

PIXEL BIT

N° 69 ENERO 2024
CUATRIMESTRAL

e-ISSN:2171-7966

ISSN:1133-8482

Revista de Medios y Educación





FECYT 16A0023
Cada 4 años (último: 6 de octubre de 2014) (no comercial)
ISSN: 1133-8482. 28 de junio de 2024



PIXEL-BIT

REVISTA DE MEDIOS Y EDUCACIÓN

Nº 69 - ENERO- 2024

<https://revistapixelbit.com>



EDITORIAL
UNIVERSIDAD DE SEVILLA

EQUIPO EDITORIAL (EDITORIAL BOARD)

EDITOR JEFE (EDITOR IN CHIEF)

Dr. Julio Cabero Almenara, Departamento de Didáctica y Organización Educativa, Facultad de CC de la Educación, Director del Grupo de Investigación Didáctica. Universidad de Sevilla (España)

EDITOR ADJUNTO (ASSISTANT EDITOR)

Dr. Juan Jesús Gutiérrez Castillo, Departamento de Didáctica y Organización Educativa. Facultad de CC de la Educación, Universidad de Sevilla (España)

Dr. Óscar M. Gallego Pérez, Grupo de Investigación Didáctica, Universidad de Sevilla (España)

EDITORES ASOCIADOS

Dra. Urtza Garay Ruiz, Universidad del País Vasco. (España)

Dra. Ivanovna Milqueya Cruz Pichardo, Pontificia Universidad Católica Madre y Maestra. (República Dominicana)

Dra. Carmen Llorente Cejudo, Universidad de Sevilla (España)

CONSEJO METODOLÓGICO

Dr. José González Such, Universidad de Valencia (España)

Dr. Antonio Matas Terrón, Universidad de Málaga (España)

Dra. Cynthia Martínez-Garrido, Universidad Autónoma de Madrid (España)

Dr. Luis Carro Sancristóbal, Universidad de Valladolid (España)

Dra. Nina Hidalgo Farran, Universidad Autónoma de Madrid (España)

CONSEJO DE REDACCIÓN

Dra. María Puig Gutiérrez, Universidad de Sevilla. (España)

Dra. Sandra Martínez Pérez, Universidad de Barcelona (España)

Dr. Selín Carrasco, Universidad de La Punta (Argentina)

Dr. Jackson Collares, Universidades Federal do Amazonas (Brasil)

Dra. Kitty Gaona, Universidad Autónoma de Asunción (Paraguay)

Dr. Vito José de Jesús Carioca. Instituto Politécnico de Beja Ciências da Educação (Portugal)

Dr. Elvira Esther Navas, Universidad Metropolitana de Venezuela (Venezuela)

Dr. Angel Puentes Puente, Pontificia Universidad Católica Madre y Maestra. Santo Domingo (República Dominicana)

Dr. Fabrizio Manuel Sirignano, Università degli Studi Suor Orsola Benincasa (Italia)

Dra. Sonia Aguilar Gavira. Universidad de Cádiz (España)

Dra. Eloisa Reche Urbano. Universidad de Córdoba (España)

CONSEJO TÉCNICO

Dra. Raquel Barragán Sánchez, Grupo de Investigación Didáctica, Universidad de Sevilla (España)

Dr. Antonio Palacios Rodríguez, Grupo de Investigación Didáctica, Universidad de Sevilla (España)

Dr. Manuel Serrano Hidalgo, Grupo de Investigación Didáctica, Universidad de Sevilla (España)

Diseño de portada: Dña. Lucía Terrones García, Universidad de Sevilla (España)

Revisor/corrector de textos en inglés: Dra. Rubicelia Valencia Ortiz, MacMillan Education (México)

Revisores metodológicos: evaluadores asignados a cada artículo

CONSEJO CIENTÍFICO

Jordi Adell Segura, Universidad Jaume I Castellón (España)

Ignacio Aguaded Gómez, Universidad de Huelva (España)

María Victoria Aguiar Perera, Universidad de Las Palmas de Gran Canaria (España)

Olga María Alegre de la Rosa, Universidad de la Laguna Tenerife (España)

Manuel Área Moreira, Universidad de la Laguna Tenerife (España)

Patricia Ávila Muñoz, Instituto Latinoamericano de Comunicación Educativa (México)

Antonio Bartolomé Pina, Universidad de Barcelona (España)

Angel Manuel Bautista Valencia, Universidad Central de Panamá (Panamá)

Jos Beishuizen, Vrije Universiteit Amsterdam (Holanda)

Florentino Blázquez Entonado, Universidad de Extremadura (España)
Silvana Calaprince, Università degli studi di Bari (Italia)
Selín Carrasco, Universidad de La Punta (Argentina)
Raimundo Carrasco Soto, Universidad de Durango (México)
Zulma Cataldi, Universidad de Buenos Aires (Argentina)
Luciano Ceconi, Università degli Studi di Modena (Italia)
Jean-François Cerisier, Université de Poitiers, Francia
Jordi Lluís Coiduras Rodríguez, Universidad de Lleida (España)
Jackson Collares, Universidades Federal do Amazonas (Brasil)
Enricomaria Corbi, Università degli Studi Suor Orsola Benincasa (Italia)
Marialaura Cunzio, Università degli Studi Suor Orsola Benincasa (Italia)
Brigitte Denis, Université de Liège (Bélgica)
Floriana Falcinelli, Università degli Studi di Perugia (Italia)
Maria Cecilia Fonseca Sardi, Universidad Metropolitana de Venezuela (Venezuela)
Maribel Santos Miranda Pinto, Universidade do Minho (Portugal)
Kitty Gaona, Universidad Autónoma de Asunción (Paraguay)
María-Jesús Gallego-Arrufat, Universidad de Granada (España)
Lorenzo García Aretio, UNED (España)
Ana García-Valcarcel Muñoz-Repiso, Universidad de Salamanca (España)
Antonio Bautista García-Vera, Universidad Complutense de Madrid (España)
José Manuel Gómez y Méndez, Universidad de Sevilla (España)
Mercedes González Sanmamed, Universidad de La Coruña (España)
Manuel González-Sicilia Llamas, Universidad Católica San Antonio-Murcia (España)
António José Meneses Osório, Universidade do Minho (Portugal)
Carol Halal Orfali, Universidad Tecnológica de Chile INACAP (Chile)
Mauricio Hernández Ramírez, Universidad Autónoma de Tamaulipas (México)
Ana Landeta Etxeberria, Universidad a Distancia de Madrid (UDIMA)
Linda Lavelle, Plymouth Institute of Education (Inglaterra)
Fernando Leal Ríos, Universidad Autónoma de Tamaulipas (México)
Paul Lefrere, Cca (UK)
Carlos Marcelo García, Universidad de Sevilla (España)
Francois Marchessou, Universidad de Poitiers, París (Francia)
Francesca Marone, Università degli Studi di Napoli Federico II (Italia)
Francisco Martínez Sánchez, Universidad de Murcia (España)
Ivory de Lourdes Mogollón de Lujo, Universidad Central de Venezuela (Venezuela)
Angela Muschitiello, Università degli studi di Bari (Italia)
Margherita Musello, Università degli Studi Suor Orsola Benincasa (Italia)
Elvira Esther Navas, Universidad Metropolitana de Venezuela (Venezuela)
Trinidad Núñez Domínguez, Universidad de Sevilla (España)
James O'Higgins, de la Universidad de Dublín (UK)
José Antonio Ortega Carrillo, Universidad de Granada (España)
Gabriela Padilla, Universidad Autónoma de Tamaulipas (México)
Ramón Pérez Pérez, Universidad de Oviedo (España)
Angel Puentes Puente, Pontificia Universidad Católica Madre y Maestra. Santo Domingo (República Dominicana)
Juan Jesús Gutiérrez Castillo, Universidad de Sevilla (España)
Julio Manuel Barroso Osuna, Universidad de Sevilla (España)
Rosalía Romero Tena, Universidad de Sevilla (España)
Hommy Rosario, Universidad de Carabobo (Venezuela)
Pier Giuseppe Rossi, Università di Macerata (Italia)
Jesús Salinas Ibáñez, Universidad Islas Baleares (España)
Yamile Sandoval Romero, Universidad de Santiago de Cali (Colombia)
Albert Sangrà Morer, Universidad Oberta de Catalunya (España)
Ángel Sanmartín Alonso, Universidad de Valencia (España)
Horacio Santángelo, Universidad Tecnológica Nacional (Argentina)
Francisco Solá Cabrera, Universidad de Sevilla (España)
Jan Frick, Stavanger University (Noruega)
Karl Steffens, Universidad de Colonia (Alemania)
Seppo Tella, Helsinki University (Finlandia)
Hanne Wachter Kjaergaard, Aarhus University (Dinamarca)



FACTOR DE IMPACTO (IMPACT FACTOR)

SCOPUS Q1 Education: Posición 236 de 1406 (83% Percentil). CiteScore Tracker 2022: 5 - Journal Citation Indicator (JCI). Emerging Sources Citation Index (ESCI). Categoría: Education & Educational Research. Posición 257 de 739. Cuartil Q2 (Percentil: 65.29) - FECYT: Ciencias de la Educación. Cuartil 1. Posición 16. Puntuación: 35,68- DIALNET MÉTRICAS (Factor impacto 2021: 1.72. Q1 Educación. Posición 12 de 228) - REDIB Calificación Glogal: 29,102 (71/1.119) Percentil del Factor de Impacto Normalizado: 95,455- ERIH PLUS - Clasificación CIRC: B- Categoría ANEP: B - CARHUS (+2018): B - MIAR (ICDS 2020): 9,9 - Google Scholar (global): h5: 42; Mediana: 42 - Journal Scholar Metric Q2 Educación. Actualización 2016 Posición: 405a de 1,115- Criterios ANECA: 20 de 21 - INDEX COPERNICUS Puntuación ICV 2019: 95.10

Pixel-Bit, Revista de Medios y Educación está indexada entre otras bases en: SCOPUS, Fecyt, DOAJ, Iresie, ISOC (CSIC/CINDOC), DICE, MIAR, IN-RECS, RESH, Ulrich's Periodicals, Catálogo Latindex, Biné-EDUSOL, Dialnet, Redinet, OEI, DOCE, Scribd, Redalyc, Red Iberoamericana de Revistas de Comunicación y Cultura, Gage Cengage Learning, Centro de Documentación del Observatorio de la Infancia en Andalucía. Además de estar presente en portales especializados, Buscadores Científicos y Catálogos de Bibliotecas de reconocido prestigio, y pendiente de evaluación en otras bases de datos.

EDITA (PUBLISHED BY)

Grupo de Investigación Didáctica (HUM-390). Universidad de Sevilla (España). Facultad de Ciencias de la Educación. Departamento de Didáctica y Organización Educativa. C/ Pirotecnia s/n, 41013 Sevilla.
Dirección de correo electrónico: revistapixelbit@us.es . URL: <https://revistapixelbit.com/>
ISSN: 1133-8482; e-ISSN: 2171-7966; Depósito Legal: SE-1725-02
Formato de la revista: 16,5 x 23,0 cm

Los recursos incluidos en Pixel Bit están sujetos a una licencia Creative Commons Attribution-NonCommercial-ShareAlike 4.0 Unported (Reconocimiento-NoComercial-CompartirIgual)(CC BY-NC-SA 4.0), en consecuencia, las acciones, productos y utilidades derivadas de su utilización no podrán generar ningún tipo de lucro y la obra generada sólo podrá distribuirse bajo esta misma licencia. En las obras derivadas deberá, asimismo, hacerse referencia expresa a la fuente y al autor del recurso utilizado.

©2024 Pixel-Bit. No está permitida la reproducción total o parcial por ningún medio de la versión impresa de Pixel-Bit.

Índice

- 1.- **Competencia digital en labores de Investigación: predictores que influyen en función del tipo de universidad y género del profesorado de Educación Superior** // Digital competence in research work: predictors that have an impact on it according to the type of university and gender of the Higher Education teacher. Francisco David Guillén-Gámez, Melchor Gómez-García, Julio Ruiz-Palmero **7**
- 2.- **La percepción de las familias acerca de las tareas digitales** // Families' perceptions about digital homework. M^a Ángeles Hernández-Prados, José Santiago Álvarez-Muñoz, Juan Antonio Gil-Noguera **35**
- 3.- **Cuestionando el papel de las tecnologías en la Educación Infantil: brechas y falsas visiones** // Questioning the role of technology in Early Childhood Education: divides and false views. Patricia Digón-Regueiro, Rosa María Méndez-García, M. Mercedes Romero-Rodrigo, Cecilia Verónica Becerra-Brito **63**
- 4.- **Uso académico del smartphone en la formación de posgrado: percepción del alumnado en Ecuador** // Academic use of smartphones in postgraduate education: student perception in Ecuador. Alexander López-Padrón, Santiago Mengual-Andrés, Ernesto Andrés Hermann-Acosta **97**
- 5.- **Equivalencias entre los indicadores de la herramienta SELFIE y el marco DigCompEdu a partir de la técnica Delphi** // Equivalences between SELFIE indicators and DigCompEdu framework based on Delphi technique. Jacoba Munar-Garau, Jorge Oceja, Jesús Salinas-Ibáñez **131**
- 6.- **Análisis del nivel de pensamiento computacional de los futuros maestros: una propuesta diagnóstica para el diseño de acciones formativas** // Analysis of the level of computational thinking of future teachers: a diagnostic proposal for the design of training actions. Lourdes Villalustre-Martín **169**
- 7.- **Una propuesta de evaluación de Recursos Educativos Digitales a través de la metodología fsQCA longitudinal** // A proposal for the evaluation of Digital Educational Resources through the longitudinal fsQCA methodology. Cristina Mendaña-Cuervo, Nieves Remo-Diez, Enrique López-González **195**
- 8.- **El uso del modelo GPT de OpenAI para el análisis de textos abiertos en investigación educativa** // Use of GPT-3 for open text analysis in qualitative research. Héctor González-Mayorga, Agustín Rodríguez-Esteban, Javier Vidal **227**
- 9.- **Estrategias pedagógicas con TIC en contextos educativos vulnerables: Repercusiones en la pandemia y proyecciones de futuro** // Pedagogical strategies with ICT in vulnerable educational contexts: Impact on the pandemic and future projections. María José Alcalá del Olmo-Fernández, María Jesús Santos-Villalba, José Luis González-Sodis, Juan José Leiva-Olivencia **255**
- 10.- **Perfiles de uso problemático de los videojuegos y su influencia en el rendimiento académico y los procesos de toma de decisiones en alumnado universitario** // Profiles of Undergraduates' videogames problematic use: the influence of academic performance and decision-making process. Barreto-Cabrera, Yenia del Sol, Suárez-Perdomo, Arminda, Castilla-Vallejo, José Luis **287**

Análisis del nivel de pensamiento computacional de los futuros maestros: una propuesta diagnóstica para el diseño de acciones formativas

Analysis of the level of computational thinking of future teachers: a diagnostic proposal for the design of training actions

  **Dra. Lourdes Villalustre-Martínez**

Profesora Titular de Universidad. Universidad de Oviedo, España

Recibido: 2023/07/27; Revisado: 2023/09/05; Aceptado: 2023/11/23; Preprint: 2023/12/01; Publicado: 2024/01/07

RESUMEN

El pensamiento computacional supone una forma de alfabetización emergente que busca fomentar el aprendizaje de la programación de forma progresiva utilizando principios básicos de codificación informática. En este estudio se evaluó el pensamiento computacional de 164 estudiantes universitarios de los grados de maestro/a en educación infantil y primaria. Se examinaron las diferencias según el género y la experiencia previa en programación robótica. Para ello, se empleó el Test de Pensamiento Computacional (TPC). Los resultados revelan que los hombres obtuvieron mejores resultados y que la experiencia previa en programación influyó en el nivel de desarrollo del pensamiento computacional. Además, se identificaron tres perfiles de estudiantes mediante un análisis de clúster. Las mujeres con experiencia previa en programación robótica y el uso de lenguajes de programación mostraron los mejores resultados en el TPC. Estos hallazgos resaltan la importancia de realizar evaluaciones diagnósticas para conocer el nivel de competencia de los estudiantes en este ámbito, ya que puede ayudar a identificar áreas de mejora y adaptar las acciones formativas de acuerdo a las necesidades de cada grupo de estudiantes.

ABSTRACT

Computational thinking is an emerging form of literacy that seeks to foster the learning of programming in a progressive manner using basic principles of computer coding. This study assessed the computational thinking of 164 undergraduate students in early childhood and elementary education teaching degrees. Differences according to gender and previous experience in robotic programming were examined. For this purpose, the Test of Computational Thinking (TPC) was used. The results reveal that males obtained better results and that previous programming experience influenced the level of development of computational thinking. In addition, three student profiles were identified through a cluster analysis. Females with prior experience in robotic programming and the use of programming languages showed the best results on the TPC. These findings highlight the importance of performing diagnostic evaluations to know the level of competence of students in this area, as it can help identify areas for improvement and adapt training actions according to the needs of each group of students.

PALABRAS CLAVES · KEYWORDS

Pensamiento computacional, lógica programación; educación superior; género; experiencia previa.
Computational thinking, programming logic; higher education; gender; previous experience

1. Introducción

El pensamiento computacional implica un proceso de resolución de problemas aplicando conocimientos básicos de ciencia, tecnología, lógica y matemáticas (Tsarava et al., 2022; Wing, 2006). El término emergió del trabajo de Papert (1980) sobre ambientes de programación constructivistas basados en el diseño y la resolución de problemas según algoritmos. Se trata de una forma de alfabetización emergente que busca fomentar el aprendizaje de la programación de forma progresiva utilizando principios básicos de codificación informática. En este enfoque, los estudiantes de diferentes niveles educativos crean secuencias de instrucciones y algoritmos para controlar las acciones de un objeto en un espacio virtual o físico. Para Wang et al. (2022) el desarrollo del pensamiento computacional involucra diversas capacidades y habilidades, como la organización y análisis lógico de la información, la identificación, investigación e implementación de posibles soluciones, etc. al tiempo que se fortalece el pensamiento algorítmico.

El pensamiento algorítmico es una habilidad fundamental en la resolución de problemas, que implica concebir una solución paso a paso para abordar un determinado problema. Bers et al. (2014) establecen que, a diferencia de la codificación, que se refiere a las habilidades técnicas necesarias para escribir código en un lenguaje de programación específico, el pensamiento algorítmico se centra en el proceso conceptual de diseñar un algoritmo. En esta línea, para Selby (2012), el pensamiento algorítmico se distingue de la codificación y se considera una habilidad independiente y previa a ella. Implica la capacidad de planificar y organizar las acciones necesarias para resolver un problema de manera secuencial y efectiva. Estas habilidades son consideradas elementos importantes del pensamiento computacional y son fundamentales en el desarrollo de habilidades de programación y resolución de problemas computacionales (Majeed et al., 2022).

Existe un amplio consenso acerca de la estrecha relación entre las habilidades de pensamiento computacional y la programación. Diferentes autores (Tikva y Tambouris, 2021) sostienen que el pensamiento computacional ha brindado un nuevo marco lógico para la enseñanza de la programación. El pensamiento computacional implica una serie de habilidades cognitivas y de resolución de problemas que son fundamentales en el ámbito de la programación robótica. Estas habilidades incluyen el desglose de problemas complejos en pasos más simples, la identificación de patrones y regularidades, el diseño de algoritmos para resolver problemas y la capacidad de pensar de manera lógica y estructurada. La programación, por su parte, proporciona una plataforma práctica para que los estudiantes ejerciten y apliquen estas habilidades del pensamiento computacional en la resolución de problemas concretos.

Los estudios existentes han evidenciado que el aprendizaje y la práctica de la programación robótica contribuyen significativamente al desarrollo del pensamiento computacional en los estudiantes (Lu et al., 2022). Al participar en actividades de programación, los discentes son desafiados a analizar y comprender el funcionamiento de los sistemas informáticos, a identificar y resolver errores, y a mejorar sus soluciones. Estas experiencias fomentan habilidades como el razonamiento lógico, la abstracción, la creatividad y la colaboración, que son fundamentales tanto en el pensamiento computacional como en el ámbito de la programación (Popat y Starkey, 2019).

El desarrollo y mejora del pensamiento computacional asociado a la programación robótica ha experimentado una evolución creciente en los últimos años. Si bien, es cierto que

este avance no ha tenido la misma repercusión en hombres que en mujeres, y que existe una cierta disparidad en el interés hacia la programación robótica entre ambos grupos (Chan, 2022). Es importante reconocer que esta situación puede estar influenciada por factores culturales y procesos de socialización que se experimentan desde la infancia y que pueden motivar más a los hombres que a las mujeres a involucrarse en estos campos (Angeli y Valanides, 2020). No obstante, Charlesworth y Banaji (2019) abogan por fomentar una educación inclusiva y equitativa que amplíen los intereses y habilidades en estos campos, independientemente del género.

Poseer habilidades sólidas de pensamiento computacional puede ser un factor motivador para que los estudiantes se interesen por carreras científicas. Sin embargo, la falta de un currículo específico que sienta las bases del pensamiento computacional desde edades tempranas representa un desafío en la educación. Existen pocos modelos o enfoques que destaquen el pensamiento computacional en las prácticas educativas actuales (Repenning et al., 2010). A pesar de la importancia del pensamiento computacional, se ha prestado poca atención a su desarrollo en la formación de los futuros maestros y maestras. La mayoría de los trabajos de investigación priorizan su enseñanza en la etapa de educación secundaria obligatoria (Maya et al., 2015), dejando un vacío en la formación de los futuros docentes.

Para abordar esta situación, es necesario promover una mayor integración del pensamiento computacional en la formación de los futuros maestros y maestras. Esto implica la incorporación de contenidos y metodologías relacionadas con el pensamiento computacional en los programas de formación docente, así como el desarrollo de recursos y materiales adecuados para su enseñanza (Ung et al., 2022). Asimismo, es relevante tener en cuenta el nivel de pensamiento computacional que poseen los estudiantes. Realizar evaluaciones diagnósticas para conocer su nivel de competencia en este ámbito puede ayudar a identificar áreas de mejora y adaptar las acciones formativas de acuerdo a las necesidades de cada grupo de estudiantes. Además, es importante considerar variables como el género y la experiencia previa en programación robótica, ya que pueden influir en el desarrollo del pensamiento computacional (Sun et al., 2022). Esto permitirá ofrecer acciones formativas más adaptadas e inclusivas.

En esta línea, varios estudios han abordado la relación entre las habilidades de pensamiento computacional de los estudiantes y el género, así como con la experiencia previa en programación robótica. Así, el trabajo realizado por Atmatzidou y Demetriadis (2016) demostró que no hay diferencias significativas en función del género en el desarrollo de habilidades de pensamiento computacional entre estudiantes de 15 a 18 años que participaron en un programa de Robótica Educativa. Al igual que el estudio llevado a cabo por Bati (2022) con estudiantes de educación infantil. Estos hallazgos sugieren que el género no influye de manera significativa en el desarrollo de estas habilidades en esos contextos. Por su parte, Master et al. (2017) encontraron que aquellos estudiantes de educación primaria con experiencia en programación reportaron un mayor interés técnico y una mayor autoeficacia en comparación con aquellos que no tenían experiencia en programación robótica.

En definitiva, la mayoría de los estudios actuales que se han centrado en el análisis del género y la experiencia en programación en relación con el pensamiento computacional lo han realizado en estudiantes de niveles no universitarios. Sin embargo, no se ha explorado a fondo cómo estos factores pueden influir en los futuros docentes. Por lo tanto, se pone en

marcha el presente trabajo con el fin de caracterizar a los estudiantes de la muestra en cuanto al nivel de pensamiento computacional. Así como, analizar si existen diferencias estadísticamente significativas entre el nivel de pensamiento computacional en función del género y la experiencia previa en lenguajes de programación robótica. Y, por último, identificar la existencia de perfiles que permitan analizar las diferencias entre ellos con las dimensiones del pensamiento computacional, para obtener una comprensión más sólida de cómo estos factores pueden influir en el pensamiento computacional de los estudiantes, para generar acciones formativas.

2. Metodología

2.1. Participantes

En el estudio participaron un total de 164 estudiantes universitarios de los Grados en Maestro/a en Educación Infantil y Primaria, siendo 145 mujeres, lo que supone el 88,4% y 19 hombres. La edad de los participantes se sitúa en su mayoría (72.3%) entre los 18 y 21 años, seguida por aquellos con edades comprendidas entre los 22 y 25 años (23,1%), y entre 26 y 29 años con 4.6%. La elección de dicha muestra siguió un paradigma no probabilístico, siguiendo criterios de conveniencia y accesibilidad (Etikan et al., 2016).

2.2. Instrumento de análisis

Se emplea el Test de Pensamiento Computacional (TPC), validado por Román-González (2015), en cuanto a su contenido, fiabilidad y validez tanto criterial como concurrente. Dicho instrumento tiene como objetivo medir el pensamiento computacional de los sujetos. Está constituido por 28 preguntas con cuatro opciones de respuestas, de las cuales sólo una de ellas es correcta. En las preguntas se presentan diferentes supuestos en los que se deben aplicar habilidades de pensamiento lógico-matemático para su correcta resolución. El TPC ha sido traducido a varios idiomas y utilizado por la comunidad científica internacional en múltiples investigaciones (Chan et al., 2021; Guggemos et al., 2022).

Aunque el instrumento ha sido validado, se lleva a cabo un análisis de sus propiedades psicométricas. Así, la correlación ítem-total corregida (r_{i-t}) fue positiva en todos los ítems, lo que indica que todos contribuían a medir el constructo general que mide el instrumento y en la misma dirección. El Alpha de Cronbach fue adecuado (.705) siendo la puntuación media de los ítems de .76. La medida de Kaiser-Meyer-Olkin de adecuación de muestreo se situó en .599 y la prueba de esfericidad de Barlett fue significativa ($p < .001$) con un valor de chi-cuadrado de .599, con 231 grados de libertad.

Para lograr los objetivos planteados, se recopilaron tres datos adicionales. En primer lugar, se determinó el nivel de conocimientos previos en programación que los estudiantes consideran poseer, a través de una escala tipo Likert de 1 a 5 (donde 1 equivale a Muy Bajo y 5 a Muy Alto). En segundo lugar, se indagó si los estudiantes tenían experiencia previa en programación robótica a través de una pregunta dicotómica (Si o No). Por último, se utilizó el nivel de desempeño de los estudiantes universitarios como medida de referencia. Este nivel de desempeño se basó en la calificación obtenida por los discentes en la asignatura de Tecnologías de la Información y Comunicación (TIC) aplicadas a la Educación

de los Grados de Maestro/a en Educación Infantil y Primaria, ya que diversos autores consideran las calificaciones obtenidas como un indicador válido para medir el rendimiento académico (Álvaro Paje, 1990; Ben et al., 2022), en este caso, en el ámbito de las tecnologías.

2.3. Procedimiento

El estudio se realizó de acuerdo con el Código de Ética de la Asociación Médica Mundial (Declaración de Helsinki), que refleja los principios éticos para la investigación con seres humanos (Morris, 2013). Previamente a la realización del TPC, los estudiantes fueron informados de que su participación era voluntaria. Los participantes debían otorgar su consentimiento informado previamente a la realización del cuestionario.

Los estudiantes cumplieron el TPC de forma digital, accediendo al mismo a través del enlace enviado a sus correos corporativos de la Universidad antes de planificar y desarrollar las acciones formativas sobre programación robótica en el marco de la citada asignatura. Por tanto, se lleva a cabo un estudio exploratorio con el fin de determinar el estado inicial de los estudiantes de nuevo ingreso a los grados Maestro/a en Educación Infantil y Primaria para generar acciones formativas que apoyen su desempeño académico.

Una vez recogidas las respuestas del TPC se ha asignado con el valor 0 aquellas que han sido contestadas incorrectamente, y 1 para las correctas. Posteriormente, se han agrupado los ítems en función del tipo de habilidades que se requiere poner en práctica para su correcta resolución, dando lugar a cuatro dimensiones, a saber: 1) Algoritmos simples (AS), constituidos por siete supuestos de programación lineal; 2) Algoritmos de bucle (AB), formados por siete supuestos prácticos donde la secuencia de programación se repite varias veces; 3) Algoritmos condicionales (AC), constituido por siete supuestos basados en estipular determinadas condiciones básicas para que los bloques de programación se ejecuten y 4) Algoritmos condicionales avanzados (ACA), formado por siete supuestos y, al igual que la anterior, centrados en establecer condiciones avanzadas para resolver los problemas planteados. Finalmente, se ha ponderado las puntuaciones en cada una de las cuatro dimensiones, obteniendo puntuaciones de 0 a 10 en cada una de ellas.

2.4. Análisis de datos

En primer lugar, para analizar la existencia de diferencias en las cuatro dimensiones que componen el pensamiento computacional en función del género y la experiencia previa de los estudiantes en lenguajes de programación se realizaron pruebas t para muestras independientes.

Para obtener los perfiles de los estudiantes se utilizó un clúster bietápico. Este análisis proporciona un procedimiento automático del número óptimo de conglomerados, permitiendo conglomerados con variables categóricas y continuas (Rubio y Vilá, 2017). Las variables incluidas en el clúster fueron “género”, “experiencia previa en lenguajes de programación robótica”, “nivel de conocimientos que los estudiantes consideren poseer sobre programación” y “nivel de desempeño” medido de 0 a 10, y calculado a partir de la calificación final obtenida en la asignatura. El modelo cumple con los supuestos de

independencia entre sus variables, las variables continuas siguen una distribución normal y las variables cualitativas categóricas siguen una distribución multinomial. Incluso aunque no se cumplieran, Rubio y Vilá (2017) indican que este procedimiento es lo bastante robusto para poder aplicarlo. Para analizar la distancia entre los conglomerados se utilizó la medida de probabilidad log-verosimilitud. Para determinar el número de conglomerados se utilizó el Criterio Bayesiano de Schwartz (BIC). Para minimizar los efectos del orden, se ordenaron los casos aleatoriamente. Los resultados aportaron un modelo satisfactorio formado por tres conglomerados.

Por último, y una vez obtenidos los perfiles de los estudiantes, se utiliza un ANOVA de un factor con pruebas post hoc para analizar cada una de las cuatro dimensiones presentes en el pensamiento computacional (algoritmos simples, algoritmos de bucle, algoritmos condicionales y algoritmos condicionales avanzados) en función de dichos perfiles.

Los datos obtenidos se analizaron con el programa SPSS 27, teniendo como referencia para la significatividad un valor $p < .05$. Los tamaños del efecto se valoraron mediante la eta parcial al cuadrado teniendo en cuenta que el tamaño del efecto es pequeño cuando $\eta_p^2 = 0.01$; medio cuando $\eta_p^2 = 0.059$ y grande cuando $\eta_p^2 = 0.080$ (Cohen, 1988).

3. Análisis y resultados

En la tabla 1 se presentan los estadísticos descriptivos de las variables, así como la matriz de correlaciones de Pearson. Los valores de asimetría y curtosis se encuentran dentro de los límites que indican una distribución normal (valores inferiores a 3 para la asimetría e inferiores a 10 para la curtosis; Kline, 2011).

Tabla 1

Estadísticos descriptivos y correlación (r) entre las variables que conforman en pensamiento computacional

	AS	AB	AC	ACA
AS	-	-	-	-
AB	.333*	-	-	-
AC	.329*	.374*	-	-
ACA	.205*	.461*	.413*	-
Asimetría	-1.618	-0.692	-0.188	-0.269
Curtosis	2,242	0.163	-0.800	-0.698
Media	9.268	8.049	6.171	5.988
Desviación Típica	1.170	1.647	2.400	2.771

Nota: AS = Algoritmos simples, AB = Algoritmos de bucle, AC = Algoritmos condicionales, ACA = Algoritmos condicionales avanzados. * $p < .001$

Con relación a las puntuaciones, las medias más elevadas se presentan en los algoritmos simples y de bucle, siendo las preguntas relativas a aplicar conocimientos lógico-matemáticos sobre condicionales y condicionales avanzadas las que obtienen puntuaciones más bajas. En lo que respecta a las correlaciones, los resultados reflejaron correlaciones estadísticamente significativas y positivas entre las variables objeto de estudio en todos los casos.

Tabla 2

Estadísticos descriptivos con relación a EP, NP, ND y el resultado en el TPC

	SI	NO			
EP	21.3%	78.7%			
	Muy Bajo	Bajo	Medio	Alto	Muy Alto
NCP	38.4%	51.2%	10.4%	-	-
	MI	MA	M	DT	
ND	2.8	9.3	7.05	1.332	
TPC	3.6	10	7.36	1.471	

Nota: EP= Experiencia previa en lenguajes de programación. NP= Nivel de conocimientos sobre programación que los estudiantes consideran poseer. ND= Nivel de desempeño de los estudiantes en la asignatura. TPC= resultado global obtenidos por los estudiantes en el Test de Pensamiento Computacional. MI= Mínimo. MA= Máximo. M= Media. DT= Desviación estándar.

Por otro lado, tal y como se recoge en la tabla 2, únicamente el 21.3% de la muestra de estudio posee experiencia previa en lenguajes de programación, considerando más de la mitad que poseen un nivel bajo en este tipo de conocimientos. No obstante, la media tanto el nivel de desempeño académico en la asignatura como el nivel de habilidades de pensamiento computacional derivado del TPC se sitúan en torno al 7.

Con el fin de determinar las diferencias en las dimensiones que componen el pensamiento computacional en base al género y a la experiencia previa de los estudiantes en lenguajes de programación se realizaron pruebas t para muestras independientes. Los resultados reflejaron que las diferencias eran estadísticamente significativas entre las puntuaciones obtenidas en la dimensión algoritmos simples y el género ($t_{162}=-1.87$, $p < .001$, $\eta_p^2 = 1.16$) con un tamaño de efecto grande. Los resultados indicaron que eran los hombres (MD=9.73; DT=.624) quienes obtuvieron mejores puntuaciones que las mujeres (MD=9.20; DT=1.21).

De igual modo, se obtienen resultados estadísticamente significativos entre las puntuaciones obtenidas en la dimensión algoritmos simples ($t_{162}=2.30$, $p < .001$, $\eta_p^2 = 1.15$) en función de la experiencia previa de los estudiantes en lenguajes de programación, con un tamaño de efecto grande. Los resultados indicaron que los estudiantes que poseían

experiencia previa (MD=9.66; DT=.97) obtuvieron mejores puntuaciones que aquellos que no la tenían (MD=9.16; DT=1.19).

También se obtuvieron resultados significativos en la dimensión algoritmos de bucle ($t_{162}=2.34$, $p= .04$, $\eta_p^2 = 1.62$) en función de la experiencia previa de los estudiantes en lenguajes de programación, con un tamaño de efecto grande. Los resultados indicaron que los estudiantes que poseían experiencia previa (MD=8.61; DT=1.24) obtuvieron mejores puntuaciones que aquellos que no la tenían (MD=7.89; DT=1.71).

En lo que respecta a la obtención de los perfiles de los estudiantes a partir de un clúster, se han obtenido tres agrupaciones con las cuatro variables con un resultado “bueno”, lo que indica, de acuerdo con las aportaciones Kaufman y Rousseeuw (2009), que los datos reflejan una evidencia razonable y sólida de que existe una estructura de clúster. La variable que tiene más peso es el nivel que los estudiantes consideran poseer en programación (con una importancia del predictor de 1 sobre 1), seguida por la experiencia previa en lenguajes de programación (con una importancia de .59 sobre 1), el género (con una importancia de .29 sobre 1) y en último lugar el nivel de desempeño en la asignatura (con una importancia de .01 sobre 1). Las características de los estos tres grupos son:

Clúster 1: formado por el 31,7% de la muestra. El 48,1% declaran poseer un nivel bajo en conocimientos sobre lenguajes de programación, el 67,3% declara poseer experiencia previa en lenguajes de programación, el 63,5% son mujeres y su nivel medio de desempeño en la asignatura es de 7,06. A este clúster lo denominaremos “iniciados en programación robótica” (IPR), formado por aquellos estudiantes que han tenido experiencias y han utilizado la programación en algún momento.

Clúster 2: formado por el 32,3% de la muestra. Todos ellos (100%) declaran poseer un nivel muy bajo en conocimientos sobre lenguajes de programación, no tienen experiencia previa en lenguajes de programación, todas son mujeres y su nivel medio de desempeño en la asignatura es de 7,13. A este clúster lo denominaremos “noveles en programación robótica” (NPR), constituido por aquellos estudiantes que han tenido muy poco contacto con la programación.

Clúster 3: grupo formado por el 36% de la muestra con las siguientes características: todos (100%) poseen un bajo nivel de conocimientos sobre lenguajes de programación, no tienen experiencia previa en lenguajes de programación, todas son mujeres y su nivel medio de desempeño en la asignatura es de 6,98. A este clúster lo denominaremos “exploradores en programación robótica” (EPR), formado por aquellos estudiantes que han empezado a indagar sobre los lenguajes de programación.

Para analizar las diferencias entre los perfiles obtenidos con los conocimientos de los estudiantes en las dimensiones que componente el pensamiento computacional se llevó a cabo un análisis ANOVA de un factor mediante el procedimiento de Welch. En él, se encontraron diferencias estadísticamente significativas en “Algoritmos simples” en función del tipo de clúster ($F_{w 3, 726} = 3.92$, $p < .022$) con un tamaño de efecto medio ($\eta_p^2 = 0.46$).

También en “Algoritmos de bucle” ($F_{w3, 952} = 4.34, p < .015$), con un tamaño de efecto medio ($\eta_p^2 = 0.51$).

De igual modo, se analiza el nivel global de pensamiento computacional que poseen los estudiantes universitarios. Obtenido diferencias significativas en función del tipo de clúster ($F_{w3, 699} = 3.65, p < .028$), con un tamaño de efecto medio ($\eta_p^2 = 0.43$).

Posteriormente, para analizar las comparaciones múltiples post hoc se utiliza el procedimiento de Bonferroni, cuyos resultados se muestran en la tabla 3:

Tabla 3

Comparaciones múltiples post hoc de “Algoritmos simples” en función del tipo de perfil

	M	DT	CM	DM	P	IC 95%
IPR	9,55	.940	IPR-NPR	.229	.88	[-.30, .75]
NPR	9,32	1.12	IPR-EPR	.620	.01*	[.07, 1.16]
EPR	8,93	1.35	NPR-EPR	.391	.22	[-.13, .91]

Nota: IPR = Iniciados en programación robótica, NPR = Noveles en programación robótica, EPR = Exploradores en programación robótica.

* $p < .001$

El análisis post hoc en lo que respecta a los resultados obtenidos por los estudiantes universitarios en aquellas preguntas que median conocimientos relacionados con algoritmos de bucle se recogen en la tabla 4:

Tabla 4

Comparaciones múltiples post hoc de “Algoritmos de bucle” en función del tipo de perfil

	M	DT	CM	DM	P	IC 95%
IPR	8.46	1.39	IPR-NPR	.325	.87	[-.41, 1.06]
NPR	8.13	1.51	IPR-EPR	.914	.01*	[.15, 1.67]
EPR	7.54	1.89	NPR-EPR	.588	.16	[-.15, 1.32]

Nota: IPR = Iniciados en programación robótica, NPR = Noveles en programación robótica, EPR = Exploradores en programación robótica.

* $p < .001$

De igual manera, el análisis post hoc en lo que respecta a los resultados obtenidos por los estudiantes universitarios en el test de pensamiento computacional en función de los perfiles definidos en el análisis clúster se recogen en la tabla 5:

Tabla 5

Comparaciones múltiples post hoc de “Nivel de pensamiento computacional” en función del tipo de perfil

	M	DT	CM	DM	P	IC 95%
IPR	7.80	1.41	IPR-NPR	.538	.15	[-.12, 1.20]
NPR	7.26	1.45	IPR-EPR	.738	.03*	[.05, 1.42]
EPR	7.06	1.47	NPR-EPR	.199	1.0	[-.46, .86]

Nota: IPR = Iniciados en programación robótica, NPR = Noveles en programación robótica, EPR = Exploradores en programación robótica.

* $p < .001$

En todos los casos, el clúster “Iniciados en programación robótica” (IPR), formado por aquellos estudiantes que han tenido experiencias y han utilizado la programación en algún momento obtienen mejores puntuaciones que el perfil de estudiantes denominado “exploradores en programación robótica” (EPR), formado por aquellos discentes que han empezado a indagar sobre los lenguajes de programación, y que declaran poseer cierto nivel de conocimientos en lenguajes de programación. Es destacable señalar que aquellos que no poseen experiencia previa en lenguajes de programación obtienen peores resultados en el TPC. Este resultado viene a confirmar la necesidad de abordar este tipo de conocimientos en la formación de los futuros maestros y maestras.

4. Discusión y conclusiones

El pensamiento computacional se caracteriza por una determina forma de pensar que propicia el análisis y la relación de ideas, para la organización y representación lógica de la información. En los últimos años, se ha detectado una necesidad creciente de desarrollarlo en el ámbito educativo para dar respuesta a las necesidades de una sociedad cada vez más tecnológica. Así, surge una nueva alfabetización que intentar favorecer el aprendizaje de la programación de forma progresiva utilizando principios básicos de codificación informática mediante la utilización de algoritmos. Por ello, el objetivo general del presente estudio pretende analizar el nivel de pensamiento computacional de los universitarios para determinar el estado inicial de los estudiantes de nuevo ingreso a los grados Maestro/a en Educación Infantil y Primaria. Concretamente se plantearon dos objetivos específicos: 1) estudiar la existencia de diferencias en el nivel de pensamiento computacional en función del género y la experiencia previa de los estudiantes universitarios con diferentes lenguajes de programación robótica; 2) identificar la existencia de perfiles que permitan analizar las diferencias entre dichos perfiles con las dimensiones del pensamiento computacional.

En relación con el primer objetivo, se pudo constatar la existencia de diferencias en función del género siendo los hombres quienes obtuvieron mejores puntuaciones en la dimensión algoritmos simples. Estos resultados están en la línea con los obtenidos por otros estudios (Fennema et al., 2016; Angeli y Valanides, 2020; entre otros) quienes detectaron

que los hombres reflejaron mayor comprensión conceptual, utilizando estrategias y algoritmos más avanzados que las mujeres en los procesos de programación robótica. E incluso algunos estudios han determinado que, aun teniendo un alto nivel de pensamiento, las mujeres carecen de interés por la programación (Zhong et al., 2016), señalando que quizás carezcan de la motivación necesaria. Es, por tanto, imprescindible atender a estas diferencias a la hora de diseñar los procesos de enseñanza-aprendizaje, ya que conociendo los intereses de los estudiantes se pueden desarrollar actividades que aumenten su motivación y repercutan en la mejora de los resultados académicos. En este sentido, tal y como apuntan Kanny et al. (2014), al promover la igualdad de género, desafiar los estereotipos y brindar oportunidades equitativas, podemos fomentar la participación de las mujeres, aprovechar el talento y la creatividad, sin importar su género.

De igual modo, se hallaron diferencias en función de la experiencia previa que poseían los estudiantes en lenguajes de programación robótica, siendo aquellos que la tenían los que obtuvieron mejores resultados en las dimensiones algoritmos simples y de bucle. Sun et al. (2022) determinaron que aquellos alumnos que aprendieron a programar en edades tempranas poseían un nivel más alto en pensamiento computacional. Por ello, es muy necesario adoptar estrategias de enseñanza adecuadas en las diferentes etapas de aprendizaje. Asimismo, el pensamiento computacional y la programación están estrechamente vinculados, y su enseñanza conjunta puede proporcionar a los estudiantes herramientas poderosas para comprender y abordar los desafíos de la era digital.

En la misma línea, el segundo objetivo, trato de identificar la existencia de perfiles y de analizar las diferencias entre ellos en función de las dimensiones del pensamiento computacional. Los resultados del estudio reflejan la existencia de tres perfiles de estudiantes. Un clúster denominado IRR donde el 48,1% declaran poseer un nivel bajo en conocimientos sobre lenguajes de programación, el 67,3% tenía experiencia previa en lenguajes de programación, el 63,5% son mujeres y su nivel medio de desempeño en la asignatura es de 7,06. Este clúster posee los niveles más altos de pensamiento computacional. El segundo clúster denominado NPR se caracteriza porque todos los estudiantes declaran poseer un nivel muy bajo en conocimientos sobre lenguajes de programación, no tienen experiencia previa en lenguajes de programación, son mujeres y su nivel medio de desempeño en la asignatura es de 7,13. Este clúster posee un nivel medio de pensamiento computacional. Y, por último, el denominado EPR, caracterizado por aquellos estudiantes que poseen un bajo nivel de conocimientos sobre lenguajes de programación, no tienen experiencia previa en lenguajes de programación, son mujeres y su nivel medio de desempeño en la asignatura es de 6,98. Este grupo presenta el nivel más bajo de pensamiento computacional.

Estos resultados ponen de relieve la complejidad de la interacción entre las variables analizadas. Sin embargo, encontramos un grupo de estudiantes, formado en su mayor parte por mujeres con experiencia previa en lenguajes de programación que obtienen niveles altos en pensamiento computacional, sobre todo, en las dimensiones algoritmos simples y de bucle. Este resultado respalda la idea propuesta por Kelleher y Pausch (2005) de que, en la planificación de las acciones formativas, el diseño instruccional basado en el andamiaje es el centro de la enseñanza de la programación. En esta línea, el diseño de acciones formativas que tengan en cuenta los perfiles de partida de los estudiantes, puede ser la clave para desarrollar el pensamiento computacional, cultivar el talento entre el profesorado en formación y potenciar las competencias en este ámbito. De este modo, se

propone adaptar el programa TangibleK (Bers, 2010) que utiliza como instrumento la programación robótica a través de algoritmos, para estimular el pensamiento computacional entre los estudiantes universitarios.

Así, el programa se basa en cuatro principios básicos (Bers, 2010): 1) hacer patente el proceso de diseño y el pensamiento computacional necesario para programar el robot, es decir: plantear un problema, investigar, planificar, desarrollar un prototipo, ponerlo a prueba, rediseñarlo y analizar las soluciones; 2) potenciar la colaboración mediante la creación de equipos que compartan recursos y materiales; 3) generar grupos de conversación tecnológica, para poner en común los proyectos de programación robótica desarrollados por los diferentes grupos de estudiantes, lo que brinda la oportunidad de corregir errores como comunidad; 4) transferir los proyectos más allá del aula para hacer visible el aprendizaje, poniéndolo a prueba a través de demostraciones, exposiciones, etc. o aplicándolo en las aulas de educación infantil y/o primaria como futuros maestros y maestras. De este modo, a través de una secuenciación sencilla, centrada en identificar los conocimientos de partida, introducir el proyecto a través del juego, proponer una tarea de programación, poner en común mediante los grupos de conversación tecnológica y evaluar la tarea, se proporciona a los futuros docentes las herramientas y habilidades necesarias para fomentar el pensamiento computacional en el aula.

Sin duda, los estudios sobre pensamiento computacional se están desarrollando y merecen más investigación, sobre todo, entre el profesorado en formación pues son la base para potenciar el pensamiento computacional en las aulas de educación infantil y primaria. Aunque esta investigación ha proporcionado algunos resultados importantes sobre los niveles que poseen los estudiantes universitarios en relación con el pensamiento computacional y las variables que pueden condicionar dichos niveles, hay que tener en cuenta algunas limitaciones. El primer lugar, el tamaño de la muestra y haber optado por un diseño transversal. En futuras investigaciones se puede realizar un seguimiento dinámico de la evolución del pensamiento computacional de los estudiantes utilizando datos longitudinales y ampliando la muestra de estudio para hacer inferencias causales más sólidas. En segundo lugar, la naturaleza de la muestra, perteneciendo ésta a dos contextos muy específicos. Para verificar mejores resultados se pretende dirigir el estudio hacia estudiantes de otras titulaciones afines a la educación. Por último, este estudio ha utilizado principalmente datos cuantitativos para el análisis, y la investigación futura puede añadir datos cualitativos como entrevistas para complementar los resultados de la investigación.

Analysis of the level of computational thinking of future teachers: a diagnostic proposal for the design of training actions

1. Introduction

Computational thinking involves a process of problem solving by applying basic knowledge of science, technology, logic and mathematics (Wing, 2006). The term emerged from Papert's (1980) work on constructionist programming environments based on design and algorithm-based problem solving. It is a form of emergent literacy that seeks to foster the learning of programming progressively using basic computer coding principles. In this approach, students at different educational levels create sequences of instructions and algorithms to control the actions of an object in a virtual or physical space. For Want et al. (2022) the development of computational thinking involves various skills and abilities, such as organising and logically analysing information, identifying, investigating and implementing possible solutions, etc. while strengthening algorithmic thinking.

Algorithmic thinking is a fundamental problem-solving skill, which involves devising a step-by-step solution to address a given problem. Bers et al. (2014) state that, unlike coding, which refers to the technical skills needed to write code in a specific programming language, algorithmic thinking focuses on the conceptual process of designing an algorithm. Along these lines, for Selby (2012), algorithmic thinking is distinguished from coding and is considered an independent skill prior to coding. It involves the ability to plan and organise the actions needed to solve a problem sequentially and effectively. These skills are considered important elements of computational thinking and are fundamental in the development of programming and computational problem-solving skills (Lu and Fletcher, 2009).

There is a broad consensus about the close relationship between computational thinking skills and programming. Different authors (Tikva & Tambouris, 2021; Berland & Wilensky, 2015) argue that computational thinking has provided a new logical framework for teaching programming. Computational thinking involves a set of cognitive and problem-solving skills that are fundamental to the field of robotic programming. These skills include breaking down complex problems into simpler steps, identifying patterns and regularities, designing algorithms to solve problems, and the ability to think in a logical and structured way. Programming, in turn, provides a practical platform for students to exercise and apply these computational thinking skills in solving specific problems.

Existing studies have shown that learning and practising robotic programming contributes significantly to the development of computational thinking in students (Angeli & Valanides, 2020; Israel et al., 2015). By participating in programming activities, learners are challenged to analyse and understand how computer systems work, to identify and resolve bugs, and to improve their solutions. These experiences foster skills such as logical reasoning, abstraction, creativity and collaboration, which are fundamental to both computational thinking and programming (Popat & Starkey, 2019).

The development and improvement of computational thinking associated with robotic programming has undergone a growing evolution in recent years. However, it is true that this progress has not had the same impact on women as on men, and that there is a certain disparity in interest in robotic programming between the two groups. It is important to recognise that this situation may be influenced by cultural factors and socialisation processes experienced from childhood that may motivate men more than women to get involved in these fields (Angeli & Valanides, 2020).

Along these lines, gender stereotypes may have been constructed by associating science, technology, engineering and mathematics (STEM) with masculine characteristics. For Chan (2022) these stereotypes may create barriers and disincentives for women to explore and actively participate in these sectors. Furthermore, as Charlesworth (2019) points out, the gender roles and expectations that are transmitted in society can also influence people's choices and preferences from an early age. It is therefore essential to promote gender equality in science and technology fields, as well as in the development of robotics programming. This implies fostering inclusive and equitable education that provides opportunities and resources for all people to expand their interests and skills in these fields, regardless of their gender (Krommer, 2006).

Strong computational thinking skills can be a motivating factor for students to be interested in STEM-related careers. However, it is not currently integrated into curricula in a broad and systematic way like other subjects such as mathematics or science (Sneider et al., 2014). The lack of a specific curriculum that lays the foundations for computational thinking from an early age represents a challenge in education. There are few models or approaches that highlight computational thinking in current educational practices (Repenning et al., 2015). Despite the importance of computational thinking, little attention has been paid to its development in the training of future teachers. Most research and experiences prioritise its teaching at the compulsory secondary education stage (Maya et al., 2015), leaving a gap in the training of future teachers.

To address this situation, it is necessary to promote greater integration of computational thinking in the training of future teachers. This implies the incorporation of content and methodologies related to computational thinking in teacher training programmes, as well as the development of appropriate resources and materials for its teaching. It is also important to take into account the level of computational thinking that students possess. Carrying out diagnostic assessments to find out their level of competence in this area can help to identify areas for improvement and adapt training actions according to the needs of each group of students. In addition, it is important to consider variables such as gender and previous experience in robotics programming, as they can influence the development of computational thinking (Sun, et al. 2022). This will allow for more tailored and inclusive training actions.

Along these lines, several studies have addressed the relationship between students' computational thinking skills and gender, as well as with previous experience in robotics programming. Thus, the work carried out by Atmatzidou et al. (2016) showed that there are no significant gender differences in the development of computational thinking skills among students aged 15-18 years who participated in an Educational Robotics programme. The same holds for the study conducted by Bati (2022) with early childhood education students. These findings suggest that gender does not significantly influence the development of these skills in these contexts. Master et al. (2017) found that elementary school students with

programming experience reported higher technical interest and self-efficacy compared to those without robotic programming experience.

In short, most of the current studies that have focused on analysing gender and programming experience in relation to computational thinking have been conducted on students at non-university levels. However, the way in which these factors may influence future teachers has not been explored in depth. Therefore, the present work was undertaken in order to characterise the students in the sample in terms of the level of computational thinking based on four dimensions of study (simple, looping, conditional and advanced conditional algorithms). The second purpose was to analyse whether there are statistically significant differences between the level of computational thinking according to gender and the previous experience in robotic programming languages that university students have. And the third purpose was to identify the existence of profiles that allow us to analyse the differences between them with the dimensions of computational thinking in order to obtain a more solid understanding of how these factors may influence students' computational thinking.

2. Method

2.1. Participants

A total of 164 university students from the Master's Degrees in Early Childhood and Primary Education participated in the study, 145 of them female (88.4%) and 19 male. The age of the participants was mostly (72.3%) between 18 and 21, followed by those aged between 22 and 25 (23.1%), and those between 26 and 29 years, with 4.6%. The choice of the sample followed a non-probabilistic paradigm, following criteria of convenience and accessibility (Etikan et al., 2016).

2.2. Analysis instrument

The Computational Thinking Test (CTT), validated by Román-González (2015), was used in terms of its content, reliability and criterion and concurrent validity. The purpose of this instrument is to measure the computational thinking of the subjects. It consists of 28 questions with four answer options, only one of which is correct. The questions present different scenarios in which logical-mathematical thinking skills must be applied in order to solve them correctly. The CTT has been translated into several languages and used by the international scientific community in multiple investigations (Chan et al., 2021; Guggemos et al., 2022).

Although the instrument had been validated, an analysis of its psychometric properties was carried out. Thus, the corrected item-total correlation (r_{i-t}) was positive for all items, indicating that they all contributed to measuring the general construct measured by the instrument and in the same direction. Cronbach's Alpha was adequate (.705) with a mean item score of .76. The Kaiser-Meyer-Olkin measure of sampling adequacy was .599 and Bartlett's test of sphericity was significant ($p < .001$), with a chi-square value of .599, with 231 degrees of freedom.

To achieve the stated objectives, three additional data were collected. Firstly, we determined the level of prior programming knowledge that students considered they had, using a Likert-type scale from 1 to 5 (where 1 equals Very Low and 5 equals Very High). Secondly, students were asked whether they had previous experience in robotics programming through a dichotomous question (Yes or No). Finally, the performance level of the university students was used as a reference measure. This level of performance was based on the grade obtained by the students in the subject of Information and Communication Technologies (ICT) applied to Education of the Master's Degrees in Early Childhood and Primary Education, since several authors consider the grades obtained as a valid indicator to measure academic performance (Álvaro Paje, 1990; Ben et al., 2022), in this case, in the field of technologies.

2.3. Procedure

The study was conducted in accordance with the World Medical Association Code of Ethics (Declaration of Helsinki), which reflects the ethical principles for research involving human subjects (Morris, 2013). Prior to conducting the CTT, students were informed that their participation was voluntary. Participants were required to give informed consent prior to completing the questionnaire.

The students completed the CTT digitally, accessing it through the link sent to their corporate emails at the University before carrying out the training actions on robotics programming within the framework of the aforementioned subject. Therefore, an exploratory study was carried out in order to determine the initial state of the incoming students to the Master's Degrees in Early Childhood and Primary Education in order to generate training actions that support their academic performance in the field of STEM competences.

Once the answers had been collected, those that were answered incorrectly were assigned a value of 0, and those that were correct were assigned a value of 1. Subsequently, the items were grouped according to the type of skills required for their correct resolution, giving rise to four dimensions, as follows: 1) Simple algorithms (SA), made up of seven linear programming scenarios; 2) Loop algorithms (LA), made up of seven practical scenarios where the programming sequence is repeated several times; 3) Conditional algorithms (CA), made up of seven scenarios based on stipulating certain basic conditions for the programming blocks to be executed; and 4) Advanced conditional algorithms (ACA), made up of seven scenarios and, like the previous one, focused on establishing advanced conditions for solving the problems posed. Finally, the scores in each of the four dimensions were weighted, obtaining scores from 0 to 10 in each of them.

2.4. Data analysis

Firstly, to analyse the existence of differences in the four dimensions that make up computational thinking according to gender and students' previous experience in programming languages, t-tests for independent samples were carried out.

A two-stage cluster analysis was used to obtain the student profiles. This analysis provided an automatic procedure of the optimal number of clusters, allowing clusters with categorical and continuous variables (Rubio & Vilá, 2017). The variables included in the

cluster were “gender”, “previous experience in robotic programming languages”, “level of knowledge that students consider possessing about programming” and “level of performance”, measured from 0 to 10, and calculated from the final grade obtained in the subject. The model complies with the assumptions of independence between its variables, the continuous variables follow a normal distribution, and the categorical qualitative variables follow a multinomial distribution. Even if these assumptions were not met, Rubio and Vilá (2017) indicate that this procedure is robust enough to be applied. To analyse the distance between clusters, the log-likelihood measure was used. The Bayesian Schwartz criterion (BIC) was used to determine the number of clusters. To minimise order effects, the cases were randomly ordered. The results yielded a satisfactory model consisting of three clusters.

Finally, once the students’ profiles were obtained, a one-factor ANOVA with post hoc tests was used to analyse each of the four dimensions of computational thinking (simple algorithms, loop algorithms, conditional algorithms and advanced conditional algorithms) in terms of these profiles.

The data obtained were analysed using SPSS 27, with a p-value < .05 as a reference for significance. Effect sizes were assessed using partial eta squared, considering that the effect size is small when $\eta_p^2 = 0.01$, medium when $\eta_p^2 = 0.059$ and large when $\eta_p^2 = 0.080$ (Cohen, 1988).

3. Analysis and results

Table 1 presents the descriptive statistics of the variables, as well as the Pearson correlation matrix. The values of skewness and kurtosis lie within the limits indicating a normal distribution (values of less than 3 for skewness and less than 10 for kurtosis; Kline, 2011).

Table 1

Descriptive statistics and correlation (r) between the variables that make up computational thinking

	SA	LA	CA	ACA
SA	-	-	-	-
BA	.333*	-	-	-
CA	.329*	.374*	-	-
ACA	.205*	.461*	.413*	-
Skewness	-1.618	-0.692	-0.188	-0.269
Kurtosis	2,242	0.163	-0.800	-0.698
Mean	9.268	8.049	6.171	5.988
Standard Deviation	1.170	1.647	2.400	2.771

Note: SA = Simple Algorithms, LA = Loop Algorithms, CA = Conditional Algorithms, ACA = Advanced Conditional Algorithms.

*p < .001

Regarding the scores, the highest means are found for simple and loop algorithms, the questions on applying mathematical-logical knowledge of conditionals and advanced conditionals obtaining the lowest scores. Regarding correlations, the results showed statistically significant and positive correlations between the variables under study in all cases.

Table 2

Descriptive statistics in relation to previous experience in programming languages, the level of programming that the students believe they possess, the students' level of performance in the subject, and the result in the CTT

	YES	NO			
PE	21.3%	78.7%			
	Very Low	Low	Medium	High	Very High
PKL	38.4%	51.2%	10.4%	-	-
	MI	MA	M	SD	
PL	2.8	9.3	7.05	1.332	
CTT	3.6	10	7.36	1.471	

Note: PE= Previous experience in programming languages. PKL = Level of programming knowledge that students believe they possess. PL= Students' level of performance in the subject. CTT= Overall score obtained by the students in the Computational Thinking Test. MI= Minimum. MA= Maximum. M= Mean. SD= Standard deviation

On the other hand, as shown in table 2, only 21.3% of the study sample has previous experience in programming languages, considering that more than half of them have a low level in this type of knowledge. However, the average level of academic performance in the subject and the level of computational thinking skills derived from the CTT are of around 7.

To determine the differences in the dimensions that make up computational thinking based on gender and students' previous experience in programming languages, t-tests were performed for independent samples. The results showed that the differences were statistically significant between the scores obtained in the simple algorithms dimension and that of gender ($t_{162}=-1.87$, $p < .001$, $\eta_p^2 = 1.16$), with a large effect size. The results indicated that males (MD=9.73; SD=.624) scored higher than females (MD=9.20; SD=1.21).

Similarly, statistically significant results are obtained between the scores obtained in the simple algorithms dimension ($t_{162}=2.30$, $p < .001$, $\eta_p^2 = 1.15$) according to the students' previous experience in programming languages, with a large effect size. The results indicated that students with prior experience (MD=9.66; SD=.97) scored higher than those without (MD=9.16; SD=1.19).

Significant results were also obtained in the loop algorithms dimension ($t_{162}=2.34$, $p=.04$, $\eta_p^2 = 1.62$) according to students' prior experience in programming languages, with a large effect size. The results indicated that students with prior experience ($MD=8.61$; $SD=1.24$) scored higher than those without ($MD=7.89$; $SD=1.71$).

In terms of obtaining student profiles from a cluster, three groupings were obtained with the four variables with a "good" result, indicating, in line with Kaufman and Rousseeuw (2009), that the data reflect reasonable and strong evidence of a cluster structure. The variable that carries the most weight is the level that students believe to possess in programming (with a predictor significance of 1 out of 1), followed by previous experience in programming languages (with a significance of .59 out of 1), gender (with a significance of .29 out of 1) and lastly, level of performance in the subject (with a significance of .01 out of 1). The characteristics of the three clusters are as follows:

Cluster 1: made up of 31.7% of the sample. 48.1% declare that they have a low level of knowledge of programming languages, 67.3% declare that they have previous experience in programming languages, 63.5% are women and their average level of performance in the subject is 7.06. We will call this cluster "initiates in robotics programming" (IRP), and it is made up of those students who have had experience and have used programming at some point.

Cluster 2: made up of 32.3% of the sample. All of them (100%) declare that they have a very low level of knowledge of programming languages, they have no previous experience in programming languages, they are all women and their average level of performance in the subject is 7.13. We will call this cluster "novices in robotics programming" (NRP), made up of those students who have had very little contact with programming.

Cluster 3: a group formed by 36% of the sample with the following characteristics: all (100%) have a low level of knowledge about programming languages, have no previous experience in programming languages, all are female and their average level of performance in the subject is 6.98. We will call this cluster "explorers in robotic programming" (ERP), and it is composed of those students who have started to investigate programming languages.

To analyse the differences between the profiles obtained with the students' knowledge in the dimensions that make up computational thinking, a one-factor ANOVA analysis was carried out using Welch's procedure. There, statistically significant differences were found in "Simple algorithms" according to the cluster type ($F_{w 3, 726} = 3.92$, $p < .022$), with a mean effect size ($\eta_p^2 = 0.46$). This was also the case in "Loop algorithms" ($F_{w 3, 952} = 4.34$, $p < .015$), with a medium effect size ($\eta_p^2 = 0.51$).

Similarly, the overall level of computational thinking possessed by university students was analysed. Significant differences were obtained according to the type of cluster ($F_{w 3, 699} = 3.65$, $p < .028$), with a medium effect size ($\eta_p^2 = 0.43$).

Subsequently, the Bonferroni procedure was used to analyse the post hoc multiple comparisons, the results of which are shown in table 3:

Table 3

Post hoc multiple comparisons of “Simple Algorithms” according to the profile type

	M	SD	CM	DM	P	IC 95%
IRP	9,55	.940	IPR-NPR	.229	.88	[-.30, .75]
NRP	9,32	1.12	IPR-EPR	.620	.01*	[.07, 1.16]
ERP	8,93	1.35	NPR- EPR	.391	.22	[-.13, .91]

Note: IRP = Initiates in robotic programming, NRP = Novices in robotic programming, ERP = Explorers in robotic programming. * $p < .001$

The post hoc analysis regarding the results obtained by the university students in those questions that measure knowledge related to loop algorithms are shown in table 4:

Table 4

Post hoc multiple comparisons of “Loop Algorithms” according to the profile type

	M	SD	CM	DM	P	IC 95%
IRP	8.46	1.39	IPR-NPR	.325	.87	[-.41, 1.06]
NRP	8.13	1.51	IPR-EPR	.914	.01*	[.15, 1.67]
ERP	7.54	1.89	NPR- EPR	.588	.16	[-.15, 1.32]

Note: IRP = Initiates in robotic programming, NRP = Novices in robotic programming, ERP = Explorers in robotic programming. * $p < .001$

Similarly, the post hoc analysis regarding the results obtained by the university students in the computational thinking test according to the profiles defined in the cluster analysis are shown in table 5:

Table 5

Post hoc multiple comparisons of “Level of computational thinking” according to the profile type

	M	SD	CM	DM	P	IC 95%
IRP	7.80	1.41	IPR-NPR	.538	.15	[-.12, 1.20]
NRP	7.26	1.45	IPR-EPR	.738	.03*	[.05, 1.42]
ERP	7.06	1.47	NPR- EPR	.199	1.0	[-.46, .86]

Note: IRP = Initiates in robotic programming, NRP = Novices in robotic programming, ERP = Explorers in robotic programming. * $p < .001$

In all cases, the cluster “Initiates in robotic programming” (IRP), formed by those students who have had experience and have used programming at some point, obtain better scores than the profile of students called “Explorers in robotic programming” (ERP), formed by those students who have started to investigate programming languages, and who state that they have some level of knowledge in programming languages. It is noteworthy that those who have no previous experience in programming languages perform less well in the CTT. This result confirms the need to address this type of knowledge in the training of future teachers to strengthen STEM competences.

4. Discussion and conclusions

Computational thinking is characterised by a specific way of thinking that favours the analysis and relation of ideas, for the organisation and logical representation of information. In recent years, there has been a growing need to develop it in education in order to respond to the needs of an increasingly technological society. We are thus dealing with the emergence of a new literacy that seeks to encourage the learning of programming in a progressive way by using basic principles of computer coding through the use of algorithms. Therefore, the general objective of this study is to analyse the level of computational thinking of university students in order to determine the initial state of incoming students to the master’s degrees of Teacher of Early Childhood and Primary Education. Specifically, two specific objectives were set out: 1) to study the existence of differences in the level of computational thinking according to gender and previous experience of university students with different robotic programming languages; 2) to identify the existence of profiles that allow us to analyse the differences between these profiles with the dimensions of computational thinking.

In relation to the first objective, it was possible to confirm the existence of differences according to gender, with men obtaining better scores in the simple algorithms dimension. These results are in line with those obtained by other studies (Fennema et al., 2016; Angeli & Valanides, 2020; among others) that found that men showed greater conceptual understanding, using more advanced strategies and algorithms than women in robotic programming processes. And some studies have even found that, despite having a high level of thinking, women lack interest in programming (Zhong et al., 2016), indicating that they may lack the necessary motivation. It is therefore essential to pay attention to these differences when carrying out teaching-learning processes, since knowing the interests of students can develop activities that increase their motivation and have an impact on improving academic results. In this sense, as Kanny et al. (2014) point out, by promoting gender equality, challenging stereotypes and providing equal opportunities, we can encourage the participation of women, harnessing the talent and creativity of all individuals, regardless of their gender.

Similarly, differences were found according to students’ prior experience with robotic programming languages, with those who had prior experience performing better on the

simple and loop algorithms dimensions. Sun et al. (2022) found that those students who had learned to program at an early age had a higher level of computational thinking. Therefore, there is a strong need for appropriate teaching strategies at different stages of learning. Moreover, computational thinking and programming are closely linked, and teaching them together can provide students with powerful tools to understand and address the challenges of the digital age.

Along the same lines, the second objective sought to identify the existence of profiles and to analyse the differences between them in terms of the dimensions of computational thinking. The results of the study reflect the existence of three student profiles. In a cluster called IRP, where 48.1% of the students declared to have a low level of knowledge of programming languages, 67.3% had previous experience in programming languages, 63.5% were women and their average level of performance in the subject was 7.06. This cluster has the highest levels of computational thinking. The second cluster called NRP is characterised by the fact that all students report a very low level of knowledge of programming languages and have no previous experience in programming languages; they are female and their average level of performance in the subject is 7.13. This cluster has an average level of computational thinking. And, finally, the so-called ERP, characterised by those students who have a low level of knowledge of programming languages, have no previous experience in programming languages, are female and their average level of performance in the subject is 6.98. This group has the lowest level of computational thinking.

These results highlight the complexity of the interaction between the variables analysed. However, we found a group of students, mostly women with previous experience in programming languages, who obtained high levels of computational thinking, especially in the dimensions of simple algorithms and loops. This result supports the idea proposed by Kelleher & Pausch (2005) that scaffolding-based instructional design is at the heart of programming instruction in training planning. In this line, the design of training actions that take into account students' starting profiles may be the key to developing computational thinking, cultivating talent among trainee teachers and enhancing STEM skills. Thus, it is proposed to adapt the TangibleK programme (Bers, 2010), which uses robotic programming through algorithms as a tool to stimulate computational thinking among university students.

Thus, the programme is based on four basic principles (Bers, 2010): 1) making visible the design process and the computational thinking required to program the robot – in other words, posing a problem, researching, planning, developing a prototype, testing, redesigning and analysing the solutions; 2) enhancing collaboration by creating teams that share resources and materials; 3) generating technology conversation groups, to share the robotic programming projects developed by the different groups of students, which provides the opportunity to correct mistakes as a community; 4) transferring the projects beyond the classroom to make the learning visible, testing it through demonstrations, exhibitions, etc., or applying it in classrooms, by making it visible in the classroom. or by applying it in infant and/or primary classrooms as future teachers. In this way, through a simple sequencing, focusing on identifying the starting knowledge, introducing the project through games, proposing a programming task, sharing it through the technology conversation groups and evaluating the task, future teachers are provided with the tools and skills necessary to foster computational thinking in the classroom.

Undoubtedly, studies on computational thinking are developing and deserve more research, especially among trainee teachers, as they are the basis for enhancing computational thinking in early childhood and primary school classrooms. Although this research has provided some important results on the levels of computational thinking among university students and the variables that may condition these levels, some limitations must be taken into account. Firstly, the size of the sample and having opted for a cross-sectional design. Future research can dynamically track the evolution of students' computational thinking by using longitudinal data and expanding the study sample to make more robust causal inferences. Secondly, the nature of the sample, belonging to two very specific contexts. In order to verify better results, the study is to be directed towards students from other education-related degrees. Finally, this study has mainly used quantitative data for analysis, and future research may add qualitative data such as interviews to complement the research results.

References

- Angeli, C. & Valanides, N. (2020). Developing young children's computational thinking with educational robotics: An interaction effect between gender and scaffolding strategy. *Computers in human behavior*, 105, 105954. <https://doi.org/10.1016/j.chb.2019.03.018>
- Atmatzidou, S. & Demetriadis, S. (2016). Advancing students' computational thinking skills through educational robotics: A study on age and gender relevant differences. *Robotics and Autonomous Systems*, 75, 661-670. <https://doi.org/10.1016/j.robot.2015.10.008>
- Álvaro Paje, M. (1990). *Hacia un modelo causal del rendimiento académico*. CIDE.
- Bati, K. (2022). A systematic literature review regarding computational thinking and programming in early childhood education. *Educ Inf Technol*, 27, 2059–2082. <https://doi.org/10.1007/s10639-021-10700-2>
- Ben, A., Dahmani, M., & Ragni, L. (2022). ICT use, digital skills and students' academic performance: Exploring the digital divide. *Information*, 13(3). <https://doi.org/10.3390/info13030129>
- Bers, M. (2010). The TangibleK robotics program: Applied computational thinking for young children. *Early Childhood Research & Practice*, 12(2), 1-20. <http://ecrp.uiuc.edu/v12n2/bers.html>
- Bers, M.U., Flannery, L., Kazakoff, E.R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157. <https://doi.org/10.1016/j.compedu.2013.10.020>
- Cohen, J. (1988). *Statistical power analysis for the behavioral sciences* (2nd ed.). Erlbaum.
- Chan, R. (2022). A social cognitive perspective on gender disparities in self-efficacy, interest, and aspirations in science, technology, engineering, and mathematics (STEM): the influence of cultural and gender norms. *International Journal of STEM Education*, 9(1), 1-13. <https://doi.org/10.1186/s40594-022-00352-0>

- Charlesworth, T. & Banaji, M. (2019). Gender in science, technology, engineering, and mathematics: issues, causes, solutions. *Journal of Neuroscience*, 39(37), 7228-7243. <https://doi.org/10.1523/JNEUROSCI.0475-18.2019>
- Chan, S., Looi, C., & Sumintono, B. (2021). Assessing computational thinking abilities among Singapore secondary students: A rasch model measurement analysis. *Journal of Computers in Education*, 8(2), 213-236. <https://doi.org/10.1007/s40692-020-00177-2>
- Etikan, I., Musa, S. A., & Alkassim, R. S. (2016). Comparison of convenience sampling and purposive sampling. *American Journal of Theoretical and Applied Statistics*, 5(1), 1-4. <https://doi.org/10.11648/j.ajtas.20160501.11>
- Fennema, E., Carpenter, T., & Jacobs, V. (2016). A longitudinal study of gender differences in young children's mathematical thinking. *Educational Researcher*, 27 (5), 6-11. <https://doi.org/10.3102/0013189X027005006>
- Guggemos, J., Seufert, S., & Román-González, M. (2022). Computational Thinking Assessment—Towards More Vivid Interpretations. *Technology, Knowledge and Learning*, 1-30. <https://doi.org/10.1007/s10758-021-09587-2>
- Kline, R. B. (2011). Principles and practice of structural equation modeling. New York: Guilford Press.
- Kaufman, L. & Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.
- Kanny, M., Sax, L., & Riggers-Piehl, T. (2014). Investigating forty years of STEM research: how explanations for the gender gap have evolved over time. *Journal of Women and Minorities in Science and Engineering*, 20(2), 127-148. <https://doi.org/10.1615/JWomenMinorScienEng.2014007246>
- Kelleher, C. & Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. *ACM Computing Surveys*, 37(2), 83-137. <https://doi.org/10.1145/1089733.1089734>
- Lu, C., Macdonald, R., Odell, B., Kokhan, V., Demmans, C., & Cutumisu, M. (2022). A scoping review of computational thinking assessments in higher education. *Journal Computer High Education*, 34, 416-461. <https://doi.org/10.1007/s12528-021-09305-y>
- Majeed, B., Jawad, L., & ALRikabi, H. (2002). Computational Thinking (CT) Among University Students. *International Journal of Interactive Mobile Technologies*, 16(10), 244-252
- Maya, I., Pearson, J., Tapia, T., Wherfel, Q., & Reese, G. (2015). Supporting all learners in school-wide computational thinking: a cross-case qualitative analysis. *Computers & Education*, 82, 263-279. <https://doi.org/10.1016/j.compedu.2014.11.022>
- Master, A., Cheryan, S., Moscatelli, A., & Meltzoff, A. (2017). Programming experience promotes higher STEM motivation among first-grade girls. *Journal of Experimental Child Psychology*, 160 (1), 92-106. <https://doi.org/10.1016/j.jecp.2017.03.013>

- Morris, K. (2013). Revising the Declaration of Helsinki. *World Report*, 381, 1889–1890. [https://doi.org/10.1016/S0140-6736\(13\)60951-4](https://doi.org/10.1016/S0140-6736(13)60951-4)
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. New York: Basic Books.
- Popat, S. & Starkey, L. (2019). Learning to code or coding to learn? A systematic review. *Computers & Education*, 128, 365-376. <https://doi.org/10.1016/j.compedu.2018.10.005>
- Repenning, A., Webb, D., & Ioannidou, A. (2010). Scalable game design and the development of a checklist for getting computational thinking into public schools. Proceedings of the 41st ACM Technical Symposium on Computer Science Education, Milwaukee, Wisconsin, USA.
- Román-González, M. (2015). Computational Thinking Test: Design Guidelines and Content Validation. *International Conference on Education and New Learning Technologies EDULEARN*. Barcelona. <https://bit.ly/3yZBd7t>
- Rubio, M.J. & Vilà, R. (2017). El análisis de conglomerados bietápico o en dos fases con SPSS. REIRE. *Revista d'Innovació i Recerca en Educació*, 10(1), 118–126. <https://doi.org/10.1344/reire2017.10.11017>
- Selby, C. (2012). Promoting computational thinking with programming. *Proceedings of the 7th workshop in primary and secondary computing education*, ACM. New York. 74-77. <https://doi.org/10.1145/2481449.2481466>
- Sun, L., Hu, L. & Zhou, D. (2022). Programming attitudes predict computational thinking: Analysis of differences in gender and programming experience. *Computers & Education*, 181, 104457. <https://doi.org/10.1016/j.compedu.2022.104457>
- Tsarava, K., Moeller, K., Román-González, M., Golle, J., Leifheit, L., Butz, M. V., & Ninaus, M. (2022). A cognitive definition of computational thinking in primary education. *Computers & Education*, 179, 104425. <https://doi.org/10.1016/j.compedu.2021.104425>
- Tikva, C. & Tambouris, E. (2021). Mapping computational thinking through programming in K-12 education: A conceptual model based on a systematic literature Review. *Computers & Education*, 162, 104083. <https://doi.org/10.1016/j.compedu.2020.104083>
- Ung, L., Labadin, J., & Mohamad, F. (2022). Computational thinking for teachers: Development of a localised E-learning system. *Computers & Education*, 177, 104379. <https://doi.org/10.1016/j.compedu.2021.104379>
- Wang, C., Shen, J., & Chao, J. (2022). Integrating computational thinking in STEM education: A literature review. *International Journal of Science and Mathematics Education*, 20(8), 1949-1972. <https://doi.org/10.1007/s10763-021-10227-5>
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>
- Zhong, B., Wang, Q., Chen, J., & Li, Y. (2016). An exploration of three-dimensional integrated assessment for computational thinking. *Journal of Educational Computing Research*, 53 (4), 562-590. <https://doi.org/10.1177/0735633115608444>

Cómo citar:

Villalustre-Martínez, L. (2024). Análisis del nivel de pensamiento computacional de los futuros maestros: una propuesta diagnóstica para el diseño de acciones formativas [Analysis of the level of computational thinking of future teachers: a diagnostic proposal for the design of training actions]. *Pixel-Bit. Revista de Medios y Educación*, 69, 169-194. <https://doi.org/10.12795/pixelbit.101205>