Exploring Non-CS Learners' Experience in Brazil

Renata Faria Gomes[1], Ricardo Ferreira Dos Santos Junior, Vinícius Cardoso Garcia[1]

[1]Centro de Informatica – Universidade Federal de Pernambuco – Recife – PE – Brazil

rfg4@cin.ufpe.br, rfdsjr01@gmail.com, vcg@cin.ufpe.br

Abstract. *Non-CS learners are increasingly interested in programming for career opportunities, productivity, and better communication with technical teams. Despite well-known challenges like logical thinking, unique issues for non-CS learners remain underexplored.* ***Purpose:*** *This study examines the motivations, experiences, and challenges of Brazilian non-CS learners, aiming to inform educational strategies.* ***Methodology:*** *We conducted an open-ended survey with a diverse sample of 36 considered respondents, analyzing data through open coding.* ***Findings:*** *Key motivations include career advancement and job performance, with a preference for self-learning. However, non-CS learners encounter challenges with knowledge organization, discipline, and technical comprehension. his highlights the need for adaptable and personalized educational solutions that effectively support learners from diverse backgrounds.* ***Recommendations:*** *Students can leverage their original field's expertise when transitioning careers or selecting relevant technologies enhancing future roles. Companies should support diverse backgrounds and provide tech training to non-technical staff to improve communication and productivity. Academic institutions may provide optional, field-integrated CS courses to prepare students with practical, applicable skills.*

1. Introduction

Rapid technological advancements, accelerated by the COVID-19 pandemic, have transformed how we live and work, increasing demand for computer-skilled professionals in a growing IT industry [Madden 2022]. Computer Science (CS) spans various fields, offering skills valuable to both technical and non-technical professionals. The recognition of computer science as a K-12 (middle and high school in Brazil) subject underscores its relevance beyond IT-related fields [Tucker 2003]. For instance, Computational thinking enhances problem-solving and logical skills across disciplines, with benefits extending beyond computer science [Vicari et al. 2018]. In this context, the concept of "conversational programmers" has been described in the literature as individuals who learn programming to improve collaboration with technical peers [Wang et al. 2018, Chilana et al. 2016, Cunningham et al. 2021]. Similar research has shown how programming education helps social science students become "technology-capable" by applying computational tools in their fields [Carr et al. 2020a].

Despite the growing interest in programming among non-CS learners, many face challenges like unclear objectives [Carr et al. 2020a], knowledge organization difficulties, and a lack of tailored materials [Mironova et al. 2016]. These issues remain underexplored in the Brazilian context, particularly regarding how learners structure their journeys and choose technologies. To bridge this gap and inform more effective educational strategies, this study aims to answer the following research questions:

RQ 1 **What are the reasons behind non-CS learners' interest in programming, and which technologies do they tend to focus on?** – Understanding their motivations can help educators and companies create more relevant and engaging learning experiences.

RQ 2 **What is the most commonly adopted learning method, and what difficulties does it involve?** – Identifying common learning methods and their gaps can guide improvements in resources and in the approach learners already use.

RQ 3 **How can computer science contribute to various fields and in what ways?** – Highlighting the role of computing knowledge in enhancing professional practices and promoting interdisciplinary collaboration

To address the research questions, we conducted an open-ended survey to effectively capture the learners' perspectives, which will be presented in the following sections.

2. Background and Related Works

This section provides a background foundation and explores related works to contextualize our discussion. We used the search string presented in Table 1 to identify related studies[1]. The identified works were then categorized into two groups: **applied education** (Table 2) and **surveys on education** (Table 3).

2.1. Challenges in Computer Science education

Learning CS subjects is challenging, with high failure rates even among CS students [Bennedsen and Caspersen 2007]. Common obstacles include logical-mathematical reasoning, syntax comprehension, and troubleshooting, along with external factors that can further impact learning [da Silva et al. 2021, Aureliano et al. 2016]. Technical-only activities tend to be less engaging and can hinder performance compared to those involving human interaction [Christensen et al. 2021]. Research also shows that self-confidence is a stronger predictor of success in programming than mathematical background [Rountree et al. 2004]. Additionally, diversity-related challenges, such as gender stereotypes, can reduce women's self-confidence [Singh et al. 2020], a pattern also observed in LGBT groups [Stout and Wright 2016].

2.2. The 'Non-Tech' learners

This study examines non-technical learners, or "non-CS", including "conversational programmers" who learn programming to communicate more effectively with technical peers [Wang et al. 2018]. Learning resources often fail to meet their goals, especially when teaching abstract concepts. Is known visual teaching methods and graphical strategies can better engage intuitive learners, such as artists, by bridging the gap between systematic and intuitive thinking [Tsai et al. 2006]. Additionally, Computational Thinking (CT) has been recognized as a fundamental skill across various domains, involving techniques like decomposition, pattern recognition, abstraction, and algorithmic thinking [Wing 2006]. These skills enhance logical reasoning, critical thinking, and decision-making, benefiting even non-CS learners as they apply computing knowledge in their fields [Vicari et al. 2018].

[1] For more details, visit: `https://zenodo.org/records/14841562`

(”Non-IT” OR ”Conversational programmers” OR ”Non-tech” OR ”Non-STEM” OR ”non-programmer”) AND (”learn” OR ”learning” OR ”education”) AND (”programming” OR ”Computing” OR ”software engineering”)

Table 1. Search String

The studies in Table 2 provide a broader view of how non-IT learners engage with programming, highlighting challenges like self-learning difficulties and the role of structured education. While most findings come from international contexts, they offer insights relevant to Brazil, which faces unique challenges such as limited access to resources, regional disparities, and language barriers. This study builds on these findings by analyzing the experiences of Brazilian non-CS learners, comparing global trends with local realities to uncover context-specific challenges.

Work	Topic	Learner Profile	Objective	Course Format	Cited Problems
[Menkhoff and Lydia Teo 2022]	Chatbot (DialogFlow), NLP	Management and Finance undergraduate Students	Provide skills to business students e.g.: Customer Service	One-day workshop	Not cited
[Pena et al. 2021]	HTML, CSS, JS, Gitlab and wordpress	Professional Women from multiple backgrounds	Students had curiosity to explore the technology	Nine-week workshops	Not cited
[Carr et al. 2020b]	Python, R and Project Management	Students from multiple backgrounds, but mostly Psychology and Economics	Improve students' data analysis skills	Interdisciplinary computing minor degree (3 courses and a project)	Pacing - Sometimes fast, sometimes slow
[Campos et al. 2021]	Matlab for Datascience	Non-IT second-year engineering students	Provide ”Digital Transformation” skills to students	60h course divided in 35h practice and 25h theory	Not cited
[Spangsberg and Brynskov 2017]	Java, HTML/CSS, MySQL and basic HCI	Master's degree students of Information Studies	Student's perception of ”what software is, how it can be utilised and how it develops”	Three sessions of 5h per week; during a month	Order of theory and practice content
[Bart et al. 2017]	Python	Non-CS majors from the humanities, arts, and the sciences.	Ability to solve problems from a data-oriented computational perspective	6 weeks of introductory classes and 2 weeks practice	Semantic Errors in code
[Xu et al. 2021]	Python	Digital media technology undergraduate students	Reinforce the students Computational Thinking (improve their problem solving skills)	13-week course	Concept of objects and lists/ dictionaries and functions.

Table 2. Applied education for Non-it

2.3. Studies related to surveys on Non-IT education

Understanding learners' profiles is an important aspect of tailoring educational approaches. While some studies focus on these profiles, they often overlook the challenges learners face, as seen in [Menkhoff and Lydia Teo 2022]. In contrast, other research tends to apply the same content to all participants, regardless of their backgrounds. However, the objectives in [Carr et al. 2020b, Spangsberg and Brynskov 2017, Bart et al. 2017] demonstrate better alignment with the learning topics. The challenges identified in these works can be grouped into two main categories: methodological issues, such as pacing and the balance between theory and practice, and learning difficulties, including semantic errors and conceptual misunderstandings.

The studies in Table 3 provide additional insights into how non-IT learners engage with programming, highlighting diverse approaches and challenges. For instance, [Kakeshita 2017] conducted a survey with non-IT students across Japanese universities, where computing is integrated into various disciplines despite limited specialized staff. The study recommended balancing practical and theoretical instruction to improve learning outcomes. Our research extends this perspective by including professionals beyond the university context.

Another study [Guo 2017] surveyed adult learners globally through the Python Tutor platform, which has over 2.5 million users from 180 countries. The survey targeted learners who accessed the site from various sources, including MOOCs, online courses, and coding forums. It identified motivations like mental challenges, family connection, and career growth, along with difficulties related to poor pedagogy, cognitive decline, and limited interaction. Suggested solutions included contextualizing lessons and bridging technical gaps. Our study contributes to this discussion by focusing on Brazilian learners, adding a distinct demographic perspective.

The last study [Chilana et al. 2015] examined first-year management engineering students at a North American university using surveys, interviews, and classroom observations. Two surveys were administered: one at the start and another near the end of the course, with response rates of 69% and 75%, respectively. Additionally, 25 semi-structured interviews and weekly classroom observations provided deeper insights into challenges like syntax comprehension and low programming confidence. While these findings reflect the experiences of younger students, they align with our observations regarding the value of conversational programming. By combining these findings with our results, we provide a broader understanding of non-CS learners' challenges and motivations, offering a different perspective of the studies listed in Table 3.

Study	Method	Author Objective	Profile of Respondents
[Kakeshita 2017]	Survey quantitative;	Analyze the state of computing education at non-IT departments in Japanese universities	Non-IT departments and faculties at Japanese universities
[Guo 2017]	Online survey qualitative	Understand why, how, and frustrations of older adults learning computer programming	Older adults aged 60-85 from 52 countries
[Chilana et al. 2015]	Mixed-methods (surveys, interviews, observations)	Explore perceptions, motivations, and learning experiences of non-CS majors in an intro programming course	First-year management engineering students at a large North American university

Table 3. Studies related to surveys on Non-IT education

3. Methodology

Before defining the methods, we considered how our limitations would impact achieving the objectives. Both researchers had limited time due to other commitments, restricting the number of participants. Additionally, the exploratory nature of our aims meant an open-ended survey could introduce bias. Thus, we opted for an online survey with open-ended questions, supplemented by a sampling strategy detailed later.

3.1. Survey design

We developed a survey with three sections: research details and ethical agreement, demographic questions, and an experience report. The full questionnaire is available online[2]. The first section explained the research, including data processing and anonymity, requiring participant consent. The second section contained demographic questions to assess sample representativeness. The final section focused on participants' backgrounds and experiences with CS-related topics, including filtering questions based on our sampling

[2] `https://zenodo.org/records/14841562`

strategy, detailed in the next subsection. The participants were invited spontaneously and online, as we disseminated our research through institutional emails, professional social networks, academic groups on social media, and messaging app groups. The survey was conducted in Portuguese, hence the respondents were exclusively Brazilian.

3.2. Sampling strategy and criteria

Our work aimed to provide insights into the Brazilian context regarding the experiences, objectives, and challenges of non-IT learners. To achieve this, we sought a sample as representative of the Brazilian education context as possible, using datasets from INEP (National Institute of Educational Studies) and IBGE (Brazilian Institute of Geography and Statistics)[3]. Although these datasets do not perfectly match the target population, they served as a reference for representativeness. The results are not intended for generalization but as exploratory insights.

The sampling strategy used two steps to achieve saturation: specific survey questions and respondent monitoring. To achieve sampling saturation, we used specific survey questions and respondent monitoring. To ensure sampling saturation, we conducted weekly assessments to identify missing attributes, targeting regions or groups that were underrepresented, such as university groups from less-represented regions. The sampling criteria considered the education level (at least incomplete undergraduate) and field, excluding all CS-related courses. Additionally, respondents were required to indicate their current location (Brazilian states), with individuals located outside Brazil being excluded.

3.3. Data analysis

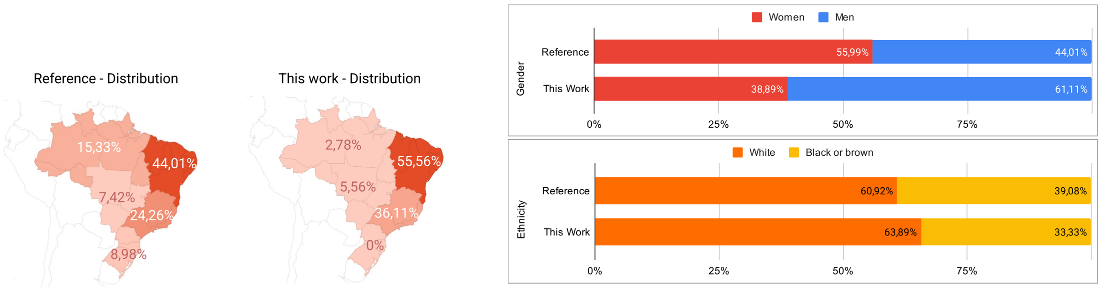
The data analysis was split into two processes: one for the demographics (quantitative data) and one for the experience reports (qualitative data). Prior to data collection, the reference data was processed for analysis and comparison. The analysis of quantitative data for representativeness was conducted alongside data collection. After closing the questionnaire collection, we began the second phase: qualitative data analysis using the Axial Coding method from [Merriam and Tisdell 2015] Collected data and codebook are available online[4]. We started with open coding, reviewing each response and summarizing it in brief comments, which were then grouped into categories, often naturally. We prioritized keeping each survey question separate in the analysis, except when thematic connections emerged, like motivation, as shown in section 4.

4. Results

The questionnaire collected responses from February to April 2023, yielding 54 total responses, of which 36 (66.67%) were valid. We excluded participants with current or prior enrollment in CS-related courses. Achieving regional representativeness proved challenging, despite efforts to reach a diverse geographic sample; thus, this data serves as a reference rather than a generalizable set. A notable difference was in gender distribution:

[3]Links: INEP Higher Education: `https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/indicadores-educacionais/indicadores-de-fluxo-da-educacao-superior`; IBGE Social Indicators: `https://www.ibge.gov.br/estatisticas/sociais/educacao/9221-sintese-de-indicadores-sociais.html`; IBGE Census: `https://censo2022.ibge.gov.br/panorama/indicadores.html`; all links accessed May 18th, 2024

[4]`https://zenodo.org/records/14841562`



Category		This work		Reference		Difference
Brazillian Region	Northeast / Nordeste	20	55,56%	745.596,00	44,01%	11,54%
	North / Norte	1	2,78%	259.665,00	15,33%	-12,55%
	Centro-Oeste	2	5,56%	125.692,00	7,42%	-1,86%
	Southeast / Sudeste	13	36,11%	410.924,00	24,26%	11,85%
	South / Sul	0	0,00%	152.094,00	8,98%	-8,98%
Gender	Women	14	38,89%	26.417.414	55,99%	-17,10%
	Men	22	61,11%	20.762.208	44,01%	17,10%
Ethnicity	White	23	63,89%	28.148.619,45	60,92%	2,97%
	Black or brown	12	33,33%	18.060.989,98	39,08%	-5,75%
	Others	1	2,78%	not available	not available	not available

Figure 1. Representativeness comparison

women represented 56% in the dataset but less than 40% of survey respondents, possibly due to challenges faced by women in CS, as discussed in subsection2.1. Conversely, the respondents' ethnicity profiles aligned between the dataset and our survey. Figure 1 provides a visual comparison between our survey and the reference data. We observed diverse profiles when considering participants' prior experience, objectives, learning methods, challenges, and other factors. Therefore, we will examine these profiles in more detail in the following subsections.

While we initially sought to explore correlations between demographic attributes (e.g., gender, region, race) and learning experiences, no significant patterns emerged, likely due to the small sample size (N=36). However, we observed motivational patterns: career-transitioners favored self-learning through online courses, while those enhancing current job skills often opted for structured, in-person courses. Differences also appeared in how various professional backgrounds applied programming knowledge, aligning with previous findings. Future research with a larger, more diverse sample could better assess demographic influences on learning strategies and challenges.

4.1. Transition career respondents

The first group (n=22 – 61%) consisted of respondents who mentioned career change either in their motivation or in their intended application of knowledge. Although they shared this characteristic, there was significant diversity within the group. By grouping them, however, we can uncover common insights. For analysis, we considered all answers provided, not just the motivation question (for more details, please check our code-book). For example, while one participant listed 'to improve my career' as motivation, their other responses clarified a goal to shift careers. Additionally, we explored the reasons for this career transition. Some respondents cited difficulty in finding well-paid jobs in their current fields; for instance, one participant noted:

> *"...CS-related jobs offer a better cost-benefit when comparing the time invested with the financial return, compared to other classical careers"*

Other reasons included opportunities for relocation and influences from family and friends. A interesting insight, not previously reported in the literature, was how their original field complemented the learning in CS:

> *"I would like to work in the Salesforce platform, utilizing my commercial and administrative skills to aid me in this new profession."*

Learners' technology choices reflected the resources and learning methods that suited their immediate goals. One respondent, for example, began learning Python independently but shifted to mobile development following an internship opportunity. YouTube was the most frequently used resource, followed by online courses and in-person classes. Python was popular among YouTube users, while course participants explored a broader range of languages. Experienced learners relied more on documentation and often engaged in practical activities, such as internships.

Challenges included those noted in the literature, such as developing a logical mindset, understanding technical terms, underrepresentation of women, and reliance on English. Unique to this study, however, were career transition challenges, including stepping back from work to focus on studies and frustration during the learning process. Apart from the previously mentioned challenges, Students who used YouTube or online courses also reported issues with daily study discipline. Some faced equipment limitations, preventing them from running updated software. Many online learners struggled with finding the correct sequence of content or lacked foundational knowledge. For example, one student said:

> *"I miss the introductory content and the programming principles."*

Some respondents also mentioned challenges with onsite courses, including issues with course structure and exhaustion due to course intensity. Many struggled to balance studies with work, which could impact their mental health.

> *"I struggled with the exhaustion of completing a 6-month intensive course, leaving me with no free time for rest during the week."*

4.2. Respondents aiming to learn skills for their current career

The second group (n=12 – 33%) included individuals aiming to advance their careers through programming. Some were not initially interested but started learning due to internships or curriculum requirements. Many saw programming as a way to automate and improve daily tasks. Post-graduate students often learned it for research, especially data analysis, enhancing their routines.

> *"Programming has given me autonomy and a critical view of the data generated."*

Others cited advantages when collaborating with computer science colleagues, similar to the "Conversational Programmers" described in the literature.

> *"Having programming skills allows me to communicate more effectively with our development team and understand the technical aspects of our projects."*

In contrast to the career transition group, this set preferred online courses, followed by in-person instruction, with some also using YouTube videos and a notable interest in books among post-graduates. Python was overwhelmingly cited, with fewer mentions of other languages. Challenges included the need for a more analytical mindset and maintaining discipline, similar to the other group, but without the added pressure of career transition. English and finding quality content was also cited.

4.3. Other respondents

Only two respondents fell into this category (6%). One was learning programming for personal enjoyment via YouTube, while the other was fulfilling a graduation requirement in a face-to-face class. Both were learning Python and faced challenges with developing a logical mindset, but neither planned to use the skills in the future.

> *"I am learning programming because I enjoy solving puzzles and want to understand how software works."*

4.4. Computing and society

Respondents were asked about the connection between programming and their original field. Some related CS to systems they used in their field, while others saw their main area influencing computer science.

> *"Psychology and computer science have some common points, such as human-machine interaction, the psychology of colors in an interface, and artificial intelligence."*

Those with the strongest connections often linked IT to daily tasks, like Designers or Project Managers. Despite differing views, most would recommend programming within their field, while others saw it as a career path. A few did not recommend it.

> *"I think not everyone is interested in programming, and that's okay. Learning it takes a lot of time, and a few hours of classes won't provide practical benefits."*

When asked about the importance of programming for society, responses fell into two groups: those who considered it universally beneficial and those who believed it was only useful within the field. Curiously, respondents who recommended programming saw it as valuable for society, while those who did not felt it offered limited benefits.

5. Discussion

The data revealed both shared and differing challenges between CS and non-CS learners, such as similar issues with logical thinking but distinct challenges in career transitioning. We will explore these further through our research questions.

5.1. What are the reasons behind non-CS learners' interest in programming, and which technologies do they tend to focus on?

The analysis shows that initial motivations for learning programming often shifted over time. Many began due to work or course requirements, later adapting these motivations for career transitions or skill enhancement. Python was widely cited, especially by those using platforms like YouTube, though no distinct technology preferences emerged. Most learners sought better job opportunities or higher salaries in IT, while others aimed to boost productivity by automating tasks. Some wanted to improve collaboration with technical teams, with a few driven by personal interest or course needs.

5.2. What is the most commonly adopted learning method, and what difficulties does it involve?

Respondents primarily preferred self-learning via online platforms. Career changers leaned toward YouTube and online courses, while post-graduates also used books. Some adapted their methods to address challenges in logical thinking, technical concepts, and terminology. Career transitioners encountered additional obstacles, including balancing learning with work, which impacted their well-being. We recommend further research on the specific pressures faced by career transitioners in this context.

5.3. How can computer science contribute to various fields and in what ways?

Many respondents pursued programming to shift careers, showing computer science's potential to "open doors" across fields. Technology enables non-CS professionals to repurpose skills and innovate beyond traditional CS paths. Key applications included automating tasks, enhancing productivity, and improving data analysis and problem-solving. Our recommendations will explore how companies and job candidates can better leverage these abilities.

5.4. Current literature and findings

It is important to begin the discussion by acknowledging this study's limitations due to the small sample size and the absence of a systematic literature review. Despite this, we highlight key points from the literature tables: Table 2 focuses on teaching methods for non-tech learners, while Table 3 examines students' experiences. Most studies emphasized face-to-face learning, except [Kakeshita 2017], which noted a preference for online courses—a trend also seen in our survey with a preference for self-learning. Career change motivations were underexplored in previous research but were prominent in our results, suggesting the need for further investigation. Other common motivations included learning data analysis [Campos et al. 2021], improving problem-solving [Bart et al. 2017], and fostering conversational programmers [Chilana et al. 2015]. Shared challenges involved balancing theory and practice [Spangsberg and Brynskov 2017] and grasping technical concepts [Bart et al. 2017]. A distinct challenge in Brazil was the language barrier, as most online resources are in English.

6. Threats to Validity

Due to limitations, the method couldn't capture a fully representative sample of non-CS learners in Brazil. There are biases to acknowledge, but our goal **was not to generalize but to explore the Brazilian context**. Recognizing potential biases and limitations is essential for the robustness of our findings. We categorize threats to validity into external, internal, construct, and conclusion validity

External Validity - Our online survey introduces selection bias, as participants were self-selected and likely had some familiarity with programming. Additionally, regional imbalance (overrepresentation of northeastern Brazil) limits generalization. **Internal Validity** Our study identifies patterns among non-CS learners without claiming causality. Self-reported data may introduce bias due to social desirability or memory inaccuracies. To mitigate this, we ensured anonymity and used open-ended questions to encourage honest reflections. The study also lacks a longitudinal perspective, preventing analysis of changes over time.

Construct Validity - We excluded CS-educated participants to ensure relevance. However, our study did not employ pre-validated instruments for assessing learning challenges and motivations, instead relying on open-ended qualitative responses and open coding analysis. While this approach provided rich insights, incorporating standardized assessment tools in future research could improve the reliability and comparability of the findings. **Conclusion Validity** - While our qualitative analysis yielded valuable insights, the sample size (N=36) is relatively small, which limits the statistical power of our conclusions. Additionally, the open coding process used for qualitative analysis carries a degree of subjectivity, as different researchers may interpret responses differently. To mitigate this, we conducted iterative coding reviews and cross-validation among researchers. Future studies should explore techniques such as inter-rater reliability assessment and triangulation with quantitative data to enhance the robustness of conclusions.

7. Conclusion

This study explored the experiences, motivations, and challenges of non-CS learners in Brazil pursuing programming skills. Through an open-ended survey, we gathered qualitative insights that highlight key findings for improving educational strategies. Based on the results, we offer recommendations for students, companies, and academic institutions.

For students: Leveraging their field-specific expertise can help career changers secure jobs that combine technical and domain knowledge, bridging the gap between tech and business. For those not aiming to transition, connecting learning to relevant technologies ensures practical application in daily tasks, adding professional value. **For companies**: Recruiting professionals with diverse backgrounds can improve communication between technical and business teams. Supporting internal talent considering a tech transition can foster innovation, while offering tech training to non-technical staff, such as in data analysis, increases efficiency and autonomy. **For academic institutions**: Professors could adopt an overview approach in mandatory courses, introducing foundational skills such as logical and computational thinking, while reserving practical, in-depth activities for optional courses. Integrating CS content with field-specific applications, such as automation for drafting petitions, can better engage students with diverse goals.

In summary, this study highlights the varied goals of non-CS learners, from career transitions to skill enhancement. It underscores the importance of flexible, context-aware educational approaches to better integrate computer science into diverse fields, fostering more inclusive and effective learning experiences.

8. Future Works

Future research should expand the sample size for better demographic analysis, conduct longitudinal studies to track evolving learning behaviors, and explore the unique challenges faced by career changers. Investigating the impact of language barriers, the effectiveness of self-learning methods, and the potential of personalized educational resources could enhance learning outcomes. Additionally, studying the interdisciplinary applications of CS skills and the relationship between mental health and learning performance may offer valuable insights for developing more effective, inclusive, and adaptable educational practices.

References

Aureliano, V. C. O., Tedesco, P. C. d. A. R., and Giraffa, L. M. M. (2016). Desafios e oportunidades aos processos de ensino e de aprendizagem de programação para iniciantes. In *Anais do XXIV Workshop sobre Educação em Computação*, pages 2066–2075. SBC.

Bart, A. C., Tibau, J., Kafura, D., Shaffer, C. A., and Tilevich, E. (2017). Design and evaluation of a block-based environment with a data science context. *IEEE Transactions on Emerging Topics in Computing*, 8(1):182–192.

Bennedsen, J. and Caspersen, M. E. (2007). Failure rates in introductory programming. *AcM SIGcSE Bulletin*, 39(2):32–36.

Campos, J. M., Lozano, E. A., Urzúa, J., and Calderón, J. G. (2021). Challenge based learning: A fast track to introduce engineering students to data science. In *2021 Machine Learning-Driven Digital Technologies for Educational Innovation Workshop*, pages 1–6. IEEE.

Carr, V., Jones, M., and Wei, B. (2020a). Interdisciplinary computing: Applied computing for behavioral and social sciences. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 400–406.

Carr, V., Jones, M., and Wei, B. (2020b). Interdisciplinary computing: Applied computing for behavioral and social sciences. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, pages 400–406.

Chilana, P. K., Alcock, C., Dembla, S., Ho, A., Hurst, A., Armstrong, B., and Guo, P. J. (2015). Perceptions of non-cs majors in intro programming: The rise of the conversational programmer. In *2015 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 251–259. IEEE.

Chilana, P. K., Singh, R., and Guo, P. J. (2016). Understanding conversational programmers: A perspective from the software industry. In *Proceedings of the 2016 CHI conference on human factors in computing systems*, pages 1462–1472.

Christensen, I. M., Marcher, M. H., Grabarczyk, P., Graversen, T., and Brabrand, C. (2021). Computing educational activities involving people rather than things appeal more to women (recruitment perspective). In *Proceedings of the 17th ACM Conference on International Computing Education Research*, ICER 2021, page 127–144, New York, NY, USA. Association for Computing Machinery.

Cunningham, K., Ericson, B. J., Agrawal Bejarano, R., and Guzdial, M. (2021). Avoiding the turing tarpit: Learning conversational programming by starting from code's purpose. In *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*, pages 1–15.

da Silva, T. R., Barros, I. S., Sousa, L. K. D. S., Sá, A. L. D., Silva, A. F. M., Araujo, M. C. S., and da Silva Aranha, E. H. (2021). Um mapeamento sistemático sobre o ensino e aprendizagem de programação. *Revista Novas Tecnologias na Educação*, 19(1):156–165.

Guo, P. J. (2017). Older adults learning computer programming: Motivations, frustrations, and design opportunities. In *Proceedings of the 2017 chi conference on human factors in computing systems*, pages 7070–7083.

Kakeshita, T. (2017). National survey of japanese universities on computing education: Analysis of non-it departments and courses. In *2017 Twelfth International Conference on Digital Information Management (ICDIM)*, pages 86–91. IEEE.

Madden, T. (2022). Why now is a good time for a career change to tech. https://www.forbes.com/sites/forbescoachescouncil/2022/03/14/why-now-is-a-good-time-for-a-career-change-to-tech/?sh=20a31e3b5296 [Accessed: 2024-04-12].

Menkhoff, T. and Lydia Teo, Y. Q. (2022). Engaging undergraduate students in an introductory ai course through a knowledge-based chatbot workshop. In *Proceedings of the 6th International Conference on Information System and Data Mining*, pages 119–125.

Merriam, S. B. and Tisdell, E. J. (2015). *Qualitative research: A guide to design and implementation*. John Wiley & Sons.

Mironova, O., Amitan, I., Vendelin, J., Vilipõld, J., and Saar, M. (2016). Teaching programming basics for first year non-it students. In *2016 IEEE Global Engineering Education Conference (EDUCON)*, pages 15–19. IEEE.

Pena, J., Hanrahan, B. V., Rosson, M. B., and Cole, C. (2021). After-hours learning: Workshops for professional women to learn web development. *ACM Transactions on Computing Education (TOCE)*, 21(2):1–31.

Rountree, N., Rountree, J., Robins, A., and Hannah, R. (2004). Interacting factors that predict success and failure in a cs1 course. *ACM SIGCSE Bulletin*, 36(4):101–104.

Singh, V. K., Chayko, M., Inamdar, R., and Floegel, D. (2020). Female librarians and male computer programmers? gender bias in occupational images on digital media platforms. *Journal of the Association for Information Science and Technology*, 71(11):1281–1294.

Spangsberg, T. H. and Brynskov, M. (2017). Code-labelling: A teaching activity encouraging deep learning in a non-stem introductory programming course. In *2017 12th International Conference on Computer Science and Education (ICCSE)*, pages 95–100. IEEE.

Stout, J. G. and Wright, H. M. (2016). Lesbian, gay, bisexual, transgender, and queer students' sense of belonging in computing: An intersectional approach. *Computing in Science & Engineering*, 18(3):24–30.

Tsai, M.-H., Huang, C.-H., and Zeng, J.-Y. (2006). Game programming courses for non programmers. In *Proceedings of the 2006 international Conference on Game Research and Development*, pages 219–223.

Tucker, A. (2003). A model curriculum for k–12 computer science: Final report of the acm k–12 task force curriculum committee. Technical report, New York, NY, USA.

Vicari, R. M., Moreira, A. F., and Menezes, P. F. B. (2018). Pensamento computacional: revisão bibliográfica.

Wang, A. Y., Mitts, R., Guo, P. J., and Chilana, P. K. (2018). Mismatch of expectations: How modern learning resources fail conversational programmers. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, pages 1–13.

Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3):33–35.

Xu, Z., Shen, X., Lin, S., and Zhang, F. (2021). Using visualization to teach an introductory programming course with python. In *2021 11th International Conference on Information Technology in Medicine and Education (ITME)*, pages 514–518. IEEE.