

Recibido 15 MAR. 2024

ReCIBE, Año 12 No.2, NOV. 2024

Aceptado 18 MAR. 2024

ENSEÑANZA DE PROGRAMACIÓN PARA NIÑOS Y JÓVENES CON DISCAPACIDAD VISUAL: MAPEO SISTEMÁTICO DE LA LITERATURA

TEACHING PROGRAMMING FOR CHILDREN AND YOUNG PEOPLE WITH VISUAL DISABILITY: SYSTEMATIC MAPPING OF LITERATURE

Raúl A. Peredo Estudillo¹
raulperestu@icloud.com
Juan C. Pérez Arriaga¹
Gerardo Contreras Vega¹
Ángel J. Sánchez García¹

¹Facultad de Estadística e Informática, Universidad Veracruzana, Xalapa, México

Resumen:

En la actualidad la educación de los estudiantes se enfoca en las disciplinas Ciencia, Tecnología, Ingeniería y Matemáticas (modelo integral STEM por sus siglas en inglés). Una estrategia efectiva para incorporar la ingeniería y la tecnología en la educación es promover el pensamiento computacional, donde la programación desempeña un papel fundamental en el desarrollo de habilidades lógicas y del pensamiento lógico estructurado. Sin embargo, es esencial destacar que las personas con discapacidad visual enfrentan desafíos específicos para adquirir estas competencias. El propósito de esta investigación es llevar a cabo un mapeo sistemático de la literatura con el fin de analizar los métodos, técnicas y herramientas utilizados para enseñar programación a niños y jóvenes con discapacidad visual. En este artículo, se exponen los resultados del mapeo sistemático de la literatura en el que se seleccionaron un total de 23 estudios. Los principales hallazgos incluyeron cinco métodos: aprendizaje háptico, resolución de casos, trabajo en equipo, enseñanza activa y enfoque basado en problemas. En cuanto a las técnicas, se identificaron cinco: basadas en tecnología, interactivas, basadas en juegos, en parejas y colaborativas. Para concluir, se encontraron cuatro tipos de herramientas: asistidas, hápticas basadas en audio y lectores de pantalla.

Palabras Clave: Discapacidad visual, Personas ciegas, Programación, Enseñanza, Accesibilidad, Mapeo sistemático.

Abstract:

Today's student education focuses on the disciplines of Science, Technology, Engineering and Mathematics (STEM) (STEM integrated model). Science, Technology, Engineering and Mathematics (STEM). model). An effective strategy for incorporating engineering and engineering and technology in education is to promote computational thinking, where programming programming plays a fundamental role in the development of logic and thinking skills. development of logical skills and structured logical thinking. Without However, it is essential to note that the visually impaired face specific challenges in acquiring these skills. specific challenges in acquiring these skills. The purpose of this The purpose of this research is to carry out a systematic mapping of the literature in order to analyze the methods, techniques and tools used to teach to analyze the methods, techniques and tools used to teach programming to visually impaired children and youth. programming to children and youth with visual impairment. In this article, the results of the the results of the systematic literature mapping in which a total of 23 studies were selected. a total of 23 studies were selected. The main findings included five methods: haptic learning, case solving, teamwork, active teaching and problem-based approach, active teaching and problem-based approach. In terms of techniques, five were five were identified: technology-based, interactive, game-based, paired and collaborative. and collaborative. In conclusion, four types of tools were found tools: assisted, audio-based haptic and screen readers.

Keywords: Visual impairment, Blind people, Programming, Teaching, Accessibility, Systematic mapping.

1. Introducción

La discapacidad visual se refiere a la pérdida parcial o total de la capacidad visual, evaluada en términos de la capacidad de lectura a diferentes distancias, el campo y la agudeza visual (World Health Organization: WHO, 2023). La Clasificación Internacional de Enfermedades 11 divide el deterioro de la visión en dos grupos: a distancia y cercano (Clasificación estadística internacional de enfermedades y problemas de salud relacionados (11ª ed.), 2019).

En el grupo de deterioro a distancia, se establecen niveles según la gravedad, desde suave hasta ceguera, basados en la agudeza visual. Para el deterioro de visión cercana, se utiliza la medida N6 o M.08 a 40 cm como criterio (Clasificación estadística internacional de enfermedades y problemas de salud relacionados (11ª ed.), 2019). En la actualidad, el campo de STEM (Ciencia, Tecnología, Ingeniería y Matemáticas) representa el 20% de los puestos de trabajo, con una cifra de 26 millones de empleos solo en los Estados Unidos (Delgado, 2022). El pensamiento computacional (PC) se define como el proceso mediante el cual se plantean problemas de tal manera que sus soluciones pueden representarse mediante secuencias de instrucciones y algoritmos (Wing, 2006).

La programación se convierte en una valiosa herramienta para cultivar este tipo de pensamiento lógico y estructurado, ayudando a desempeñar un papel crucial en STEM al proporcionar una base sólida para abordar problemas, desarrollar soluciones eficientes y aprovechar la tecnología de manera efectiva.

La problemática por tratar se relaciona con la falta de información y recursos disponibles para enseñar programación a personas con discapacidad visual. En la mayoría de los casos, los educadores y personas interesadas en brindar esta enseñanza se enfrentan a un desafío significativo debido a la ausencia de métodos, herramientas y técnicas de enseñanza adecuadas para este público. La falta de acceso a información relevante dificulta la inclusión de personas con discapacidad visual en el mundo de la programación, limitando sus oportunidades de desarrollo.

Por otro lado, los principales aportes de este estudio son fundamentales para fomentar el desarrollo del pensamiento lógico estructurado desde una edad temprana, a través de la programación. Esto no solo beneficia a las personas con discapacidad visual al abrir nuevas oportunidades educativas y profesionales, sino que también enriquece el campo de la programación al promover la diversidad y el acceso a talento variado.

Este mapeo sistemático de la literatura (MSL) se centra en métodos y herramientas de enseñanza en programación, buscando ampliar el acceso y promover la inclusión educativa para todas las personas, incluyendo aquellas con limitaciones o barreras, en busca de igualdad de oportunidades. El presente MSL está estructurado como se describe a continuación: La sección 2 presenta los antecedentes y trabajos relacionados. La sección 3 presenta el método de investigación empleado para la ejecución de esta investigación. La sección 4 presenta los resultados de la investigación. La sección 5 presenta la discusión. La sección 6 las amenazas a la validez y la sección 7 conclusiones.

2. Trabajo Relacionado

Un estudio realizado por Al-Ratta & Al-Khalifa (2013), explora la literatura sobre la enseñanza de la programación a personas ciegas, calificándola mediante una revisión sistemática y abordando temas como: necesidades específicas, desafíos en cursos convencionales y la posible necesidad de lenguajes de programación adaptados. El artículo establece dos tipos de cursos para alumnos ciegos: uno que utiliza manuales en Braille y otro que presenta instrucciones programadas en forma de audiolibros en casetes. Se observa que los estudiantes ciegos tienden a depender más de la experiencia concreta previa que de la abstracción, lo que dificulta seguir la lógica de programación. Además, se destaca que las personas con discapacidad visual no son una entidad homogénea, ya que cada individuo puede tener diferentes necesidades y habilidades. Por lo tanto, se sugiere

enfocarse en el desarrollo de herramientas que se adapten a diversas categorías de usuarios, lo que abre un nuevo campo de investigación.

Ying Li (2016), propone un estudio con el objetivo de explorar y presentar un enfoque educativo que utilice el pensamiento computacional como base para enseñar programación. Destaca la importancia del pensamiento computacional en la ciencia de la computación y enfatiza que los estudiantes no deben centrarse en la memorización de conocimientos o en imitar a las computadoras para resolver problemas, sino en desarrollar su propio pensamiento y conciencia a través del aprendizaje. El estudio subraya cómo este enfoque impacta tanto en el proceso de enseñanza como en el aprendizaje de los estudiantes, destacando la relevancia del pensamiento computacional en la educación en programación.

Nuestra investigación se basa en estos enfoques y los orienta hacia el ámbito académico con el propósito de fomentar el pensamiento lógico estructurado desde una edad temprana en niños con discapacidad visual. Además, incentivar el desarrollo de habilidades que les sean beneficiosas durante su proceso de formación educativa.

3. Método de Investigación

Para llevar a cabo esta investigación se siguió el método propuesto por Kitchenham et al. (2015) el cual se describe a continuación:

3.1 Etapa de Planeación

Para esta investigación se definieron tres preguntas de investigación y se muestran en la Tabla 1.

Pregunta de investigación	Motivación
PI1. ¿Cuáles son los métodos y técnicas utilizados para enseñar programación a niños y jóvenes con discapacidad visual?	A partir de la identificación de métodos y técnicas efectivas para enseñar programación para este grupo de estudiantes, se pueden desarrollar enfoques adecuados que les permitan desarrollar habilidades en programación y fomentar su participación en el campo de las áreas STEM.
PI2. ¿Cuáles son las herramientas disponibles para la enseñanza de programación dirigidas a niños y jóvenes con discapacidad visual?	Al conocerlas herramientas disponibles y su utilidad, se podrán tomar decisiones informadas sobre qué recursos utilizar en entornos educativos y cuál puede ser el impacto de uso en su aprendizaje.
PI3. ¿Cuál es la efectividad de las técnicas, métodos y herramientas utilizadas en la enseñanza de programación para niños y jóvenes con discapacidad visual?	Permitirá identificar las mejores prácticas y enfoques que maximicen el aprendizaje y la inclusión de estos estudiantes. Asimismo, identificar parámetros o formas de medición de efectividad en esta área de investigación para ser aplicada en futuras investigaciones.

Tabla 1. Preguntas de Investigación

De las fuentes sugeridas por la guía de Kitchenham et al. (2015) se escogieron IEEEExplore, ACM Digital Library, Science Direct y Springer Link y MDPI como base de datos complementaria. Las fuentes seleccionadas se pueden visualizar en la Tabla 2.

Fuente	Sitio web
IEEEExplorer	https://ieeexplore.ieee.org/Xplore/home.jsp
ACM Digital Library	https://dl.acm.org/
Science Direct	https://www.sciencedirect.com/
Springer Link	https://link.springer.com/
MDPI	https://www.mdpi.com

Tabla 2. Fuentes de información seleccionadas

Para la generación de la cadena de búsqueda, se identificaron los términos claves para la búsqueda de artículos. Los términos definidos para esta RSL se muestran en la Tabla 3.

Término clave	Término en inglés	Término relacionado
Discapacidad visual	Visual Impairment	Blind people
Enseñanza	Teaching	Learning
Programación	Programming	Coding

Tabla 3. Términos clave de búsqueda

La cadena de búsqueda representa la entrada para recuperar estudios relevantes de las fuentes de datos mostradas en la Tabla 2. Se generaron diversas cadenas de búsqueda a partir de las palabras de la Tabla 3 enlazados con operadores booleanos (OR y AND), y se seleccionó aquella que obtuvo un mejor desempeño, en términos de *precision* y *recall*.

La métrica *precision* es la proporción de documentos recuperados que son relevantes para la consulta, véase la ecuación (1); por su parte, la métrica *recall* es la proporción de documentos relevantes que se recuperan con éxito, véase la ecuación (2).

$$Precision = \frac{\text{Documentos relevantes recuperados}}{\text{Total de documentos recuperados}} \quad (1)$$

$$Recall = \frac{\text{Documentos relevantes recuperados}}{\text{Total de documentos relevantes}} \quad (2)$$

La Tabla 4 muestra la cadena de búsqueda aplicada, con su respectiva medida de *recall* y *precision*.

Cadena de búsqueda	Valor de <i>recall</i>	Valor de <i>precision</i>
("Visual impairment" OR "Visual disability" OR "Blind Students" OR "Visually Impaired") AND ("Teaching" OR "Learning") AND ("Programming" OR "Coding")	0.1	0.02

Tabla 4. Cadenas de búsqueda aplicadas por fuente de datos

3.2 Etapa de Conducción

En este apartado se describen los criterios de selección de los artículos encontrados con las cadenas de búsqueda y se explica el proceso de selección de los estudios primarios que serán analizados.

3.2.1 Criterios de selección

Se propusieron cinco criterios de inclusión (CI) y dos criterios de exclusión CE, para seleccionar los estudios primarios de los estudios recuperados en las fuentes seleccionadas, véase Tablas 5 y 6, respectivamente.

Clave	Descripción
CI1	La publicación debe estar escrita en el idioma inglés.
CI2	La fecha de publicación es de 2015 - 2023.
CI3	La publicación debe ser artículo de investigación sobre software (Artículo de Journal, conferencia, revista).
CI4	El título o abstract de la publicación debe dar indicios de responder al menos una pregunta de investigación.
CI5	Una lectura completa de la publicación debe responder al menos una pregunta de investigación.

Tabla 5. Criterios de inclusión

Clave	Descripción
CE1	Estudios que son carteles, presentaciones, resúmenes o tutoriales.
CE2	No se tiene acceso completo a la publicación.
CE3	Estudios duplicados.

Tabla 6. Criterios de exclusión

3.2.2 Proceso de selección

Se definieron cuatro etapas para el proceso de selección de estudios. La Figura 1 muestra los criterios aplicados en cada etapa.

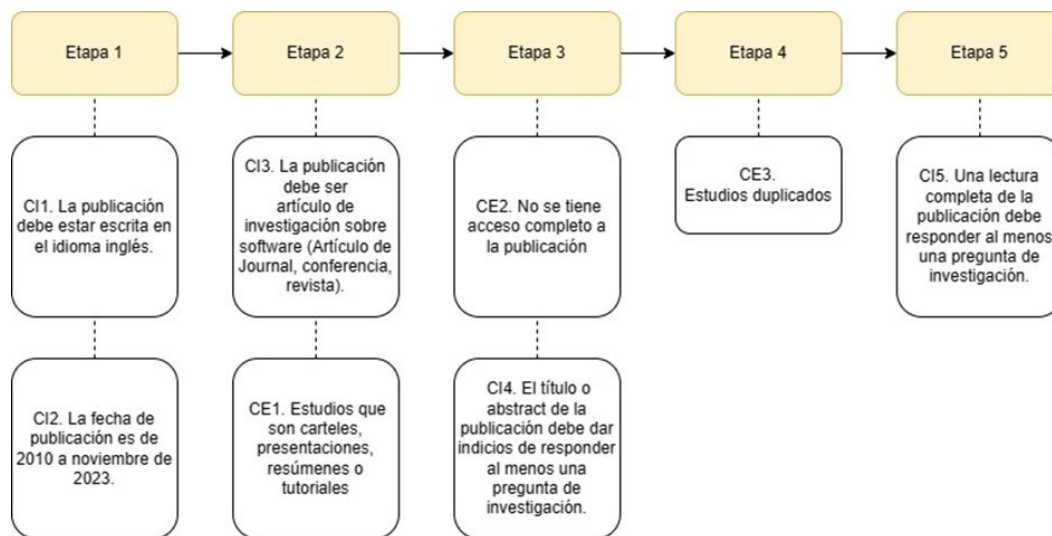


Figura 1. Proceso de selección de estudios primarios.

La Tabla 7 muestra el número de estudios primarios seleccionados por etapa, donde se puede observar que IEEEExplore seleccionó la mayor cantidad de estudios.

Fuente	Resultados	Etapa 1	Etapa 2	Etapa 3	Etapa 4	Etapa 5
IEEEExplore	361	268	263	32	32	19
ACM Digital Library	1605	1084	780	10	2	1
MDPI	175	143	132	11	3	2
Science Direct	175	3999	113	6	0	0
Springer Link	2284	1628	554	65	2	1
Total	12188	7122	1842	124	39	23

Tabla 7. Número de estudios primarios seleccionados por etapa.

3.3 Síntesis

La extracción de datos fue realizada con base en las preguntas de investigación, identificando 23 artículos que las contestan y categorizando estos hallazgos por métodos, técnicas, herramientas y efectividad. Para elaborar la síntesis narrativa se utilizó la metodología propuesta por Popay et al. (2006). Esta metodología está dirigida a las revisiones sistemáticas que se ocupan de la efectividad de alguna intervención o de los factores que incluyen la implementación de las intervenciones. El proceso de síntesis, así como las referencias de los estudios primarios analizados se encuentra en https://uvmx-my.sharepoint.com/:x/g/personal/zs19014050_estudiantes_uv_mx/Ef0MGoxl9pxGqh8hSXDyUxMB5Vff_Uplw-cGQ95sWm4JRA?rttime=Mevjm08d3Eg

4. Resultados

El año con mayor número de artículos publicados fue 2022, con 5 artículos. Durante los años 2015 hasta el 2018 se obtuvo una media de 3 artículos por año y a partir del 2019 al 2021 se detectó una disminución en la cantidad de publicaciones de menos de 2 artículos por año. La Figura 2 muestra la distribución de los estudios primarios encontrados por fuente de publicación y la Figura 3 muestra la distribución de los estudios por año de publicación.

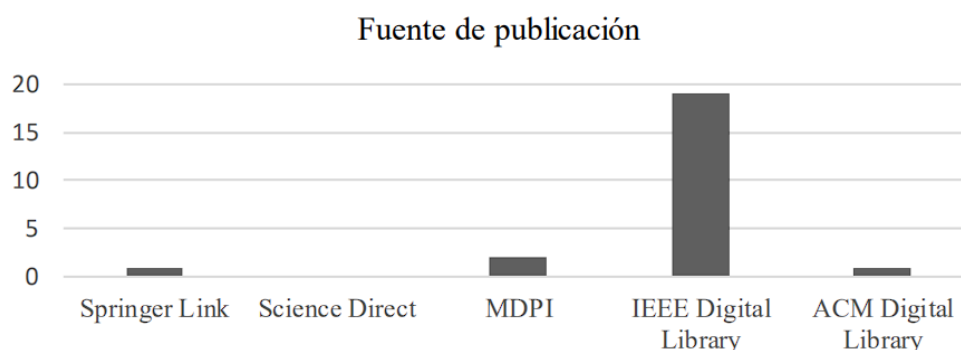


Figura 2. Distribución de estudios por fuente de publicación.

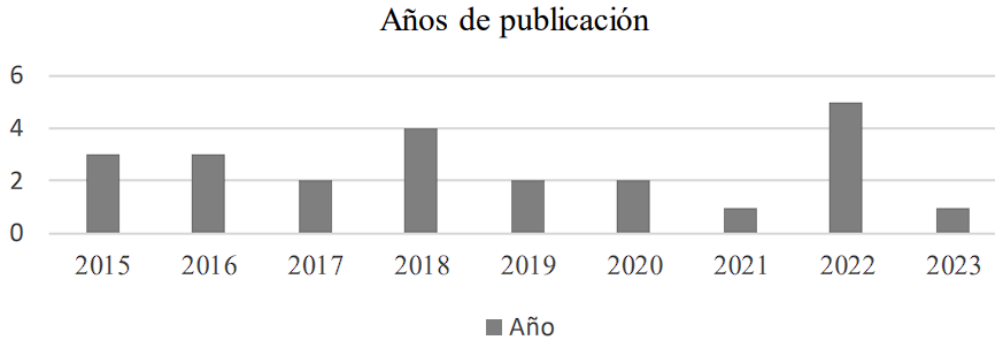


Figura 3. Distribución de estudios por año de publicación.

Para el proceso de síntesis de la información se optó por la estrategia de síntesis narrativa. Este tipo de síntesis descrita por Popay et al. (2006) implica integrar los hallazgos de múltiples estudios en una narrativa coherente que resuma la evidencia disponible sobre nuestro tema de interés. A continuación, se presentan las respuestas a cada pregunta de investigación.

4.1 PI1. ¿Cuáles son las técnicas y métodos de enseñanza que sirven para enseñar programación a niños y jóvenes con discapacidad visual?

De los 23 estudios revisados, se identificaron 5 diferentes métodos de aprendizaje, el aprendizaje háptico, la resolución de casos, el trabajo en equipo, la enseñanza activa y el enfoque basado en problemas como se muestra en la Figura 4. Los métodos de resolución de casos, trabajo en equipo, enseñanza activa y enfoque basado en problemas tuvieron una menor frecuencia de aparición, todos encontrados en el estudio S23.

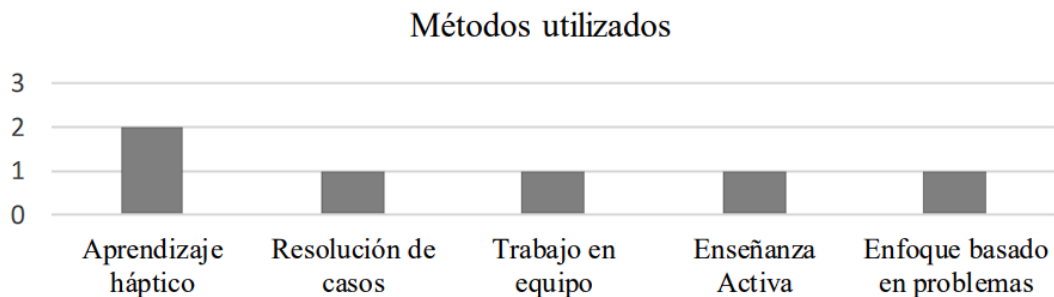


Figura 4. Frecuencia de métodos.

El aprendizaje háptico presenta una mayor frecuencia en los resultados, ya que se basa en el uso de materiales físicos o manipulativos para facilitar el aprendizaje y la comprensión de conceptos abstractos o complejos [10]. En el estudio S02, se utilizan bloques físicos 2D para enseñar programación de forma táctil. Una cámara captura y procesa la información, luego un compilador traduce el código, y los resultados se presentan mediante voz.

Por otro lado, los métodos de enseñanza con una mejor frecuencia fueron el enfoque de enseñanza activa y el enfoque basado en problemas. La enseñanza activa implica la participación de los estudiantes en el proceso de aprendizaje, fomentando su implicación, reflexión y construcción de conocimientos (Torres et al., 2014). Por su parte, el enfoque basado en problemas se basa en presentar a los estudiantes situaciones complejas o desafiantes que requieren su análisis crítico, investigación y búsqueda de soluciones (Labra et al., 2011).

Además, se identificaron cinco técnicas de enseñanza en cuatro artículos diferentes. Las técnicas de aprendizaje colaborativo y la técnica de aprendizaje interactivo y envolvente fueron las más mencionadas, como se muestra en la Figura 4. El aprendizaje háptico presenta una mayor frecuencia en los resultados, ya que se basa en el uso de materiales físicos o manipulativos para facilitar el aprendizaje y la comprensión de conceptos abstractos o complejos (Torres et al., 2014). En el estudio S02, se utilizan bloques físicos 2D para enseñar programación de forma táctil. Una cámara captura y procesa la información, luego un compilador traduce el código, y los resultados se presentan mediante voz.

Por otro lado, los métodos de enseñanza con una mejor frecuencia fueron el enfoque de enseñanza activa y el enfoque basado en problemas. La enseñanza activa implica la participación de los estudiantes en el proceso de aprendizaje, fomentando su implicación, reflexión y construcción de conocimientos (Del Carmen Bernal González & Dueñas, 2009). Por su parte, el enfoque basado en problemas se basa en presentar a los estudiantes situaciones complejas o desafiantes que requieren su análisis crítico, investigación y búsqueda de soluciones (Labra et al., 2011).

Además, se identificaron cinco técnicas de enseñanza en cuatro artículos diferentes. Las técnicas de aprendizaje colaborativo y la técnica de aprendizaje interactivo y envolvente fueron las más mencionadas, como se muestra en la Figura 5.

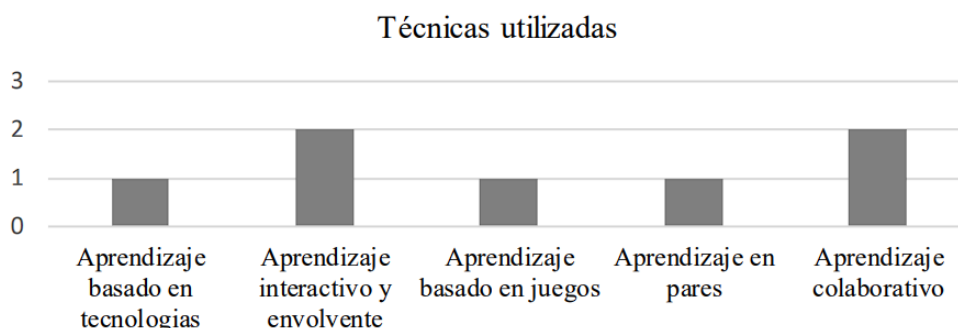


Figura 5. Frecuencia de técnicas.

El aprendizaje colaborativo promueve la construcción conjunta del conocimiento, alentando la interacción y cooperación en grupos de estudiantes. Se basa en la idea de que trabajar juntos, compartir ideas y resolver problemas facilita el aprendizaje (Chew et al., 2020). Un ejemplo concreto de esta técnica es la enseñanza en pares utilizada en el artículo S23 donde un estudiante universitario de informática enseña conceptos y prácticas de programación a un estudiante de secundaria con discapacidad visual utilizando lectores de pantalla y software de programación adaptados.

La segunda técnica de aprendizaje que más se reportó fue la técnica de aprendizaje interactivo y envolvente. Esta técnica se basa en la premisa de que los estudiantes aprenden mejor cuando están activamente comprometidos y participan de manera activa en su entorno de aprendizaje (Freeman et al., 2014). El artículo S11 destaca por su enfoque en promover la interacción con materiales educativos para estudiantes con discapacidad visual. En este estudio, los alumnos utilizaban bloques físicos para identificar variables y luego los unían de manera secuencial para representar declaraciones lógicas.

4.2 PI2. ¿Cuáles son las herramientas disponibles para la enseñanza de programación dirigidas a niños y jóvenes con discapacidad visual?

Una herramienta es un dispositivo que realiza o ayuda en el desempeño de tareas de procesos de usuario u organización que respaldan, directa o indirectamente, el logro de objetivos de producción («ISO/IEC/IEEE International Standard - Systems and software engineering--Vocabulary», 2017). Puede ser tanto física como digital.

La figura 6 muestra la frecuencia por tipo de herramienta donde el lector de pantalla fue la herramienta más mencionada durante la recopilación de información estando presente en los estudios S3, S4, S6, S10, S14, S15, S17, S18, S21, S22, S23 es una herramienta de software ampliamente utilizada que brinda asistencia a las personas con discapacidad visual al proporcionarles información auditiva sobre el contenido mostrado en la pantalla de un dispositivo electrónico (Hayward et al., 2004).

El segundo tipo de herramienta más reportada fue la háptica. Las herramientas hápticas juegan un papel fundamental al proporcionar una experiencia táctil y sensorial que compensa la falta de visión. Estas herramientas permiten a las personas con discapacidad visual explorar y percibir el entorno de manera más completa, brindándoles información táctil y sensorial que les ayuda a comprender y navegar por su entorno estando presente en los estudios S7, S9, S10, S12, S17, S18, S20, S21, S23.

En las figuras 7, 8, 9 y 10 se puede apreciar con mayor detalle qué herramientas se encontraron en cada categoría.

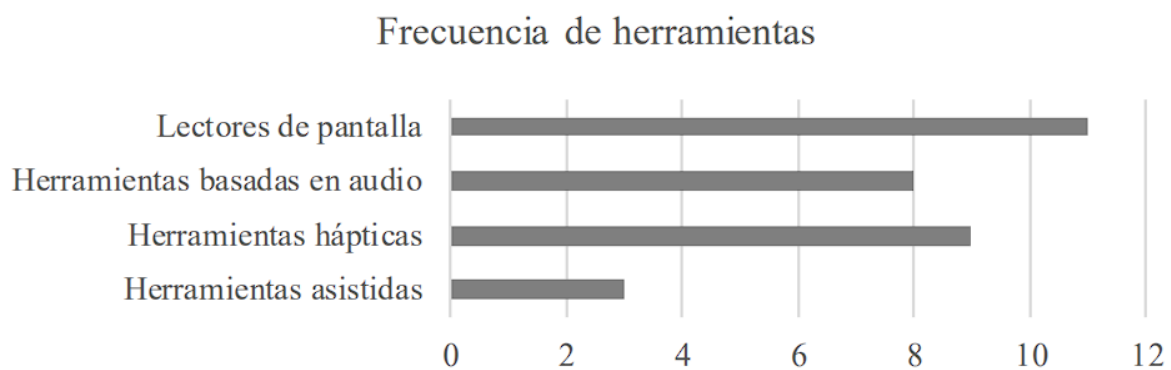


Figura 6. Frecuencia de herramientas.

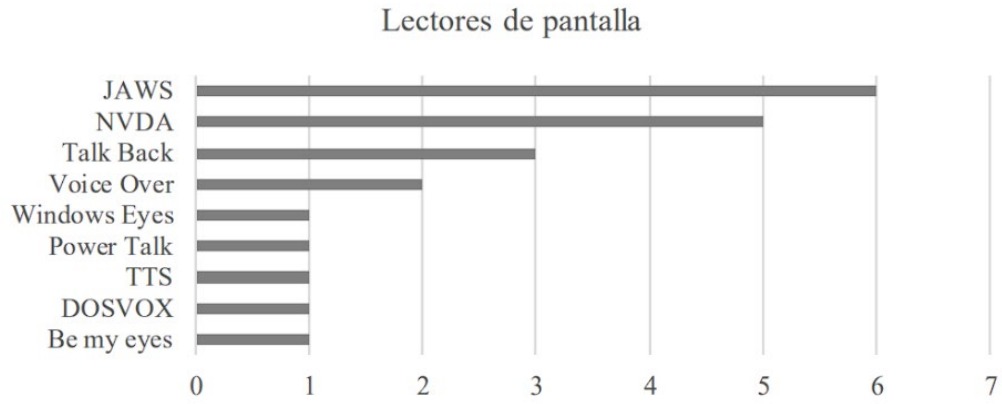


Figura 7. Frecuencia lectores de pantalla.

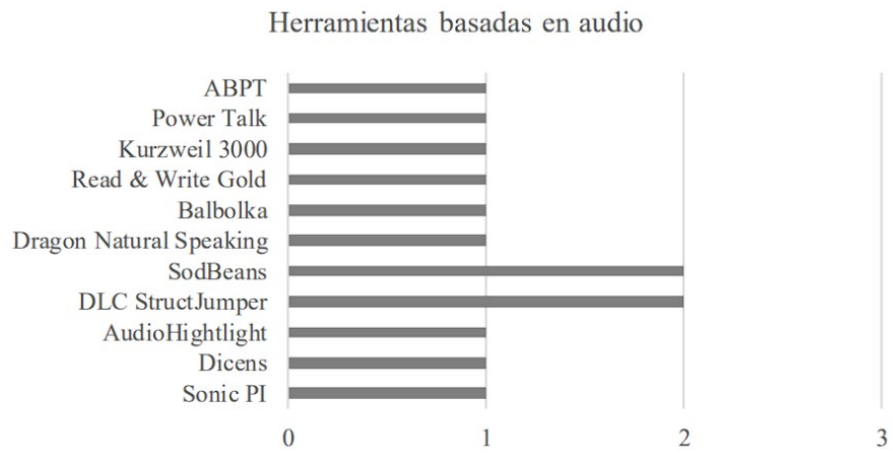


Figura 8. Frecuencia herramientas basadas en audio.

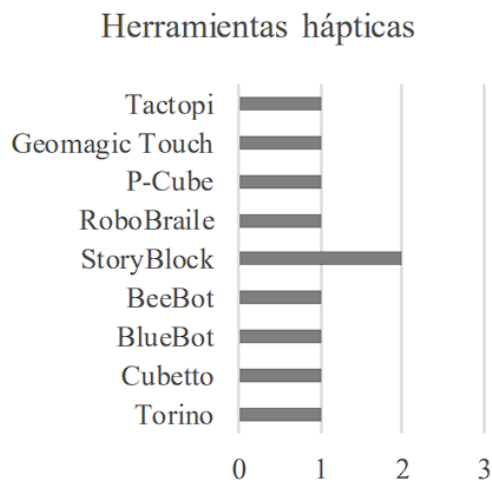


Figura 9. Frecuencia herramientas hápticas.

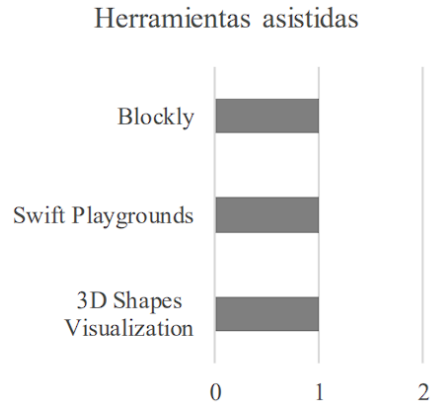


Figura 10. Frecuencia herramientas asistidas.

4.2 PI3. ¿Cuál es la efectividad de las técnicas, métodos y herramientas de enseñanza de programación en niños y jóvenes con discapacidad visual?

La efectividad se refiere a la capacidad de lograr resultados deseados u objetivos 7 establecidos. Es utilizada para evaluar y medir el grado de éxito o eficacia en la consecución de metas específicas («ISO/IEC/IEEE International Standard - Systems and software engineering--Vocabulary», 2017).

La evaluación de la efectividad se vio obstaculizada debido a la falta de pruebas realizadas por muchos autores con relación a los métodos, técnicas o herramientas que se analizaron. Esto dificulta el establecimiento de parámetros precisos para la medición de la efectividad. No obstante, en los estudios S9, S12 y S23 se empleó el tiempo de resolución de problemas como indicador de efectividad. Aunque este enfoque puede ser válido en ciertos casos, es esencial recordar que existen otros factores que pueden influir en la eficacia de un método, técnica o herramienta.

La comprensión se refiere a la habilidad de entender y procesar la información que se presenta a través de distintas formas de comunicación, como el lenguaje oral o escrito, imágenes, gestos y otros medios (Perkins, 1999). Cuando se evalúa la comprensión, es crucial adoptar enfoques personalizados que se adapten a las necesidades y capacidades individuales. Para medirla, existen diversas estrategias, como la evaluación basada en el rendimiento, que se utilizó en los estudios S2, S20 y S23.

En la Figura 11, se pueden observar la frecuencia de los parámetros mencionados en los artículos.

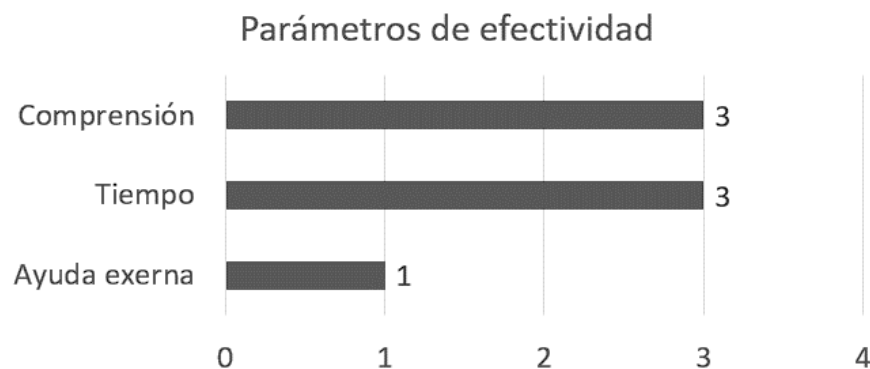


Figura 11. Frecuencia parámetros de efectividad.

5. Amenazas a la validez

En los resultados de esta investigación se identificaron algunas amenazas, como la falta de acceso a la mayoría de los documentos completos. La búsqueda en un número reducido de fuentes aumenta el riesgo de que la muestra de literatura revisada no sea representativa o completa. Otras posibles amenazas incluyen la posibilidad de que los autores que llevaron a cabo las prácticas hayan omitido información relevante, lo que podría conducir a la alteración de los resultados. Estas amenazas se mitigaron mediante el uso de formatos para la síntesis, validaciones de la cadena de búsqueda, y ampliando los años de investigación para capturar un mayor número de estudios primarios.

6. Conclusiones y trabajo futuro

En conclusión, la investigación destaca la importancia de ampliar los métodos de enseñanza para niños y jóvenes con discapacidad visual. Se han identificado enfoques prometedores, como el enfoque háptico, el aprendizaje colaborativo y el aprendizaje interactivo. Además, se ha observado que herramientas como los lectores de pantalla y las herramientas hápticas ofrecen beneficios significativos en el proceso de aprendizaje. Aunque la evaluación de la efectividad plantea desafíos, se sugiere utilizar el tiempo de resolución de problemas como un indicador inicial para evaluar los resultados. Es necesario seguir investigando y desarrollando estrategias educativas inclusivas que faciliten un aprendizaje efectivo y significativo en programación desde una edad temprana para todos los estudiantes con discapacidad visual. Como parte de trabajos futuros, se espera que esta investigación sirva como base para el desarrollo de conocimiento en este ámbito.

Referencias

World Health Organization: WHO. (2023, 10 agosto). *Ceguera y discapacidad visual*. <https://www.who.int/es/news-room/fact-sheets/detail/blindness-and-visual-impairment>

Clasificación estadística internacional de enfermedades y problemas de salud relacionados (11ª ed.). (2019). Organización Mundial de la Salud. <https://icd.who.int/en>

Delgado, P. (2022, 3 noviembre). *Educación STEM: ¿Qué es y cómo sacarle provecho?* Observatorio / Instituto para el Futuro de la Educación. <https://observatorio.tec.mx/edu-news/educacion-stem-que-es-y-como-sacarle-provecho/>

Wing, J. M. (2006). Computational thinking. *Communications of The ACM*, 49(3), 33-35. <https://doi.org/10.1145/1118178.1118215>

Al-Ratta, N. M., & Al-Khalifa, H. S. (2013). Teaching programming for blinds: A review. *Fourth International Conference on Information and Communication Technology and Accessibility*. <https://doi.org/10.1109/icta.2013.6815285>

Li, Y. (2016). Teaching programming based on Computational Thinking. *2016 IEEE Frontiers in Education Conference*. <https://doi.org/10.1109/fie.2016.7757408>

Kitchenham, B., Budgen, D., & Brereton, P. (2015). Evidence-Based software engineering and systematic reviews. En *Chapman and Hall/CRC eBooks*. <https://doi.org/10.1201/b19467>

Zhang, H., Babar, M. A., & Tell, P. (2011). Identifying relevant studies in software engineering. *Information & Software Technology*, 53(6), 625-637. <https://doi.org/10.1016/j.infsof.2010.12.010>

Popay, J., Roberts, H.M., Sowden, A.J., Petticrew, M., Arai, L., Rodgers, M., & Britten, N. (2006). Guidance on the conduct of narrative synthesis in systematic Reviews. A Product from the ESRC Methods Programme. Version 1.

Torres Cosío, Verónica; Vázquez Alejandre, Emilia Jazmín; Cossio Franco, Edgar (2014). Accesibilidad de los materiales educativos a través de la percepción háptica. En: No Solo Usabilidad, nº 13, 2014. <nosolousabilidad.com>. ISSN 1886-8592

Del Carmen Bernal González, M., & Dueñas, M. S. M. (2009). METODOLOGÍAS ACTIVAS PARA LA ENSEÑANZA y EL APRENDIZAJE. *Revista panamericana de pedagogía*. <https://doi.org/10.21555/rpp.v0i14.1790>

Labra, P., Kokaly, M. E., Herrera, C. I., Concha, A., Sasso, P., & Vergara, M. (2011). El enfoque ABP en la formación inicial docente de la Universidad de Atacama: el impacto en el quehacer docente. *Estudios pedagógicos*, 37(1), 167-185. <https://doi.org/10.4067/s0718-07052011000100009>

Chew, S. W., Huang, X., Hsu, F., & Chen, N. (2020). Enhancing Critical Thinking Skills of Elementary School Students through Collaborative Learning. *2020 IEEE 20th International Conference on Advanced Learning Technologies (ICALT)*. <https://doi.org/10.1109/icalt49669.2020.00082>

Freeman, S., Eddy, S. L., McDonough, M. J., Smith, M. K., Okoroafor, N., Jordt, H., & Wenderoth, M. P. (2014). Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences of the United States of America*, 111(23), 8410-8415. <https://doi.org/10.1073/pnas.1319030111>

"ISO/IEC/IEEE International Standard - Systems and software engineering--Vocabulary," in *ISO/IEC/IEEE 24765:2017(E)*, vol., no., pp.1-541, 28 Aug. 2017, doi: 10.1109/IEEESTD.2017.8016712.

Hayward, V., Astley, O. R., Cruz-Hernandez, M., Grant, D. F., & Robles-De-La-Torre, G. (2004). Haptic interfaces and devices. *Sensor Review*, 24(1), 16-29. <https://doi.org/10.1108/02602280410515770>

"ISO/IEC/IEEE International Standard - Systems and software engineering--Vocabulary," in *ISO/IEC/IEEE 24765:2017(E)*, vol., no., pp.1-541, 28 Aug. 2017, doi: 10.1109/IEEESTD.2017.8016712.

Perkins, D. (1999). ¿Qué es la comprensión? En W. Stone (Comp.), *La Enseñanza para la Comprensión: vinculación entre la investigación y la práctica*. Buenos Aires, Argentina: Paidós.

Notas Biográficas

	<p>Raúl Arturo Peredo-Estudillo es estudiante de la Licenciatura en Ingeniería de Software de la Facultad de Estadística e Informática de la Universidad Veracruzana. Sus intereses principales son el desarrollo móvil Android y IOS, diseño de interfaces UI y UX y por temas de investigación enfocados hacia la accesibilidad.</p>
	<p>Juan Carlos Pérez-Arriaga es Maestro en Ciencias de la Computación por la Fundación Arturo Rosenblueth. Profesor de la Facultad de Estadística e Informática de la Universidad Veracruzana, desde el 2006. Sus intereses incluyen seguridad en el desarrollo de software y desarrollo de software accesible.</p>
	<p>Gerardo Contreras Vega es Licenciado en Informática, cuenta con una maestría en Ciencias de la Computación y estudia el Doctorado en Tecnologías de la Información. Cuenta con experiencia de 27 años en la Universidad Veracruzana, donde ha laborado como Técnico Académico y Profesor de Tiempo Completo. Sus áreas de interés son la seguridad de la información, el manejo de servicios e infraestructura de red y la accesibilidad para personas con discapacidad. Disfruta de utilizar y promover el software libre.</p>
	<p>Ángel J. Sánchez-García es doctor en Inteligencia Artificial por la Universidad Veracruzana en 2018. Desde 2017 es profesor de tiempo completo de la Facultad de Estadística e Informática, Universidad Veracruzana. Sus áreas de investigación de interés son Aprendizaje Máquina, Minería de datos, Estimación en Ingeniería de Software.</p>

