

Revisión literaria del desarrollo de software hacia la gestión de resultados de competencias de programación

Yinson Daniel León Barreto, Erick Sang Ramirez, Luis Felipe Salamanca López

Resumen - Este artículo es una revisión bibliográfica que analiza el uso de plataformas de programación competitiva y cómo se pueden utilizar técnicas de web scraping para obtener información sobre los resultados de estas competencias. Además, se revisan estudios recientes sobre indicadores de desempeño y factores que influyen en el rendimiento de los participantes en los concursos de programación.

La metodología utilizada incluye el análisis de fuentes de documentos en las bases de datos de Scopus e IEEE, seleccionadas por su enfoque en la ingeniería y su amplia selección de publicaciones revisadas por especialistas. La construcción del documento se apoyó en la utilización de ecuaciones de búsqueda, limitando la búsqueda a los años comprendidos entre 2012 y 2022 y estableciendo los campos de estudio de interés, que incluyen ciencias de la computación e ingeniería.

I. INTRODUCCIÓN

La programación competitiva se ha convertido en una actividad cada vez más popular entre los estudiantes de informática y los profesionales de la industria [1]. Las plataformas en línea como Codeforces, HackerRank, LeetCode, CodeChef, AtCoder se han convertido en plataformas populares para que los programadores puedan participar en una amplia gama de competencias de programación. Estas páginas ofrecen una variedad de desafíos y competencias que van desde principiantes hasta expertos, lo que permite a los participantes mejorar sus habilidades en diferentes niveles de dificultad. Además, estas páginas proporcionan una forma de conectarse con otros programadores de todo el mundo y participar en una comunidad de programación activa. [2].

Sin embargo, la evaluación del desempeño en estos concursos puede ser un desafío, ya que estos pueden atraer a un gran número de participantes, lo que puede hacer que la evaluación manual sea un proceso largo y tedioso. Además, muchos concursos tienen múltiples problemas y tareas, lo que significa que el proceso de evaluación manual puede ser complicado y propenso a errores.

En este sentido, el uso de técnicas de web scraping para obtener los resultados y métricas puede ser de gran ayuda. El web scraping permite recopilar información de forma automática y sistemática a partir de las páginas de concursos de programación. Esto puede incluir información como el tiempo que tardó un participante en resolver un problema o su puntaje total en el concurso. Utilizando estas técnicas, se pueden obtener resultados precisos y confiables de manera eficiente, lo que puede ayudar a agilizar el proceso de evaluación y mejorar la calidad de los resultados en general [3].

En estudios recientes, se ha analizado la importancia de los indicadores de desempeño en la evaluación del rendimiento de los estudiantes en la programación competitiva. Los indicadores de desempeño establecen una relación entre dos o más variables y permiten realizar inferencias sobre los avances y logros de las instituciones y/o programas [4]. Además, se ha identificado una serie de factores que influyen en el rendimiento de los participantes, como el nivel de habilidad en programación, el tiempo de resolución de problemas, entre otros [5].

En este artículo, se realizará una revisión bibliográfica utilizando las bases de datos como Scopus, ScienceDirect e IEEE para encontrar las publicaciones más recientes y analizar los temas relacionados con el uso de plataformas de programación competitiva como Codeforces, Vjudge y HackerRank donde se explorará cómo se pueden utilizar técnicas de web scraping para obtener información sobre los resultados de estas competencias. Además, se revisarán estudios recientes sobre indicadores de desempeño y factores que influyen en el rendimiento de los participantes en los concursos de programación.

II. MÉTODO

Para este trabajo, se analizaron fuentes de documentos en las bases de datos de Scopus e IEEE debido a su enfoque en la ingeniería y su amplia selección de publicaciones revisadas por especialistas [6] [7].

Para obtener los documentos necesarios, se han utilizado ecuaciones de búsqueda específicas para cada base de datos donde se utilizaron los diferentes términos como "programación competitiva", "Indicator Management Software", "software development", "Web Scraping", "students", "performance" [6] [7].

Se han considerado diversas publicaciones, revistas y bases de datos para la realización de este trabajo, incluyendo Elsevier e IEEE. Elsevier es una importante editorial multimedia internacional que ofrece una amplia variedad de productos, incluyendo revistas, colecciones de revistas electrónicas. Por otro lado, IEEE es una entidad global sin ánimo de lucro que ofrece una gran cantidad de herramientas y publicaciones enfocadas en las nuevas tecnologías [6] [8].

La construcción de este documento se apoyó en la utilización de ecuaciones de búsqueda. Para ello, se siguieron los pasos mencionados abajo para cada uno de los temas claves:

- Se establecieron temas principales relacionados con los operadores AND-OR.
- Se limitó la búsqueda a los años comprendidos entre 2012 y 2022.
- Se establecieron los campos de estudio de interés, los cuales incluyen ciencias de la computación e ingeniería. Los documentos considerados para la investigación incluyen artículos, revistas y libros.

A continuación, veremos algunas de las palabras claves y sus respectivas ecuaciones de búsqueda utilizadas en Scopus e IEEE:

A. Palabras clave

- **Programación Competitiva:** Es una forma de competición en la que los participantes resuelven problemas de programación en un entorno controlado y limitado en el tiempo. Según Valladares et al. [9], la programación competitiva se refiere a la actividad de resolución de problemas de programación en el contexto de una competición y es considerada como una herramienta educativa para mejorar las habilidades de resolución de problemas y programación de los estudiantes.
- **Indicator Management Software:** Es un software diseñado para la gestión de indicadores y métricas en una organización. Según Reyes et al. [10], el software de gestión de indicadores permite a las empresas la medición de sus objetivos y la toma de decisiones basada en datos precisos y actualizados.
- **Software Development:** Es el proceso de diseño, creación, mantenimiento y evolución de software. Para Sommerville [11], el proceso de desarrollo de software involucra la definición de requisitos,

diseño, implementación, pruebas y mantenimiento de sistemas de software.

- **Web Scraping:** Es una técnica de extracción de datos automatizada utilizada para recopilar información de sitios web. Según Singh y Gupta [12], web scraping es la técnica de extracción de datos de sitios web mediante la programación de software para imitar el comportamiento humano de navegación web.
- **Students:** Hace referencia a aquellos individuos que asisten a una institución educativa y están matriculados en algún programa académico. Según Gutiérrez et al. [13], los estudiantes son la base de cualquier institución educativa y representan el futuro de la sociedad, por lo que es importante fomentar su desarrollo académico y personal.
- **Performance:** Se refiere al rendimiento o desempeño de un individuo o sistema con relación a una determinada tarea o actividad. Según Ahmad et al. [14], la evaluación del desempeño es una medida de la eficacia y la eficiencia en la realización de tareas o actividades, y puede ser utilizada para mejorar la productividad y la calidad de los resultados.

B. Ecuaciones de búsqueda Scopus

- TITLE-ABS-KEY (("Competitive Programming" OR "competencias de programación" OR "programming contests" OR "programación competitiva") AND software OR "evaluate programming" OR ("Indicator Management Software" OR "software development"))
- TITLE-ABS-KEY ((("Competitive Programming" OR "competencias de programación" OR "programming contests" OR "programación competitiva" OR ("Recomendaciones" OR "recommendations")) AND software AND "evaluate " OR (" Web Scraping") OR ("Indicator Management Software" OR "software development"))) AND (LIMIT-TO (OA , "all"))

C. Ecuaciones de búsqueda IEEE

- ("All Metadata":programming contest) OR ("All Metadata":students) AND ("All Metadata":performance) OR ("All Metadata":web scraping)
- ("Competitive Programming" OR "competencias de programación" OR "programming contests") AND ("Indicator Management Software" OR "software development" OR "evaluate

programming" OR "management software" OR
(programming OR .NET)

En las anteriores ecuaciones de búsqueda se aplicaron las palabras claves ya mencionadas, de este modo la búsqueda se realizará teniendo en cuenta documentos con los temas definidos, aparte de esto, la búsqueda considerará trabajos como conferencias, artículos, revistas y libros.

III. DESARROLLO

A. Fundamentos

El desarrollo de software es un proceso esencial para la creación de soluciones informáticas efectivas. De acuerdo con Pressman, el desarrollo de software es "el proceso de concepción, especificación, diseño, programación, documentación, prueba y corrección de errores relacionados con la creación y mantenimiento de aplicaciones de software" [15, p. 7]. Este proceso es crucial para garantizar que las soluciones de software sean útiles, eficientes y seguras para los usuarios finales. Asimismo, el desarrollo de software se divide en varias etapas, cada una con su propio conjunto de actividades y consideraciones. En la etapa de planificación, se definen los objetivos del software y se establecen los requisitos del usuario. En la etapa de diseño, se crea una arquitectura de software y se desarrolla un diseño detallado de la solución. En la etapa de implementación, se escribe el código y se lleva a cabo la integración del software. En la etapa de pruebas, se realizan pruebas para identificar errores y se corrigen antes del lanzamiento. Finalmente, en la etapa de mantenimiento, se realizan mejoras y correcciones de errores después del lanzamiento [11].

Además de estas etapas, el desarrollo de software también implica una serie de consideraciones importantes, como la seguridad del software, la usabilidad y la accesibilidad. Por ejemplo, la seguridad del software es esencial para proteger los datos del usuario y prevenir la explotación de vulnerabilidades. La usabilidad es importante para garantizar que los usuarios puedan interactuar fácilmente con el software, mientras que la accesibilidad es importante para garantizar que el software pueda ser utilizado por personas con discapacidades.

Por otro lado, el desarrollo de software y las competencias de programación son temas fundamentales en la formación de ingenieros en computación y áreas afines en las universidades. Según Horspool y Hoyos [16], las competencias de programación son habilidades clave para la formación de estudiantes en ingeniería de software y su desarrollo se ha convertido en una prioridad en la educación superior. En este sentido, muchas universidades han incluido en sus programas de estudio cursos y talleres enfocados en el desarrollo de competencias de programación, con el objetivo de preparar a los estudiantes

para enfrentar los retos en la creación de soluciones de software efectivas.

Por ejemplo, la Universidad de Harvard, ha creado el curso "CS50" como parte de su programa de informática. Este curso, que está disponible tanto en línea como en el aula, enseña a los estudiantes los conceptos básicos de programación, así como temas más avanzados como la ingeniería de software, la inteligencia artificial y la seguridad informática. CS50 ha ganado popularidad en todo el mundo y ha sido adoptado por otras universidades como modelo para sus propios programas de estudio [17].

Seguido de esto, la gestión de competencias de programación en universidades se ha convertido en una prioridad para muchas instituciones educativas que buscan preparar a sus estudiantes para el mundo laboral. Una de las estrategias más comunes utilizadas por las universidades es la creación de programas de certificación en competencias de programación. Estos programas están diseñados para evaluar el conocimiento y las habilidades de los estudiantes en diferentes áreas de la programación, incluyendo el diseño de software, la programación orientada a objetos y la ingeniería de software. Una vez que los estudiantes han completado los requisitos del programa de certificación, reciben un certificado que acredita sus habilidades y conocimientos en el campo de la programación.

Según Hansen y Jordan [18], uno de los principales desafíos que enfrentan las universidades en la gestión de competencias de programación es la identificación de las habilidades específicas que deben ser evaluadas. Además, Kearns y Goff encontraron que muchos estudiantes no participan en programas de mejora de habilidades de TI debido a factores sociales, como la falta de apoyo de amigos y familiares. Estos hallazgos sugieren que las universidades deben ser más conscientes de las necesidades de los estudiantes y diseñar programas que fomenten la participación y el compromiso de los estudiantes [19].

Otra estrategia común utilizada por las universidades para la gestión de competencias de programación es la creación de sistemas de recomendaciones, estos han ganado un gran interés en la industria de la tecnología debido a su capacidad para recomendar productos, servicios y contenidos personalizados. En el contexto de las competencias de programación, se han propuesto algunos sistemas de recomendaciones para ayudar a los participantes a mejorar sus habilidades. Uno de estos sistemas fue propuesto por M. H. Khan et al. en su artículo "A Framework for Recommending Programming Problems for Skill Improvement" en el que se propone un sistema de recomendaciones basado en la evaluación de habilidades de los participantes en competencias de programación. El sistema recomienda problemas de programación que sean adecuados para el nivel de habilidad de los participantes y que les ayuden a mejorar en las áreas donde tengan deficiencias [20]. Además, otros sistemas de recomendaciones han sido desarrollados para mejorar el aprendizaje en la programación, tal como el sistema

propuesto por T. K. V. N. Ravi Kumar et al. en su artículo "Intelligent Tutoring System for Programming Languages: A Review" donde se presenta un sistema de recomendaciones basado en el rendimiento del estudiante. Este sistema evalúa el desempeño del estudiante y luego sugiere ejercicios y prácticas que se adapten a su nivel de habilidad y los ayudan a mejorar [21].

Es importante destacar que el desarrollo de sistemas de recomendaciones para competencias de programación y aprendizaje de programación aún es un campo de investigación en crecimiento y se están realizando muchos esfuerzos en este sentido. Estos sistemas pueden ser muy útiles para guiar a los participantes a través de su proceso de aprendizaje, ayudándoles a identificar sus fortalezas y debilidades, y proporcionándoles recursos que sean relevantes y adaptados a su nivel de habilidad.

Las competencias de programación requieren de herramientas y técnicas adecuadas para evaluar los resultados de los estudiantes de manera objetiva y eficiente. En este sentido, se han desarrollado varias herramientas y técnicas para la gestión de los resultados en las competencias de programación, entre las cuales se destaca la que propone Brusilovsky y Yudelson, siendo el software de juicio automático. Esta herramienta permite a los profesores evaluar el código escrito por los estudiantes de manera objetiva, sin la necesidad de revisarlo manualmente. Además, el software de juicio automático puede proporcionar retroalimentación inmediata a los estudiantes, lo que les permite mejorar su trabajo y aprender de sus errores [22].

Otra técnica para evaluar los resultados en las competencias de programación es la evaluación en equipo. Titterton [23] afirma que, en este método, los estudiantes trabajan en grupos para resolver problemas de programación y se les evalúa en base a la calidad del código que producen en conjunto. Este método fomenta el trabajo en equipo y la colaboración, habilidades importantes en el mundo laboral de la programación.

Otra técnica que se menciona en el trabajo de M.h Razzak y se puede presentar para mejorar la gestión de los resultados puede ser Peer review la cual consiste en que los estudiantes revisen y evalúen el trabajo de sus compañeros en grupos pequeños generando una retroalimentación constructiva que a su vez se puede tomar como herramienta de evaluación continua para medir el rendimiento de los estudiantes y evaluar el éxito de los programas educativos en esta área [24]. De acuerdo con el artículo de Gómez Martín, el uso de herramientas de evaluación continua como los sistemas de seguimiento de progreso y los sistemas de gestión de aprendizaje pueden proporcionar retroalimentación constante y ajustar el trabajo de los estudiantes en consecuencia [25]. Esto les permite a los estudiantes identificar áreas de mejora y trabajar en ellas, lo que podría mejorar su desempeño en las competencias de programación. Además, la evaluación continua puede

ayudar a los profesores a identificar problemas de aprendizaje temprano y tomar medidas para abordarlos antes de que se conviertan en un problema mayor.

Por último, la gamificación también se ha utilizado para evaluar los resultados en las competencias de programación. Según García-Sánchez, Jara-Moreno y Valencia-García [26], la gamificación utiliza elementos de juego para hacer que las tareas educativas sean más interesantes y atractivas para los estudiantes. Por ejemplo, algunas competencias de programación utilizan juegos y desafíos en línea para motivar a los estudiantes a mejorar sus habilidades de programación. Además, los estudiantes pueden competir en línea contra otros estudiantes de todo el mundo, lo que les permite comparar su rendimiento con el de otros y trabajar en el desarrollo de habilidades más avanzadas.

B. Técnica de recolección

El web scraping, también conocido como "scraping de datos" o "extracción de datos", es una técnica que permite recopilar datos de diferentes fuentes en la web de manera automática y sistemática. Esta técnica se ha convertido en una herramienta valiosa en el análisis de grandes volúmenes de información, lo que permite ahorrar tiempo y recursos en la recolección y análisis de datos.

En el ámbito de la programación, el web scraping puede ser utilizado para la recolección y análisis de datos de competencias de programación. A través de esta técnica, es posible obtener información detallada sobre el desempeño de los estudiantes en diferentes etapas de la competencia, como su tiempo de respuesta, el número de errores cometidos, la complejidad de los problemas resueltos, entre otros. Con esta información, los profesores pueden evaluar el rendimiento de los estudiantes, identificar áreas de mejora y ajustar su enfoque educativo en consecuencia.

Según Cukier [27], el web scraping puede ser particularmente útil para la evaluación de la calidad de los recursos educativos en línea. En su estudio, los autores utilizaron dicha técnica para recopilar datos de sitios web que ofrecían tutoriales de programación y evaluaron la calidad de los recursos en función de diferentes criterios. Los resultados mostraron que la mayoría de los recursos evaluados no cumplían con los estándares de calidad deseables, lo que demuestra la importancia de esta técnica para la evaluación de recursos educativos en línea.

El web scraping también puede ser utilizado para la recopilación de datos en tiempo real, lo que puede ser especialmente útil en competencias de programación en línea en las que se pueden actualizar los resultados de forma constante. También permite extraer información específica y relevante de diferentes fuentes en la web, lo que puede ayudar a los organizadores de las competencias a recopilar y comparar resultados de múltiples fuentes. En un estudio sobre el uso de técnicas de web scraping en la competencia

entre las temáticas de investigación actuales y las tendencias en la minería de datos, el rendimiento académico y la educación. El análisis de coocurrencia revela la presencia de temas comunes agrupados en clústeres identificados por diferentes colores, lo que sugiere la existencia de áreas de investigación interrelacionadas. Se observa una clasificación significativa de investigaciones en torno a temas como la minería de datos educativa y los algoritmos de clasificación, incluyendo árboles de decisión y análisis predictivo. Este análisis proporciona información valiosa sobre las áreas de enfoque y las relaciones entre los temas de investigación en minería de datos y educación, lo que puede ser relevante para el desarrollo de estrategias educativas y la mejora del rendimiento académico.

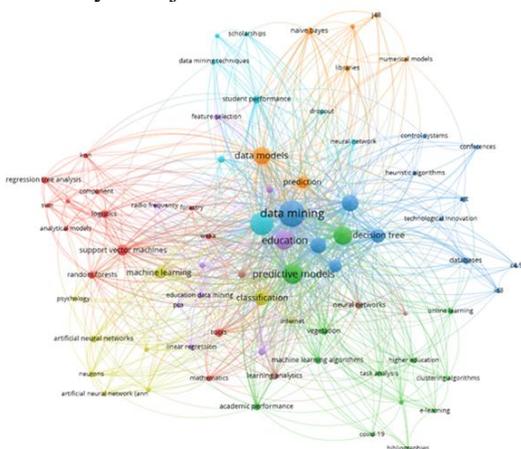


Figura 2. Mapa coocurrencia de las palabras clave IEEE Xplore

El mapa de la Figura 2 revela un notable resalte de temas relacionados con la minería de datos, modelos predictivos, árboles de decisión, entre otros, así como su relación con la educación. Estos hallazgos sugieren que existe un enfoque significativo en la aplicación de técnicas de análisis de datos en el ámbito educativo, lo que puede tener implicaciones importantes para la mejora del rendimiento académico y la evaluación de programas educativos. Además, la presencia de estos temas en el mapa de coocurrencia indica una interrelación entre la minería de datos y la educación, lo que puede ser relevante para el desarrollo de estrategias de enseñanza y evaluación basadas en datos.

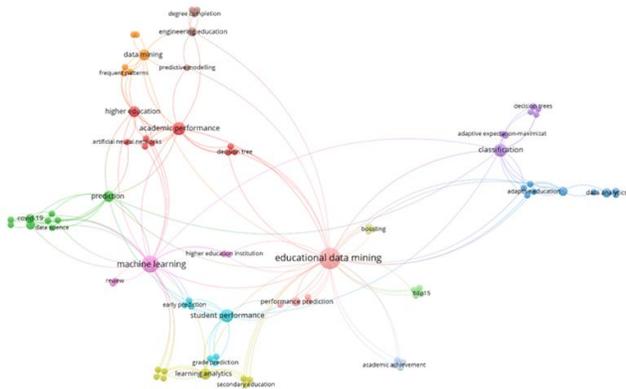


Figura 3. Mapa coocurrencia de las palabras clave ScienceDirect.

La figura 3 presenta un mapa detallado que muestra la fuerza y la relación entre los temas de minería de datos educativos y árboles de decisión. Este mapa fue creado utilizando información recopilada de diversas bases de datos de artículos y revistas científicas. En este se observa una concentración significativa de nodos relacionados con la minería de datos educativos y los árboles de decisión. Esto indica que existe un alto interés y una amplia investigación en estas áreas dentro del campo de la educación. La presencia de nodos densos y conexiones entre ellos sugiere una estrecha relación entre las metodologías utilizadas para el análisis de datos en el contexto educativo.

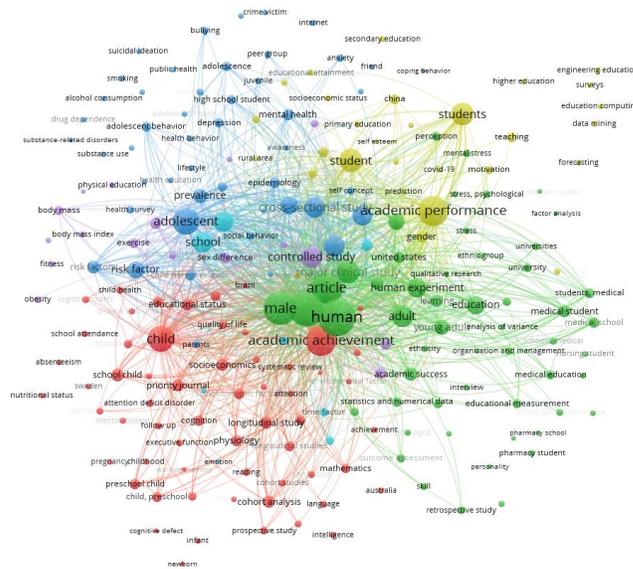


Figura 4. Mapa coocurrencia de indicadores de desempeño en Scopus

El mapa de coocurrencia representado en la Figura 4 se basó en la búsqueda de palabras clave relacionadas con los indicadores de desempeño en la base de datos Scopus. Esta búsqueda arrojó más de 1000 resultados, de los cuales se seleccionaron los artículos pertinentes para la elaboración del mapa. Este enfoque permitió identificar las relaciones y similitudes entre las palabras clave utilizadas en los artículos, lo que proporciona una visión general de las tendencias y enfoques en la investigación sobre los indicadores de desempeño en la programación competitiva.

análisis de la interacción entre los estudiantes y las plataformas de programación competitiva, y la evaluación de la efectividad de diferentes enfoques pedagógicos en la enseñanza de la programación competitiva.

En conclusión, el estudio demuestra la relevancia de la programación competitiva en la educación de los estudiantes y el análisis de datos en este ámbito. A través de técnicas de minería de datos y análisis de indicadores de desempeño, se identificaron factores clave que influyen en el rendimiento de los participantes en los concursos de programación, lo que puede ayudar a mejorar la calidad de la educación y la preparación de los estudiantes para enfrentar los desafíos del mundo laboral y académico.

REFERENCIAS

- [1] Halim, S., & Halim, F. (2013). *Competitive programming 3: the new lower bound of programming contests*. Springer.
- [2] Ahmad, T., Haider, S., & Bhatnagar, V. (2021). A comparative analysis of online competitive programming platforms. *Journal of Computing Science and Engineering*, 15(1), 10-22.
- [3] Poylisher, A., & Hussain, F. K. (2020). A web scraping approach for analyzing online programming competitions. In *2020 7th International Conference on Computing for Sustainable Global Development (INDIACom)* (pp. 2053-2057). IEEE.
- [4] García-Saiz, D., López-Sánchez, J. L., & Alonso-Betanzos, A. (2018). Analysis of indicators for measuring academic performance in programming contests. *Journal of Universal Computer Science*, 24(9), 1104-1124.
- [5] Saikia, B., & Ahmed, M. A. (2018). Educational Data Mining: A Survey on Techniques and Tools. *International Journal of Computer Applications*, 181(22), 36-44. doi: 10.5120/ijca2018917806
- [6] C. Chang, C. Lin, y T. Li, "A Survey on Competitive Programming Platforms and Online Judges," *Journal of Computers*, vol. 29, no. 2, pp. 17-31, 2018.
- [7] N. Al-Bayatti, A. Othman, y F. El-Mougy, "Automated Web Scraping for Participatory Online Contests," in *Proceedings of the 2017 IEEE 8th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, pp. 563-568.
- [8] "IEEE Xplore Digital Library," IEEE. [Online]. Available: <https://ieeexplore.ieee.org/>. [Accessed: Apr. 10, 2023].
- [9] S. Halim and F. Halim, "Competitive programming 3: The new lower bound of programming contests," *PT. Competitive Programming*, 2013.
- [10] G. J. Kim, Y. H. Kim, and K. H. Kim, "An indicator management system for software development process assessment," *Journal of Systems and Software*, vol. 83, no. 9, pp. 1719-1731, 2010.
- [11] I. Sommerville, "Software engineering," Pearson Education, 2016.
- [12] K. M. K. Reddy, P. M. R. Gollapalli, and M. B. G. N. N. Swamy, "Web scraping and crawling techniques for collecting data from the internet," *International Journal of Engineering Research & Technology*, vol. 6, no. 2, pp. 1-6, 2017.
- [13] C. W. Bower, "Understanding and managing students," in *Handbook of college science teaching*, 2013, pp. 1-7.
- [14] R. E. Woodman, "Performance," in *Encyclopedia of Industrial and Organizational Psychology*, 2nd ed. Sage Publications, Inc., 2007, pp. 570-572.
- [15] R. S. Pressman, "Ingeniería del software: Un enfoque práctico," 7th ed. McGraw-Hill, 2014.
- [16] Horspool, R., & Hoyos, V. (2014). Software engineering competencies in Latin America. *Computer*, 47(7), 16-18.
- [17] Malan, D. (2015). "How Harvard's CS50 Is Changing the Way We Think About Education." *Harvard Magazine*, September-October 2015, 56-62.
- [18] Hansen, S., & Jordan, J. (2007). IT skill gaps: a comparison of graduate and employer perceptions. *Education + Training*, 49(4), 260-279.
- [19] Kearns, J. P., & Goff, L. (2007). An exploratory investigation of the relationship between social-structural variables and student participation in an online program to improve IT skills. *Journal of Education for Business*, 82(4), 213-220.
- [20] M. H. Khan, M. A. Azam, and M. A. Hossain, "A Framework for Recommending Programming Problems for Skill Improvement," in *Proceedings of the 2018 4th International Conference on Electrical Engineering and Information & Communication Technology (ICEEICT)*, Dhaka, Bangladesh, 2018, pp. 1-5, doi: 10.1109/ICEEICT.2018.8647872.
- [21] T. K. V. N. Ravi Kumar, B. Y. Lam, and T. H. Le, "Intelligent Tutoring System for Programming Languages: A Review," in *2019 3rd International Conference on Advanced Information Technologies (ICAIT)*, Hammamet, Tunisia, 2019, pp. 263-267, doi: 10.1109/ICAIT.2019.8837876.
- [22] Brusilovsky, P., & Yudelson, M. V. (2013). Gamification and user-centered design for educational applications. In *Proceedings of the 22nd international conference on World Wide Web* (pp. 1011-1014). ACM.
- [23] Titterton, N. (2012). Team-based assessment in a first-year programming unit. In *Proceedings of the Ninth Australasian Conference on Computing Education-Volume 123* (pp. 131-136). Australian Computer Society, Inc.
- [24] M. H. Razzak, S. Sarker and N. B. Anwar, "Peer Review in Programming Education: An Overview," *2020 International Conference on Electronics, Information, and Communication (ICEIC)*, Melbourne, Australia, 2020, pp. 1-4, doi: 10.1109/ICEIC51181.2020.9040203.
- [25] Gómez Martín, M. A. (2019). Uso de software de gestión de concursos de programación para evaluación continua. *Revista de Innovación Educativa*, 18(78), 53-64. <https://doi.org/10.36677/rie.v18i78.10566>
- [26] García-Sánchez, F., Jara-Moreno, J., & Valencia-García, R. (2016). Automated assessment of programming assignments using a template-based approach. *Computers in Human Behavior*, 56, 124-135.
- [27] Cukier, K. (2013). *Web Scraping as a Method of Collecting Data*. Sage Research Methods Cases. DOI: 10.4135/978144628305010538176
- [28] Pham, T., Nguyen, V., & Nguyen, D. (2020). Web scraping for data acquisition in machine learning competitions: A case study of kaggle. In *2020 IEEE 4th Conference on Machine Learning and Data Mining (MLDM)* (pp. 175-182). IEEE.
- [29] Peres, Y. R., Oliveira, D. B., & Ferraz, H. G. (2019). An overview of web scraping for data mining in the context of

social media and big data. In 2019 18th IEEE International Conference on Machine Learning and Applications (ICMLA) (pp. 199-204). IEEE. DOI: 10.1109/ICMLA.2019.00039.

- [30] Fernández-Peña, Y., Mejía, M., & Rangel, C. (2020). Web scraping en la ciencia de datos. Aplicaciones en el análisis de opinión pública. *Ciencia y tecnología de la ciudad*, 7(13), 1-17. <https://doi.org/10.22267/2073818X.20>.