

Niveles de apropiación de los lenguajes de programación en los estudiantes de ingeniería de sistemas de la Universidad Simón Bolívar

Arnol Danilo Garay Angarita
Código estudiantil: 201923219179

Trabajo de Investigación del programa: **Ingeniería en Sistemas**

Tutor:
Miguel Eduardo Posada Haddad

RESUMEN

Antecedentes: En esta investigación se analizaron los niveles de apropiación de los lenguajes de programación en los estudiantes de ingeniería de sistemas de la Universidad Simón Bolívar. La programación es una habilidad fundamental en el campo de la ingeniería de sistemas, y es crucial para el desarrollo de software y la resolución de problemas en el ámbito tecnológico. Por lo tanto, es importante comprender cómo los estudiantes adquieren y aplican estos lenguajes en su formación académica.

Objetivo: El objetivo de este estudio fue analizar los niveles de apropiación de los lenguajes de programación en los estudiantes de ingeniería de sistemas de la Universidad Simón Bolívar sede Cúcuta. Se buscó evaluar el conocimiento y la habilidad de los estudiantes en el uso de diversos lenguajes de programación, así como identificar las fortalezas y debilidades en este aspecto de su formación.

Materiales y Métodos: Para llevar a cabo este estudio, se utilizó una encuesta estructurada basada en la escala de Likert como método de medición. La encuesta incluyó preguntas relacionadas con el conocimiento teórico de los lenguajes de programación, la capacidad de aplicarlos en situaciones reales y la confianza de los estudiantes en su dominio de estos lenguajes. La muestra estuvo compuesta por estudiantes de ingeniería de sistemas de la Universidad Simón Bolívar que se encontraban cursando el séptimo semestre de su carrera. Se realizaron análisis estadísticos descriptivos para examinar los datos recopilados y se utilizaron gráficos y tablas para presentar los resultados de manera clara y concisa. Se tuvieron en cuenta las diferentes dimensiones de la apropiación de los lenguajes de

programación, como la comprensión de la sintaxis, la resolución de problemas y la creatividad en la implementación de soluciones.

Resultados: Una vez recopilados los datos de la encuesta, se realizó un análisis estadístico para identificar patrones y tendencias en los niveles de apropiación de los lenguajes de programación por parte de los estudiantes. Los resultados mostraron que la mayoría de los estudiantes poseían un nivel medio de apropiación de los lenguajes de programación evaluados. En cuanto a los lenguajes de programación más utilizados y mejor dominados por los estudiantes, se encontró que Python, Java y C++ eran los más populares. Estos lenguajes fueron ampliamente utilizados en diferentes cursos y proyectos de programación. Por otro lado, se observó que otros lenguajes menos comunes, como Perl y Ruby, presentaron un menor nivel de apropiación entre los estudiantes. Se encontró que los estudiantes mostraban mayor confianza en su capacidad para comprender la sintaxis de los lenguajes de programación que en su capacidad para aplicarlos de manera efectiva en la resolución de problemas. Esto indica que hay oportunidades de mejora en el desarrollo de habilidades prácticas relacionadas con la programación.

Conclusiones: Los resultados obtenidos en este estudio permitieron comprender mejor la relación de los estudiantes de ingeniería de sistemas de la Universidad Simón Bolívar con los lenguajes de programación y su capacidad para aplicarlos en situaciones reales. Se encontró que los estudiantes tenían un nivel medio de apropiación de los lenguajes de programación evaluados, con algunos estudiantes destacándose por su alto nivel de dominio en estos lenguajes. Es fundamental destacar que la apropiación de los lenguajes de programación no solo implica el conocimiento teórico de la sintaxis y las estructuras de los lenguajes, sino también la capacidad de aplicarlos de manera efectiva en la resolución de problemas y la creación de soluciones innovadoras. Estos hallazgos tienen implicaciones importantes para la formación de los estudiantes de ingeniería de sistemas. Se sugiere fortalecer los programas de estudio en cuanto a la enseñanza y el aprendizaje de los lenguajes de programación, brindando a los estudiantes oportunidades de práctica y desarrollo de proyectos reales que les permitan aplicar y profundizar su conocimiento en estos lenguajes. Se recomienda la implementación de estrategias pedagógicas innovadoras, como el aprendizaje basado en proyectos y el uso de plataformas virtuales de programación, que fomenten la participación activa de los estudiantes y promuevan su capacidad para resolver problemas de manera creativa utilizando los lenguajes de programación. Este estudio proporcionó una visión clara de los niveles de apropiación de los lenguajes de programación en los estudiantes de ingeniería de sistemas de la Universidad Simón Bolívar. Si bien se encontró que los estudiantes tienen un buen dominio de los lenguajes de programación más utilizados, también se identificó la necesidad de fortalecer el conocimiento en otros lenguajes menos comunes. Estos hallazgos pueden servir como base para mejorar los programas de estudio y promover una formación más sólida y actualizada en el campo de la programación.

Palabras clave: Niveles de apropiación, lenguajes de programación, estudiantes de ingeniería de sistemas, mercado laboral, autodidacta.

ABSTRACT

Background: This research aimed to analyze the levels of proficiency in programming languages among students of systems engineering at the Universidad Simón Bolívar. Programming is a fundamental skill in the field of systems engineering and is crucial for software development and problem-solving in the technological sphere. Therefore, it is important to understand how students acquire and apply these languages in their academic training.

Objective: The objective of this study was to analyze the levels of proficiency in programming languages among students of systems engineering at the Universidad Simón Bolívar in Cúcuta. The study sought to evaluate students' knowledge and skill in using various programming languages and identify strengths and weaknesses in this aspect of their education.

Materials and Methods: To conduct this study, a structured survey based on the Likert scale was used as the measurement method. The survey included questions related to theoretical knowledge of programming languages, the ability to apply them in real-life situations, and students' confidence in their mastery of these languages. The sample consisted of systems engineering students at the Universidad Simón Bolívar who were in their seventh semester of study. Descriptive statistical analyses were performed to examine the collected data, and graphs and tables were used to present the results in a clear and concise manner. Different dimensions of language proficiency were taken into account, such as understanding syntax, problem-solving, and creativity in implementing solutions.

Results: Once the survey data were collected, a statistical analysis was conducted to identify patterns and trends in the levels of proficiency in programming languages among students. The results showed that the majority of students had a moderate level of proficiency in the evaluated programming languages. Regarding the most widely used and best mastered programming languages, Python, Java, and C++ were found to be the most popular. These languages were extensively used in various programming courses and projects. On the other hand, less common languages such as Perl and Ruby were observed to have a lower level of proficiency among students. It was found that students displayed greater confidence in their ability to understand the syntax of programming languages than in their ability to effectively apply them in problem-solving. This indicates that there are opportunities for improvement in developing practical programming skills.

Conclusions: The results obtained in this study provided a better understanding of the relationship between systems engineering students at the Universidad Simón Bolívar and programming languages, as well as their ability to apply them in real-life

situations. It was found that students had a moderate level of proficiency in the evaluated programming languages, with some students standing out for their high level of mastery in these languages. It is essential to highlight that language proficiency not only involves theoretical knowledge of syntax and language structures but also the ability to apply them effectively in problem-solving and creating innovative solutions.

These findings have important implications for the education of systems engineering students. It is suggested to strengthen curriculum programs regarding the teaching and learning of programming languages by providing students with opportunities for practice and development of real projects that allow them to apply and deepen their knowledge in these languages. The implementation of innovative pedagogical strategies, such as project-based learning and the use of virtual programming platforms, is recommended to foster active student participation and promote their ability to creatively solve problems using programming languages.

This study provided a clear insight into the levels of proficiency in programming languages among systems engineering students at the Universidad Simón Bolívar. While it was found that students have a good command of the most widely used programming languages, the need to enhance knowledge in less common languages was identified. These findings can serve as a basis for improving curriculum programs and promoting a more solid and updated education in the field of programming.

Keywords: Levels of appropriation, Programming languages, Systems engineering students, Job market, Self-taught.

REFERENCIAS

1. Buriticá, O. I. T. (2019). Episteme vs. Tekné: Una aproximación al aprendizaje de la programación de computadores en Ingeniería de Sistemas. *Revista Educación en Ingeniería*, 14(28), 39-44
2. Cardador Cabello, A. L. (2022). Programación con lenguajes de guion en páginas web. IFCD0110. 1. IC Editorial. <https://ezproxy.unisimon.edu.co:2258/es/lc/unisimon/titulos/221032>
3. Cristancho, F. (2022, 6 abril). ¿Cuáles son las ramas de la programación? - Talently. Talently Blog. <https://talently.tech/blog/ramas-de-la-programacion/>

4. Dussan Rojas, J. C., & Florez, C. A. (2018). Diseño de una estrategia didáctica para el aprendizaje del nivel A1 de inglés orientado por el Instituto de Programación y Sistemas del Norte (INPROSISTEMAS) de la ciudad de Cúcuta.
5. González Barrera, A. M., & Ayala Llanes, C. J. (2020). TRAINING CENTER HIGH SCHOOL. Encuentro Internacional De Educación En Ingeniería. <https://doi.org/10.26507/ponencia.828>
6. Juganaru Mathieu, M. (2015). Introducción a la programación.. Grupo Editorial Patria. <https://ezproxy.unisimon.edu.co:2258/es/lc/unisimon/titulos/39449>
7. Martín Villalba, C. Urquía Moraleda, A. & Rubio González, M. Á. (2021). Lenguajes de programación.. UNED - Universidad Nacional de Educación a Distancia. <https://ezproxy.unisimon.edu.co:2258/es/lc/unisimon/titulos/184827>
8. Martínez, A. A. D., & Reyes, Y. M. (2018). EVALUACIÓN DE COMPETENCIAS DE PROGRAMACIÓN EN LA FORMACIÓN DEL INGENIERO EN SISTEMAS COMPUTACIONALES. ANFEI Digital, (8).
9. Miranda, B. A., & Lenoyr, M. (2022). Indicadores en la enseñanza-aprendizaje de los lenguajes de programación a nivel universitario. Revista Dialogus, 1(4), 11-20.
10. PYPL popularity of Programming Language index (no date) index. Available at: <https://pypl.github.io/PYPL.html> (Accessed: 23 August 2022).
11. Quintero Córdoba, W. (2016). Estrategias de enseñanza de los lenguajes de programación de software a nivel técnico en grado décimo.
12. Ramírez, E. R., Cordero, J. N., Posada, R. C., & Posada, G. E. C. (2018). Enseñanza de la programación: la importancia de promover actitudes autodidactas en los estudiantes. Atenas, 4(44), 46-59.
13. Robert W. Sebesta. (2016). Concepts of Programming Languages. Edinburgh Gate, Harlow, Essex CM20 2JE, England: Pearson Education Limited.

14. Rodríguez, H. E. D. (2017). Tecnologías de la información y comunicación y crecimiento económico. *Economía Informa*, 405, 30-45.
15. Romero Romero, H. J., Cano Olivera, L. E., Charry Mora, C. G., & Pardo Morales, J. R. (2019). Deficiencia de adquisición de competencias mínimas en estudiantes de desarrollo de software:: hacia un nuevo modelo de enseñanza pedagógico praxeológico. *IyD*, 6(2), 85–97. <https://doi.org/10.26620/uniminuto.inclusion.6.2.2019.85-97>
16. Romero, S., & Quintero, J. (2014). Nivel de conocimiento en programación y el desarrollo de sistemas de información automatizados. *TECHNO REVIEW. International Technology, Science and Society Review*, 3(2).
17. Rosa de Rivera, V. I. (2021). Diseño causal de sistema piramidal para aprendizaje colaborativo de los lenguajes de programación.
18. Ruiz, J. G., López, M. H., & Brito, J. A. L. (2015). Pensamiento sistémico y desarrollo de competencias, en el aprendizaje de los lenguajes de programación. *ANFEI Digital*, (2).
19. Sánchez, B. A. N., & Pihuave, S. T. G. (2021). Empleabilidad de los graduados de la carrera de ingeniería de sistemas de la universidad politécnica salesiana de Guayaquil, del año 2017 al 2019. *Revista Boletín Redipe*, 10(10), 367-379.
20. Stack overflow developer survey 2021 (no date) Stack Overflow. Available at: <https://insights.stackoverflow.com/survey/2021> (Accessed: 23 August 2022).
21. The state of developer ecosystem in 2021 infographic (2021) JetBrains. Available at: <https://www.jetbrains.com/lp/devecosystem-2021/> (Accessed: 23 agosto 2022).
22. Trejos Buriticá, O. I. (2014). Relaciones de aprendizaje significativo entre dos paradigmas