



REVISTA DE FILOSOFÍA

I. ÉTICA, GLOBALIDAD CRÍTICA Y BIENESTAR HUMANO

II. DIMENSIÓN EPISTÉMICA Y DESARROLLOS CULTURALES

*III. LA EDUCACIÓN EN CONTEXTO INTERCULTURAL Y
DECOLONIAL*

*IV. REPENSAR LA EDUCACIÓN SUPERIOR: TEORÍAS Y
PRÁCTICAS*

Universidad del Zulia
Facultad de Humanidades y Educación
Centro de Estudios Filosóficos
"Adolfo García Díaz"
Maracaibo - Venezuela

Nº 99
2021-3
Septiembre-Diciembre

Revista de Filosofía
Vol. 38, N°99, (Sep-Dic) 2021-3, pp. 369 - 379
Universidad del Zulia. Maracaibo-Venezuela
ISSN: 0798-1171 / e-ISSN: 2477-9598

Carencias en la evaluación del pensamiento computacional

shortcomings in the evaluation of computational thinking

Jesús Pérez

ORCID: <https://orcid.org/0000-0002-6585-2648>
Universidad de Los Andes
Mérida – Venezuela
Perezj89@gmail.com

José Castro

ORCID: <https://orcid.org/0000-0001-6353-9941>
Universidad de los Andes
Mérida - Venezuela
castroj@ula.ve

Oriana Pedroza

ORCID: <https://orcid.org/0000-0001-9026-4824>
Universidad de los Andes
Mérida – Venezuela
Oriana.p@ula.ve

Este trabajo está depositado en Zenodo:
DOI: <https://doi.org/10.5281/zenodo.5651282>

Resumen

El pensamiento computacional es una competencia que favorece la resolución de problemas tanto en ciencias de la computación como en las demás disciplinas y la vida cotidiana. Actualmente, es de interés incluir el pensamiento computacional en los currículos de todos los niveles educativos para mejorar el desempeño profesional en todas las disciplinas, sin embargo, todavía hay carencias, tales como la ausencia de herramientas de evaluación en contextos reales y la inexistencia de modelos de aplicación orientados a cada disciplina. El propósito de este ensayo es mostrar los argumentos que llevaron a identificar estas carencias. Primero, se presenta una discusión sobre la definición del pensamiento computacional. Luego, se introduce el marco de referencia que se utiliza comúnmente para diseñar evaluaciones en ciencias de la computación, y después, se compara con un marco de referencia utilizado para evaluaciones en el área de la medicina, en aras de contextualizar el origen de las carencias en la evaluación del pensamiento computacional. El análisis sugiere que las carencias son causadas por la utilización de la taxonomía de Bloom como marco de referencia.

Palabras clave: evaluación; competencia; pensamiento computacional; resolución de problemas.

Recibido 30-08-2021 – Aceptado 27-10-2021

Esta obra está bajo una licencia de Creative Commons Atribución-CompartirIgual 4.0 Internacional
(CC BY-SA 4.0)

<https://creativecommons.org/licenses/by/4.0/deed.es>

Abstract

Computational thinking is a competency that benefits problem solving as in computer sciences as in the other disciplines and everyday life. Currently, it is of interest to include computational thinking in the curricula of all educational levels, however, there are shortcomings such as the lack of evaluation tools in real contexts and the nonexistence of application models oriented on each discipline. This essay's goal is to show the arguments that support these shortcomings. First, we present a discussion about the computational thinking concept. Later, we introduce a common framework used to design evaluations in computer sciences. After, to contextualize the causes of the evaluation's shortcomings on computational thinking, we compare with a framework used in medicine. The analysis suggests that the shortcomings are caused by the utilization of Bloom's taxonomy as the framework.

Keywords: evaluation; skills; computational thinking; problem solving.

1. Introducción

El pensamiento computacional ha surgido por la necesidad de programar las computadoras, permitiendo nuevas posibilidades para las diferentes disciplinas. Aunque inicialmente estaba destinado para las ciencias de la computación, actualmente es de interés para profesionales en las demás disciplinas. Esto se debe a que el pensamiento computacional, tal como lo expresan Grover, et al. (2017), “es reconocido como un medio para la resolución creativa de problemas en otras disciplinas”.

Desde las ciencias de la computación, el pensamiento computacional beneficia a los estudiantes principalmente en tres aspectos: primero, permite la resolución de problemas mediante la computación; segundo, enseña a identificar los problemas que se pueden resolver; y tercero, impulsa a investigar modelos computacionales para situaciones que tradicionalmente no están relacionadas con la computación. De manera general, conforme con Mohaghegh y McCauley (2016), “el pensamiento computacional beneficia al estudiante en cualquier disciplina”.

El enfoque de este ensayo atiende a la idea de que el pensamiento computacional favorece a todas las disciplinas. Esta idea es consistente con lo expresado por Pérez (2019a): “el pensamiento computacional es fundamental por su carácter transdisciplinario en la resolución de problemas”. En ese sentido, ha nacido el interés por incorporar formalmente el pensamiento computacional en los currículos de todos los niveles educativos. Al respecto, González y Muñoz-Repiso (2020) mencionan que: “los países de gran desarrollo han decidido modificar sus programas de estudio oficial adoptando estrategias que faciliten el aprendizaje del pensamiento computacional en los distintos niveles educativos”.

De esta forma, la incorporación del pensamiento computacional se convierte en un tema de interés para las ciencias de la educación. En ese sentido, Grover, et al. (2017)

mencionan que: “el pensamiento computacional es reconocido como una competencia fundamental para el éxito en las carreras STEM (Ciencias, Tecnología, Ingeniería y Matemáticas)”. Además, el pensamiento computacional se considera como una competencia del siglo XXI, cuyos beneficios también se extienden a la vida cotidiana, implicando mejoras en la calidad de vida.

En ese orden de ideas, se considera que el pensamiento computacional es una competencia propia del ámbito de la computación, y que además puede ser aplicada tanto en otras disciplinas como en la vida cotidiana. Al respecto, Wing (2006) menciona que el pensamiento computacional “representa una actitud y un conjunto de habilidades universalmente aplicables que todos, no sólo científicos informáticos, estarían ansiosos por aprender y usar”. Desde esta perspectiva, la adquisición del pensamiento computacional se lleva a cabo en el proceso de aprendizaje de la programación de computadoras; este proceso es ampliamente conocido por las dificultades que presentan los estudiantes.

En concordancia con las características de los nativos digitales expuestas por Pérez (2019b), se han elaborado varias propuestas que ofrecen alternativas orientadas a facilitar el aprendizaje de la programación de computadores. Entre las propuestas se incluyen: un juego para estimular el razonamiento lógico (Pérez y Castro, 2018a), una metodología de estudio (Pérez y Pedroza, 2018), la inclusión de un robot social en el salón de clases (Pérez y Castro, 2018b), y una estrategia amistosa de aprendizaje orientada a la motivación (Pérez y Azuaje, 2019). Aunque cada propuesta genera efectos positivos en los estudiantes, ninguna representa una solución definitiva. Por lo tanto, es oportuno investigar otras aproximaciones.

La revisión de aproximaciones orientadas a desarrollar el pensamiento computacional a través de la programación de computadoras ha permitido encontrar las estrategias utilizadas actualmente para evaluarlo. El análisis de estas estrategias muestra carencias relacionadas con el interés de impulsar este pensamiento como una competencia útil para el desempeño profesional tanto en las ciencias de la computación como en otras disciplinas.

El propósito de este ensayo es mostrar argumentos que soportan la creencia de que el marco de referencia para la evaluación del pensamiento computacional ha promovido carencias. La sección de desarrollo se organiza de la siguiente manera: primero, se presenta una discusión sobre la definición del pensamiento computacional; segundo, se muestra el marco de referencia que se utiliza comúnmente para diseñar la evaluación en ciencias de la computación, resaltando sus carencias; y tercero, se introduce un marco de referencia para la evaluación del área de medicina, en aras de comparar y resaltar las carencias presentadas.

2. Desarrollo

En la teoría del pensamiento computacional hay varias definiciones que abordan la misma idea, pero desde distintas perspectivas. Román-González, et al. (2017) distinguen tres tipos de definiciones: genérica, operacional, y educacional. La definición *genérica* es

presentada por la precursora del pensamiento computacional Wing (2006), afirmando que: “involucra resolución de problemas, diseño de sistemas, y comprensión del comportamiento humano, basado en conceptos fundamentales de las ciencias de la computación”. Esta definición presenta la premisa principal de aplicar los fundamentos de la computación para todo.

Por su parte, Aho (2012) presenta una definición *operacional*: “es el proceso de pensamiento involucrado en la formulación de problemas, por lo que sus soluciones se pueden representar como pasos y algoritmos computacionales”. Esta definición hace énfasis en el proceso de resolución de problemas, implicando dos capacidades fundamentales: la formulación del problema, y la representación de la solución. En estas capacidades se deben aplicar los conceptos de la computación, es decir, cuando no se usan conceptos computacionales para la resolución de un problema, no se aplica el pensamiento computacional.

Por otra parte, Sondakh, et al. (2020) presentan una definición *educacional* que relaciona al pensamiento computacional con once conceptos: abstracción, pensamiento algorítmico, automatización, descomposición, depuración, evaluación, generalización, resolución de problemas, trabajo en equipo, comunicación, e inteligencia espiritual. A excepción de los últimos tres, estos conceptos son bien conocidos en ciencias de la computación.

Dado que, según Pérez (2021), los estudiantes universitarios perciben el pensamiento computacional como “el uso de conceptos que se aprenden en el ámbito de la informática para mejorar la resolución de problemas o la realización de actividades tanto en contextos universitarios como en la vida cotidiana”, en este ensayo se entiende al pensamiento computacional como el uso de los conceptos básicos de la computación para la resolución de problemas en diferentes disciplinas y la vida cotidiana. La discusión que se realiza está enfocada en la aplicación del pensamiento computacional en el ámbito profesional, sin embargo, la aplicación en la vida cotidiana pueda ser incluida con los mismos argumentos.

En el diseño de evaluaciones en ciencias de la computación es común encontrar, como marco de referencia, la aplicación de la dimensión cognitiva de la taxonomía revisada de Bloom. Esta taxonomía ofrece los objetivos que deben lograrse con la educación formal, los cuales están clasificados en seis niveles: recordar, comprender, aplicar, analizar, evaluar, y crear. De acuerdo con Thompson, et al. (2008), cada nivel es aplicado en ciencias de la computación. A continuación, se presentan ejemplos de aplicación de cada uno de los niveles.

El nivel *recordar* consiste en recuperar conocimiento de la memoria. Este nivel se alcanza cuando se es capaz de: recordar un concepto, identificar un constructo de un algoritmo, reconocer la implementación de un concepto, o reconocer la descripción apropiada de un término. El nivel *comprender* se relaciona con la construcción de

significados, como, por ejemplo, convertir un algoritmo de una representación a otra, explicar un concepto, o presentar un ejemplo de un algoritmo.

El nivel *aplicar* se refiere a la utilización de algún procedimiento en una situación dada, como, por ejemplo, utilizar un algoritmo conocido en un contexto familiar donde cambian los datos, o aplicarlo a un contexto desconocido. El nivel analizar consiste en descomponer algo en sus partes constituyentes para determinar cómo se relacionan entre ellas y cómo forman el todo. Por ejemplo, descomponer un programa en módulos, organizar módulos para cubrir un objetivo, identificar componentes críticos de desarrollo, e identificar componentes irrelevantes.

El nivel *evaluar* consiste en hacer juicios basados en criterios y estándares, como, por ejemplo, determinar si un algoritmo satisface los requerimientos, o criticar su calidad. Finalmente, el nivel crear consiste en unir elementos de forma coherente o funcional. Por ejemplo, crear un algoritmo que permita solucionar un problema, o encontrar una nueva estrategia para aplicar algoritmos conocidos.

Para mostrar el estado del arte de la evaluación del pensamiento computacional, se presenta un ejemplo de cada tipo según la clasificación realizada por Román-González, et al. (2019): herramientas de diagnóstico, herramientas formativas-iterativas, herramientas de procesamiento de datos, herramientas de transferencia de habilidades, herramientas de evaluación sumativa, escalas de percepciones-actitudes, y evaluación de vocabulario.

Las herramientas de diagnóstico se enfocan en medir el nivel de aptitud del estudiante y se pueden emplear con estudiantes sin experiencia previa. Este tipo se puede aplicar antes y después de una intervención educativa para verificar si el nivel de aptitud ha tenido algún cambio. Además, puede ser usado en cualquier área, tanto de forma presencial como no presencial. Un ejemplo de este tipo lo presenta Román-González (2015) a través de un cuestionario vía Internet de selección múltiple. Como su enfoque es diagnosticar, por lo general este tipo de herramientas proporcionan información sobre cómo los estudiantes recuerdan y comprenden los conceptos, es decir, este tipo cubre los niveles recordar y comprender de la taxonomía de Bloom.

Las herramientas formativas-iterativas se enfocan en proporcionar retroalimentación al estudiante, generalmente de forma automática, para desarrollar y mejorar sus habilidades. Este tipo no evalúa a la persona, sino a sus productos de aprendizaje, que generalmente son proyectos de programación creados tanto dentro como fuera de las aulas de clases. Un ejemplo de este tipo lo presentan Moreno-León, et al. (2015) con Dr. Scratch, el cual permite evaluar proyectos construidos con la herramienta de programación visual por bloques Scratch, en términos de las habilidades del pensamiento computacional utilizadas en su elaboración. Dado que los proyectos son productos que requieren predominantemente la unión de elementos de forma funcional, este tipo de herramientas se ubican en el nivel crear de la taxonomía de Bloom.

Las herramientas de procesamiento de datos están enfocadas en el proceso de aprendizaje, por lo tanto, registran y recuperan la actividad del estudiante en tiempo real. Este tipo de herramientas proporcionan datos valiosos para el análisis del aprendizaje que permite inferir los procesos cognitivos del sujeto, y son especialmente útiles para detectar conceptos computacionales erróneos. Por ejemplo, Grover, et al. (2017) interpretan los registros (tipos de acciones llevadas a cabo, cantidad de veces que se repite una acción, entre otros) de los estudiantes para medir su nivel de aprendizaje y comprensión del proceso. En este tipo de herramientas predomina el nivel aplicar de la taxonomía de Bloom porque se hace énfasis al procedimiento.

Las herramientas de transferencia de habilidades se enfocan en evaluar en qué medida los estudiantes pueden transferir sus habilidades a diferentes tipos de problemas, contextos y situaciones. Este tipo de herramientas pueden ser diseñadas para uso digital o desconectado. Un ejemplo lo representan los ejercicios de Bebras (Dagiene, y Futschek, 2008), los cuales se centran en medir la transferencia de habilidades del pensamiento computacional a planteamientos de problemas de la vida real. Los planteamientos pueden incluir una gran variedad de problemas que requieren aplicar uno o varios de los niveles de la taxonomía, dependiendo de su complejidad. Por lo tanto, este tipo de herramientas abarca todos los niveles de la taxonomía de Bloom.

Las herramientas de evaluación sumativa se enfocan en evaluar si el estudiante ha logrado aprender lo suficiente del contenido y es capaz de desempeñarse adecuadamente después de recibir alguna capacitación. El principal uso de este tipo es posterior a una intervención educativa. Un ejemplo lo presentan Meerbaum-Salant y McCauley (2013), a través de un cuestionario especialmente diseñado para medir el aprendizaje mediante Scratch. Este tipo de herramientas, al igual que el tipo anterior, puede abarcar todos los niveles de la taxonomía de Bloom según los problemas que se utilicen.

Las escalas de percepciones-actitudes están enfocadas en evaluar las percepciones y las actitudes de los estudiantes. En este tipo no sólo se abarca el pensamiento computacional, sino que también se abarcan temas relacionados como las computadoras, la informática, la programación de computadoras, o la alfabetización digital. Estas escalas pueden ser aplicadas antes o después de una intervención educativa. Un ejemplo lo presentan Korkmaz, et al. (2017), mediante la incorporación de ítems sobre pensamiento crítico, resolución de problemas, y aprendizaje cooperativo, para estudiar el nivel de habilidades del pensamiento computacional de los estudiantes. Por lo general, este tipo de herramientas no evalúan procesos cognitivos, por lo tanto, no se incluyen en la taxonomía de Bloom.

La evaluación de vocabulario está relacionada con medir varios elementos sobre el uso del lenguaje cuando el estudiante se expresa verbalmente. En este enfoque el lenguaje es esencial para el desarrollo del pensamiento computacional. Grover (2011) presenta un ejemplo donde evalúan el lenguaje de los estudiantes para determinar su apropiación del pensamiento computacional. Dado que este tipo de herramientas no se enfocan

precisamente en la aplicación del pensamiento computacional, sino en la expresión de los estudiantes, tampoco se incluye en la taxonomía de Bloom.

De manera general, se puede apreciar que tanto las herramientas de transferencia de habilidades como las herramientas de evaluación sumativa, pueden abarcar todos los niveles de la taxonomía de Bloom porque permiten la posibilidad de incluir cualquier tipo de problema. A diferencia, las herramientas de diagnóstico, formativas-iterativas, y de procesamiento de datos, predominan en alguno de los niveles: recordar y comprender, crear, y aplicar, respectivamente. Por otro lado, las herramientas de escalas de percepciones-actitudes y evaluación de vocabulario no se incluyen en la taxonomía de Bloom porque no están orientadas a los procesos cognitivos. Es importante mencionar que, aunque Román-González, et al. (2019) proporcionan una apreciación diferente, no se discute ni se compara porque las diferencias relacionadas con la ubicación de los tipos de herramientas en la taxonomía de Bloom no son relevantes para el objetivo de este ensayo.

Las herramientas presentadas para evaluar el pensamiento computacional no están diseñadas para ser utilizadas en contextos reales, sino que son propuestas a través de enunciados y simulaciones. Este hecho es inconsistente con la idea de fomentar el pensamiento computacional en aras de mejorar el desempeño profesional en las demás disciplinas, porque no se considera este aspecto en la evaluación, sino que se apela a la esperanza de que el estudiante pueda transferir las habilidades por cuenta propia de manera satisfactoria. En este ensayo se atribuye esta debilidad al marco de referencia utilizado, es decir, a la taxonomía de Bloom. Para mostrar argumentos de este planteamiento, a continuación, se presenta un marco de referencia utilizado en la medicina, denominado pirámide de Miller.

La pirámide de Miller es muy utilizada para la evaluación de competencias en el área de la medicina. De acuerdo con Fernández (2011), esta pirámide abarca cuatro niveles: saber, saber cómo, demuestra cómo, y hace. El nivel *saber*, comprende el dominio de conocimientos teóricos, y se evalúa principalmente con preguntas escritas, siendo las preguntas de selección múltiple muy empleadas porque se consideran válidas y fiables. El nivel *saber cómo*, consiste en aplicar los conocimientos teóricos adquiridos como si se tuvieran que poner en práctica en un contexto particular con el objetivo de obtener una visión previa de la práctica profesional. Este nivel se evalúa con los clásicos exámenes orales y las pruebas escritas (cortas o largas).

El nivel *demuestra cómo*, se enfoca en una demostración cercana a la realidad de los conocimientos. Esta evaluación se basa fundamentalmente en simulaciones que intentan reproducir situaciones similares a las de la vida real, y permite que los observadores puedan analizar las actuaciones específicas que se pretenden evaluar. Finalmente, el nivel *hace*, tiene el objetivo de aplicar los conocimientos adquiridos en la práctica profesional, es decir, es una actuación real en una situación profesional de desempeño.

La pirámide de Miller tiene similitudes con la taxonomía de Bloom. El nivel *saber* es similar al nivel recordar porque ambos abarcan los conocimientos teóricos. Los niveles *saber cómo* y *demuestra cómo* se relacionan con los niveles restantes (comprender, aplicar, analizar, evaluar, y crear) de la taxonomía de Bloom porque se refieren a la aplicación del conocimiento. A diferencia, el nivel *hace* no se relaciona con los niveles de la taxonomía de Bloom, porque en este nivel se requiere una actuación real en una situación de desempeño profesional, y en la taxonomía de Bloom esto no está contemplado.

En el nivel *hace* de la pirámide de Miller, se considera que la evaluación en los niveles anteriores no es transferible automáticamente a escenarios clínicos en la vida real, y por esa razón es importante continuar con el proceso de evaluación en el espacio de trabajo. De acuerdo con Thampy, et al. (2019), esta evaluación habitualmente la realizan los profesores a través de la observación, según alguna lista de pasos secuenciales que incluye principalmente elementos como la recopilación de datos, resumen de las características del caso, generación de diagnósticos diferenciales, aplicación de conocimientos previos, y diagnóstico final.

En ese orden de ideas, si se quieren ofrecer los beneficios del pensamiento computacional a todas las disciplinas, también es necesaria la evaluación en un entorno real de desempeño profesional. Además, en cada disciplina se debe conocer cómo aplicarlo adecuadamente según el nivel de complejidad de las actividades que realizan. La cobertura de esas dos carencias podría aumentar la probabilidad de que el pensamiento computacional sea aplicado en las diferentes disciplinas. Finalmente, considerando que también se desean obtener los beneficios de este pensamiento en la vida cotidiana, es necesaria la evaluación de éste en la cotidianidad.

3. Conclusiones

El pensamiento computacional fue entendido en este ensayo como el uso de los conceptos básicos de la computación, que son aprendidos a través de la programación de computadoras, para la resolución de problemas tanto en ciencias de la computación como en las demás disciplinas, incluyendo a la vida cotidiana. Dada la inconsistencia con la idea de incorporar el pensamiento computacional en los distintos niveles educativos, con la ausencia de herramientas de evaluación para contextos reales y la inexistencia de modelos de aplicación orientados a cada disciplina, se presentaron argumentos que sugieren que la causa de estas carencias está relacionada con la utilización de la taxonomía de Bloom como marco de referencia para el diseño de las evaluaciones.

Por un lado, la ausencia de herramientas de evaluación para contextos reales promueve que la evaluación se limite a contextos netamente educativos. Por otro lado, la inexistencia de modelos de aplicación orientados a cada disciplina promueve que los profesionales apliquen el pensamiento computacional de acuerdo con sus propios criterios sin la posibilidad de recibir una guía o retroalimentación. Estas carencias ponen de manifiesto que los profesores deben asumir que las habilidades del pensamiento computacional son

transferibles automáticamente a contextos de la vida real, y que deben limitarse a mantener la esperanza de que los estudiantes apliquen adecuadamente el pensamiento computacional en otras disciplinas, en vez de orientarlos durante su ejecución.

La exposición de estas carencias en la evaluación del pensamiento computacional es importante como directriz para construir un marco de referencia más completo. La aplicación de la taxonomía de Bloom promueve que estas carencias no se consideren, es decir, guiarse netamente por esta taxonomía es limitante. Si bien es cierto que éstas se presentan como una debilidad de la taxonomía de Bloom, también es cierto que no se favorece por completo a la pirámide de Miller, porque ésta tiene deficiencias de otros tipos. No obstante, para el enfoque de este ensayo la pirámide de Miller sí considera la carencia que se ha detectado en la taxonomía de Bloom.

En ese orden de ideas, quizás sea suficiente con unir ambos enfoques para solventar las carencias presentadas en este ensayo. De manera general, los trabajos futuros deben dirigirse a proponer un modelo de referencia más completo que la taxonomía de Bloom, orientado a la evaluación del pensamiento computacional, y, además, promover el diseño de modelos para cada disciplina y la vida cotidiana.

Referencias bibliográficas

- Aho, A. (2012). Computation and computational thinking. *The Computer Journal*, 55(7), 832-835. <https://doi.org/10.1093/comjnl/bxs074>
- Dagiene, V., & Futschek, G. (2008). Bebras international contest on informatics and computer literacy: Criteria for good tasks. In *International Conference on Informatics in Secondary Schools-Evolution and Perspectives*, 19-30. https://doi.org/10.1007/978-3-540-69924-8_2
- Fernández, J. T. (2011). La evaluación de las competencias en contextos no formales: dispositivos e instrumentos de evaluación. *Revista de educación*, (354), 341-342. <https://doi.org/10.4438/1988-592X-RE-2011-354-018>
- González, Y. A. C., & Muñoz-Repiso, A. G. V. (2020). Fortaleciendo el pensamiento computacional y habilidades sociales mediante actividades de aprendizaje con robótica educativa en niveles escolares iniciales. *Píxel-Bit. Revista de Medios y Educación*, (58), 117-142. <https://doi.org/10.12795/pixelbit.75059>
- Grover, S. (2011). Robotics and engineering for middle and high school students to develop computational thinking. In *Annual Meeting of the American Educational Research Association, New Orleans, LA*.
- Grover, S., Basu, S., Bienkowski, M., Eagle, M., Diana, N., & Stamper, J. (2017). A framework for using hypothesis-driven approaches to support data-driven learning analytics in measuring computational thinking in block-based programming environments. *ACM Transactions on Computing Education (TOCE)*, 17(3), 1-25. <https://doi.org/10.1145/3105910>

- Korkmaz, Ö., Cakir, R., & Özden, M. Y. (2017). A validity and reliability study of the computational thinking scales (CTS). *Computers in Human Behavior*, 72, 558-569. <https://doi.org/10.1016/j.chb.2017.01.005>
- Meerbaum-Salant, O., Armoni, M., & Ben-Ari, M. (2013). Learning computer science concepts with scratch. *Computer Science Education*, 23(3), 239-264. <https://doi.org/10.1145/1839594.1839607>
- Mohaghegh, M., & McCauley, M. (2016). Computational Thinking: The Skill Set of the 21st Century. *International Journal of Computer Science and Information Technologies*, 7(3), 1524-1530. <https://www.researchbank.ac.nz/handle/10652/3422>
- Moreno-León, J., Robles, G., & Román-González, M. (2015). Dr. Scratch: Automatic analysis of scratch projects to assess and foster computational thinking. *RED. Revista de Educación a Distancia*, 15(46), 1-23. https://www.um.es/ead/red/46/moreno_robles.pdf
- Pérez, J. (2021). Percepción de estudiantes universitarios sobre el pensamiento computacional. *REDU. Revista de Docencia Universitaria*, 19(1), 111-127. <https://doi.org/10.4995/redu.2021.15491>
- Pérez, J. (2019a). El pensamiento computacional en la vida cotidiana. *Revista Científica*, 4(13), 293-306. <https://doi.org/10.29394/Scientific.issn.2542-2987.2019.4.13.15.293-306>
- Pérez, J. (2019b). DINADI: una estrategia para el diagnóstico de nativos digitales en el ámbito universitario. *Revista Paradigma*, 40(1), 56-75. <http://revistaparadigma.online/ojs/index.php/paradigma/article/view/719>
- Pérez, J. & Azuaje, M. (2019) LE1: una estrategia amistosa para un curso introductorio de programación. *Revista Educación En Ingeniería*, 14(28), 45-53. <https://doi.org/10.26507/rei.v14n28.998>
- Pérez, J., & Castro, J. (2018a). Estímulo del razonamiento lógico mediante el juego Millonario en C para la asignatura “Programación 1”. *Revista Tekhné*, 21(3), 8-15. <http://revistasenlinea.saber.ucab.edu.ve/temas/index.php/tekhne/article/view/3804>
- Pérez, J., & Castro, J. (2018b). LRS1: Un robot social de bajo costo para la asignatura “Programación 1”. *Revista Colombiana de Tecnologías de Avanzada*, 2(32), 68-77. http://revistas.unipamplona.edu.co/ojs_viceinves/index.php/RCTA/article/view/3028
- Pérez, J., & Pedroza, O. (2018). LM1: una metodología de estudio para la asignatura “Programación 1”. *Educere*, 22(73), 635-648. <http://erevistas.saber.ula.ve/index.php/educere/article/view/13496>

- Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. In *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)*, 2436-2444. <http://doi.org/10.13140/RG.2.1.4203.4329>
- Román-González, M., Moreno-León, J., & Robles, G. (2019). Combining Assessment Tools for a Comprehensive Evaluation of Computational Thinking Interventions. *Computational Thinking Education*, 79-98. https://doi.org/10.1007/978-981-13-6528-7_6
- Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the *Computational Thinking Test*. *Computers in Human Behavior*, 72, 678-691. <https://doi.org/10.1016/j.chb.2016.08.047>
- Sondakh, D. E., Osman, K., & Zainudin, S. (2020). A Proposal for Holistic Assessment of Computational Thinking for Undergraduate: Content Validity. *European Journal of Educational Research*, 9(1), 33-50. <https://doi.org/10.12973/eu-jer.9.1.33>
- Thampy, H., Willert, E. & Ramani, S. (2019). Assessing Clinical Reasoning: Targeting the Higher Levels of the Pyramid. *Journal of general internal medicine*, 34(8), 1631-1636. <https://doi.org/10.1007/s11606-019-04953-4>
- Thompson, E., Luxton-Reilly, A., Whalley, J. L., Hu, M., & Robbins, P. (2008). Bloom's taxonomy for CS assessment. In *Proceedings of the tenth conference on Australasian computing education*, 78, 155-161. <https://dl.acm.org/doi/10.5555/1379249.1379265>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>



UNIVERSIDAD
DEL ZULIA

REVISTA DE FILOSOFÍA

Nº 99-3 _____

*Esta revista fue editada en formato digital y publicada en octubre de 2021, por el **Fondo Editorial Serbiluz**, Universidad del Zulia. Maracaibo-Venezuela*

www.luz.edu.ve
www.serbi.luz.edu.ve
www.produccioncientificaluz.org