratch and Project-Based Learning Approach on the Learning Achievements of Gifted Students in Computer Courses. In *2014 IIAI 3rd International Conference on Advanced Applied Informatics*. IEEE, 382–387.
- Yukselturk, E., & Altioek, S. (2017). An investigation of the effects of programming with Scratch on the preservice IT teachers' self-efficacy perceptions and attitudes towards computer programming. *British Journal of Educational Technology*, 48(3), 789–801.
- Zaranis, N., Orfanakis, V., Papadakis, S., & Kalogiannakis, M. (2016). Using Scratch and App Inventor for teaching introductory programming in Secondary Education. A case study. *International Journal of Technology Enhanced Learning*, 1(1).