A computational system to support learning self-regulation to measure knowledge acquisition by students of Computer Science-related courses

Paula Torales Leite[1] **, Ismar Frango Silveira**[1]

[1]Faculdade de Computação e Informática - Univerisdade Presbiteriana Mackenzie
01302-907 - São Paulo - SP - Brasil

{paula.leite, ismar.silveira}@mackenzie.br

Abstract. *The evaluation of knowledge acquisition is a constant challenge in the educational field, mainly for learners in higher education – in this paper, students of Computer Sciences-related courses are the main target. In this context, self-regulation and metacognition emerge as tools that might help improve this process, allowing students to monitor their learning process. The objective of this project is to evaluate the applicability of a computational system based on self-regulation for the verification of knowledge acquisition in higher education, in the area of technology. Thereby, we aim to contribute to learners' learning process and facilitate teachers in keeping up with this evolution. In order to do this, a literary review was conducted, followed by the development of the computational system. The results of the tests conducted with educators indicated a consensus in regard to the viability of the application of the system, highlighting that the proposal contributes to a more humane evaluation and that it helps in the identification of the learners' needs. In future works, it's viable to take into consideration points found throughout the research, focusing on the betterment of the system with the aim of applying it effectively in educational environments.*

1. Introduction

The pursuit of knowledge is a constant journey fraught with various obstacles that must be overcome. The process of assessing knowledge acquisition emerges as one of the challenges in the educational journey and will be further discussed in this project. Other research has presented several techniques for validating knowledge acquisition, classified as diagnostic, formative and summative [Bloom et al. 1983].

Further more, exams, dissertations, essays, oral exams, interviews, objective tests, observations, practical projects, concept maps, and portfolios can be used to aid in evaluations [Pimentel 2006]. For the most part, they represent traditional approaches to monitoring the evolution of students' knowledge. Also, studies have indicated that the perception of success in learning, as well as the application of written exams, can impact their motivation [Ramos 2019]. This perspective highlights how the choice of assessment instruments is a critical dimension for promoting an educational environment that fosters learners' development and is a core concept that is encompassed in this paper.

As such, further research suggest that evaluation in Computer Science (CS) courses could be multifaceted, involving automated assessment of code, various techniques for assessing programming ability, project-based learning, stakeholder attitudes,

DOI: 10.5753/sbie.2024.242147

continuous grading, curriculum development, and multiple factors including knowledge, motivation, and practical skills, using tools ranging from questionnaires to analytic network processes [Paiva et al. 2022].

However, as can be seen in Ulbricht's [Ulbricht 1982] classic paper, evaluations in the context of Computer Science-related courses are often still based on summative assessment activities, like exams or final papers. Even courses that facilitate the implementation of different forms of formative assessment have difficulties persuading internal bureaucrats at Universities, tending to a more summative-oriented approach because of their essence or tradition. As such, different strategies and instruments related to Active Learning [Hartwig et al. 2019], like Problem/Project-Based Learning or Challenge-Based Learning [Johnson et al. 2009], are left behind.

Within the realm of formative assessments, metacognition stands out as a technique that enables individuals to reflect about their knowledge, providing control over their own learning [Bransford et al. 2003]. Simultaneously, self-regulation, known as an active process, empowers students to monitor, regulate, and control their cognition and motivation [Pintrich 2003], allowing for more autonomous learning where students can prioritize their studies [Veiga 2017]. The application of these different teaching strategies, such as assessment practices and class dynamics, have been found to influence student satisfaction, performance, and absenteeism in Introduction to Computer Science classes [Dias Canedo et al. 2017].

The use of the Dimensions [Dimensions 2023] software revealed significance in the topic of self-regulation. This keyword was used and presented an approximate 64% increase in scientific productions on self-regulation between the years 2018 and 2022, indicating a growing interest in this area compared to the previous five years, which had an increase of about 13%. Furthermore, studies indicate that the use of metacognitive tools resulted in greater learning gains, emphasizing goal setting, monitoring of knowledge acquisition, and reflections [Zheng et al. 2019].

Given this context, this paper presents the evaluation of the applicability of a computational system based on self-regulation in relation to monitoring the cognitive development of a student enrolled in Computer Sciences-related courses. This was made possible with the development of a cellphone application, which can be accessed by the teacher and the learner, providing a well-rounded solution for both.

2. Theoretical Framework

Pimentel [Pimentel 2006], when analyzing a diverse set of studies, four fundamental paradigms were identified: behavioral, cognitivist, humanistic, and social. These paradigms offer distinct perspectives on how the learning process occurs, contributing to the understanding of this dynamic.

In the behavioral paradigm [Dewaele and Li 2021], learning is conceived as the result of planned experiences [Pimentel 2006]. These experiences, in turn, trigger modifications in the individual's behavior. For the execution of these experiences, the development of learning is centered on the role of the teacher, who carefully observes the behaviors and interconnections established by the students throughout the proposed activities. The effectiveness of the feedback provided by the teacher is associated with the application of praise and recognition, guiding students towards the desired behavior. In this approach, repetition is considered a fundamental principle, with students perceived as learning through it, without a prominent consideration of prior knowledge. Consequently, it is the responsibility of the teacher to present the content in a planned manner, aligned with an understanding of the desired result.

This behavioral approach, grounded in planned experience and repetition, highlights the importance of the teacher in the teaching-learning process. A strategy that contributes to this process is the emphasis on positive feedback and continuous monitoring of students, promoting the development of desired behaviors.

On the other hand, the cognitivist paradigm [Clark 2018] emerges from the process by which the student actively engages in the acquisition of new information, self-managed and elaborated by the individual. In this context, learning is expected to occur through trial and error, eliminating the formal limitations associated with traditional education. This approach demands that the teacher assume the role of a guide, presenting problems and challenging situations to the students, while they take on the responsibility of resolving these challenges [Pimentel 2006].

The analysis of the cognitivist paradigm reveals the presence of active learning [Pimentel 2006], an approach that has gained prominence, especially in the context of higher education [Hartwig et al. 2019]. This methodology is based on student-centered instructional methods, with the role of the teacher assuming the function of an instructor [Hartikainen et al. 2019].

The applications of these competencies in the educational context are vast, highlighting the relevance of active learning in the development of practical and social skills of students. Among the various approaches to incorporating this methodology into the teaching of Computer Science courses, notable examples include flipped classrooms, pair programming/discussions, tablet or notebook-supported learning, online learning, teacher/student learning contracts, instant feedback, project-based learning, challenge-based learning, and just-in-time learning [Hartwig et al. 2019].

Within the specific scope of this project, emphasis will be placed on project and challenge-based learning, highlighting the focus on practical application and problem-solving, aligning with the objectives of promoting environments capable of tracking the journey of knowledge acquisition.

Challenge-based learning emerges as an extension of project-based learning and problem-based learning. Both the challenge-based and project-based approaches share similarities, focusing on real-world problem-solving and the teacher's role as a guide seeking to steer students. The information provided subsequently about this methodology will be grounded in the work of Johnson et al. [Johnson et al. 2009]. The creation of this approach aims to engage students, direct the curriculum towards real-world problem-solving, promote collaborative work, time management, and enable the student to drive their own learning process.

Design Thinking will also be used, a methodology that helps with the development of innovative solutions, focusing on the main target audiences of the presented solution [Sahli et al. 2022]. It divides this process into five phases: empathize, define, ideate, prototype and test [d.school Stanford 2010].

3. Methodology

The development of this research project consisted of four distinct phases. Following this, a synthesis of each stage will be provided to allow for a detailed presentation of each element in the subsequent subsections.

The first step involved initiating the construction of the scientific foundation through a literature review. The purpose was to collect information related to self-regulation of knowledge within the scope of higher education and the delineation of active learning environments. This analysis enabled the interpretation of the collected data, facilitating a comprehensive understanding of contemporary research and providing the necessary scientific foundation for this study [Hulland and Houston 2020].

The second phase of the project involved the development of a computational system designed to monitor the student's learning journey. The intention was to create a tool that would facilitate the assessment of knowledge acquisition, integrating self-regulation methods and processes previously established in the earlier phase. The tests were conducted targeting students and professors of CS-related courses.

For the development of this system, the decision was made to use the Swift programming language. Its efficiency, native integration with Apple devices, emphasis on security and privacy, as well as ease of learning, make it an option for developing a dynamic application [Apple 2023]. Besides, it must be considered the context of the development of research – a department that has a partnership with Apple at the University supported the project. Therefore, it is expected that this system will contribute to improving the self-regulation process of learning.

Following that, the third stage involved the validation of the proposed application. In this context, tests were conducted with educators, specialists in the domains of active learning and self-regulation. The execution of these tests allowed for the evaluation of the system's applicability.

As a complement to this investigation, a Mood board was developed, a process involving the collection of data on real-world solutions associated with the study's theme, capturing the desired product sensation [Preece et al. 2023]. For this purpose, an analysis of apps available in the market was conducted to understand what users are already familiar with using, as well as to identify common elements in educational systems. Additionally, ideas from the community were considered to enrich the understanding of design trends. The final result of the Mood board consists of a collage of images taken from the aforementioned contexts, presented in a clear format for later reference, available in Figure 1.

At this stage, the process of defining the components comprising the proposed solution for the problem at hand begins [d.school Stanford 2010]. To achieve this objective, a set of requirements was developed to provide a summarized view of the desires of students and teachers, comparing them with potential functionalities that would be available on the platform. Additionally, the learning objectives were determined to illustrate the use of the application. Subsequently, both topics will be addressed in greater detail.

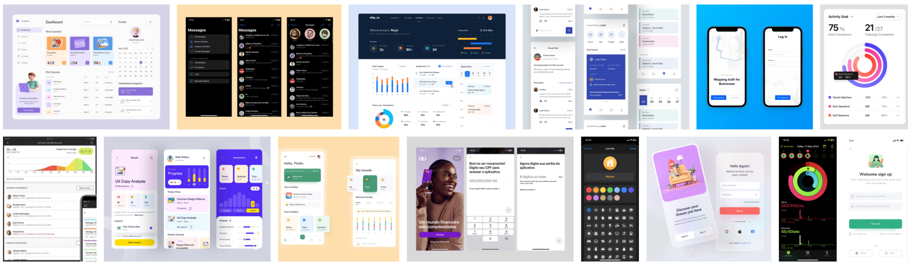

Figure 1. System's Mood board

4. Development

Following the theoretical framework review, the conception of the system was initiated, based on self-regulated learning to assess a student's knowledge acquisition. The information gathered in the previous phase provided the necessary scientific foundation to address the proposed hypothesis.

To implement this proposal, the use of a specific problem-solving technique became necessary. In this context, the decision was made to employ Design Thinking, an ideology that enables the development of creative, user-centered, competitive, and easily applicable solutions in real-world situations [Sahli et al. 2022]. Design Thinking emerges as an approach that identifies individual needs and aims to create innovative solutions using tools and methods characteristic of designers. Dobrigkeit and Paula [Dobrigkeit and de Paula 2019] present the evolution of the role of Design Thinking over time. The authors state that the core principles of Design Thinking focus on user prioritization, the application of divergent and convergent thinking, empathy, experimentation, collaboration, and creativity. Various structural models have been proposed by different researchers, with this work analyzing the one presented by Stanford [d.school Stanford 2010]. The process is structured into five phases: empathize, define, ideate, prototype, and test.

4.1. Phase 1: Empathize

In this phase, it is essential to understand the context of the presented problem and the target audience involved [Pintrich 2003]. To achieve this understanding, the theoretical framework played a pivotal role, allowing for an in-depth analysis of the topics addressed in the research. Notable are the systems studied focusing on self-regulation, which demonstrate improvements in performance, increased efficacy, time management, and progress in self-regulation skills [Lai et al. 2018], [Prather et al. 2019] and [Saadati et al. 2021]. Each of these elements helped establish effective empathy with the subject under study, as well as contributed to the construction of the state-of-the-art overview [Hulland and Houston 2020].

4.2. Phase 2: Define

At this stage, the process of defining the components comprising the proposed solution for the problem at hand begins [d.school Stanford 2010]. To achieve this objective, a set of requirements was developed to provide a summarized view of the desires of students and teachers, comparing them with potential functionalities that would be available on the platform. Additionally, the learning objectives were determined to illustrate the use of the application. Subsequently, both topics will be addressed in greater detail.

Requirements specification is an integral technique in the software development cycle in Software Engineering, being the stage where the system's impact is analyzed in relation to business objectives, client expectations, and user interaction with the application [Pressman and Maxim 2021]. To initiate this procedure, precise definition of the solution is necessary. Therefore, in the context of this research, the application is conceptualized as: "A computational system for assessing knowledge acquisition through self-regulation during assessment moments, where learners conduct self-assessments. Both the teacher and the student have access to the application, although their functionalities differ. Students perform self-assessments, involving visual reflection, textual reflection, and selecting the degree of agreement with learning objectives. Both parties can view the assessment results, as well as relevant details and analytical graphics related to the objectives. The teacher, on the other hand, can view the class as a whole and individually, as well as create a new discipline and establish assessment moments, linking them to learning objectives. The initial application will involve students and teachers from courses related to the field of Computer Science.".

Thus, it was possible to begin the elaboration of the requirements. Based on the collected information, the interests of the students can be inferred. Initially, the desire to visualize the knowledge acquisition journey is highlighted [Hartwig et al. 2019]. Furthermore, it was observed that metacognition, understood as the regulation and control of cognitive activities during the learning process, is favored by this monitoring.

Another aspect identified as significant for students is the conduct of self-assessments, an essential component of self-regulation, which contributes to cognitive and motivational processes related to learning [Pintrich 2003]. Furthermore, the evaluation of learning objectives stimulates academic performance. Finally, there is a yearning for reflections throughout the disciplines, which are part of the metacognitive process and aid in the development of self-regulation skills [Saadati et al. 2021].

Table 1 presents a set of functionalities that address these needs, including an overview with the use of graphics, self-assessment in relation to each learning objective, and the execution of textual and visual reflections at specified assessment moments. This table also outlines the connection with the theoretical framework.

Table 2 highlights the functionalities aligned with the educator's needs, including visualization of assessed objectives and individual reflections, an overall graph, class filtering, and the option to create a course. In the last column, the graph presents the references that were used to substantiate this connection.

In the context of the instructors, essential needs were identified to support their students. Firstly, a comprehensive visualization of each student and the class as a whole would be beneficial, allowing the instructor to obtain the necessary information to offer assistance to individuals in need. Another requirement pertains to the ability to select the learning objectives that will comprise a course, as these objectives form the basis for the self-regulation process, triggering reflective moments of perception and behavior in the student.

With these concepts, interactions within the application begin to become more

Table 1. Students' Expectations versus Functionalities

Goals	*Functionalities*	*References*
Assist the learner in broadly visualizing the progression of their knowledge.	Overview of the evolution of learning objectives using graphics.	(LIVINGSTON, 2003)
Enable the learner to engage in the process of self-regulation in relation to the learning objectives selected by the instructor.	Reflection on the assessment moment, where students can conduct a self-assessment related to each objective.	(PINTRICH, 2003)
Enable the learner to engage in reflections at the assessment moments determined by the instructors.	Reflections on the assessment moment, with visual and textual interactions.	(ASH et al., 2005)

Table 2. Professors' Expectations versus Functionalities

Goals	*Functionalities*	*References*
Allow the instructor to have an overview of the progress of knowledge acquisition for each learner.	Individual visualization of each learner, with their assessed objectives and reflection data.	(KENDAL, 2018)
Allow the instructor to have an overall view of the progress of the class as a whole.	Overall class visualization, able to see those who have not reached the minimum levels on a general chart.	(ASH et al., 2005)
Allow the instructor to choose the moments when the learning objectives will be assessed.	Creation of a course, adding the learning objectives and their expected levels.	(MITCHELL and MANZO, 2018)

tangible. Furthermore, by outlining user needs based on a survey, this process has been facilitated.

4.3. Phase 3: Ideate

The next step, known as Ideate, requires a creative approach that transcends conventional standards [d.school Stanford 2010]. Several activities can be part of this process; however, for this project, low-fidelity prototypes, and the conventional Software Engineering diagrams were developed. The prototypes will be detailed below.

To begin the creative process, low-fidelity prototypes were designed, available for reference in Figure 2. These prototypes enabled agile and effective exploration of various ideas, as well as capturing the desired product sensations [Preece et al. 2023].

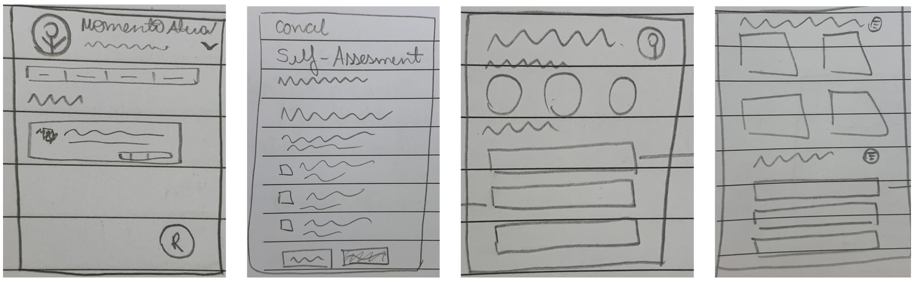

Figure 2. Low-Fidelity Prototypes

4.4. Phase 4: Prototype

A significant part of assessing knowledge acquisition is self-regulation, which contributes to the control of cognitive processes [Pintrich 2003]. Therefore, in the proposed system, when prompted by the instructor to complete an assessment, the student will engage in a visual reflection, a textual reflection, and a self-assessment of the learning objectives corresponding to the evaluation moment, thus exploring the foundations of self-regulation.

This process can be visualized in Figure 3, showing some high-fidelity prototypes, as follows.

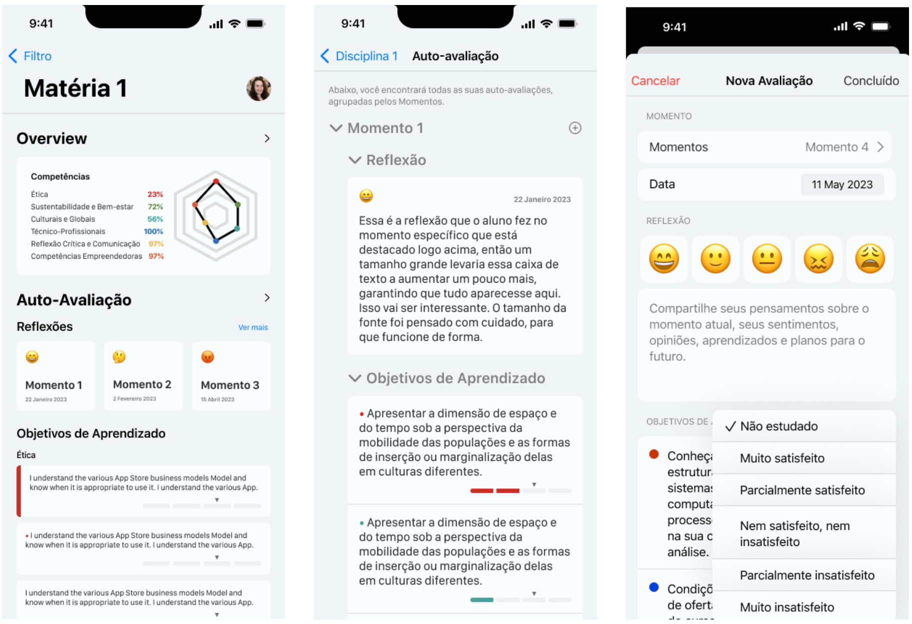

Figure 3. First Version of Students' Flow (Screenshots in Portuguese)

The reflection involves questioning the student's feelings regarding the evaluation moment, using images that represent a visual adaptation of the five-level Likert scale, which provides a way to assess the individual's attitudes. Additionally, the fact that these representations are commonly used in existing applications provides familiarity. To conclude the reflection, the student should write their perceptions, opinions, and learning regarding the moment in evidence.

The self-reflection process is completed in the software when the user performs the self-assessment of each of the learning objectives. Again, this is based on Likert, now with six levels, thus not allowing neutral responses like odd scales.

In relation to the instructor, there are two relevant flows. The first consists of the overall class visualization, as presented in Figure 2. In this view, the instructor has access to a bar graph, where each bar represents one of the competencies and the number of learning objectives completed by the class up to that point. The diagram also indicates the goal according to the levels selected by the teachers for each objective. Just below, there is a list with all the students, accompanied by filters. The first refers to students who assessed at least one objective above the expected level after at least two self-assessments. The second is similar to the previous one but related to levels below the expected. The next filter presents the learners who have not assessed more than two objectives after at least one self-assessment. Next, there is the filter for those who have already completed all assessments, and finally, the set that is outside the expected level. When the instructor selects a student, they can specifically see which objectives that student assessed below the expected level, which ones were not assessed, as well as the history of all self-assessments already completed. This way, the instructor has the necessary tools to monitor the cognitive evolution within their disciplines.

The second flow that the instructor can perform is the creation of a discipline, as illustrated in Figure 4. The main interaction of this step consists of creating assessment moments. To do so, the instructor must select the option "Assessment Moments" and be directed to the subsequent screen, where it is possible to name the moments and choose which objectives will be part of them. This way, the application provides the student with the learning objectives that form the basis of the self-regulation process.

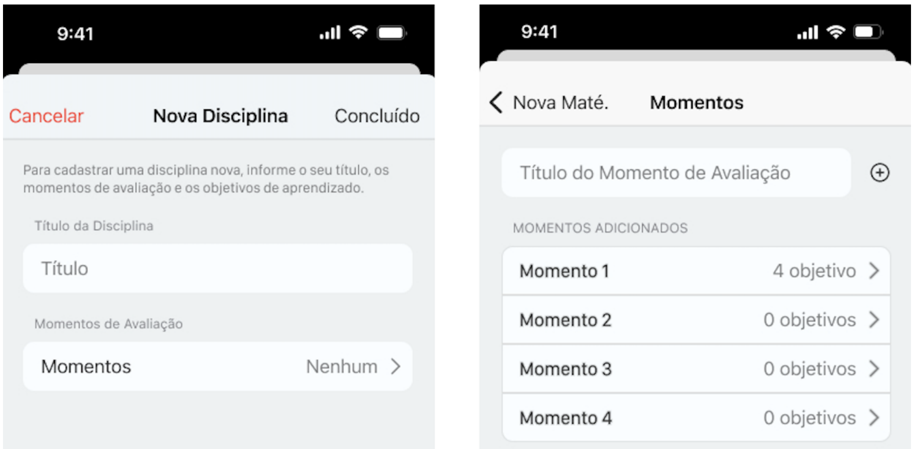

Figure 4. Course Creation Flow - First Version (Screenshots in Portuguese)

4.5. Phase 4: Test

After the development of the initial versions of the system interface, it was necessary to validate it. This evaluation aims to verify if the system meets the users' needs, employing tests that range from informal approaches to analyses based on methodologies [Pressman and Maxim 2021]. In this context, we chose to use the concept of Lenses, a process for evaluating mobile device interfaces designed, by applying heuristics to each of the analysed flows [Santos 2021]. This approach allowed us to apply the various heuristics ourselves, as such, conducting an evaluation focused on the 'Usability' Lens, before presenting the final results to the target audience.

This process led to the reformulation of both sets of interfaces, identifying and correcting some important usability fails, as can be seen in Figs. 5 (students' flow) and 6 (professors' flow).

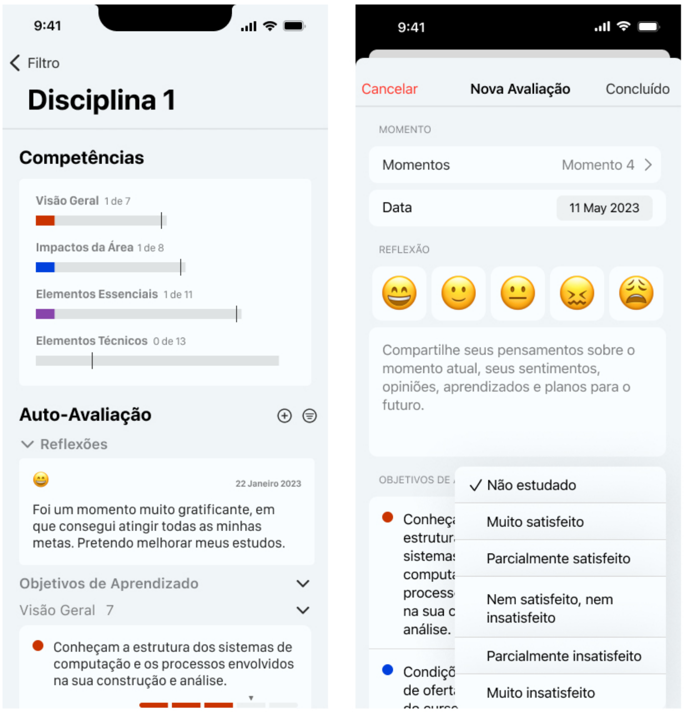

Figure 5. Second Version of Students' Flow (Screenshots in Portuguese)

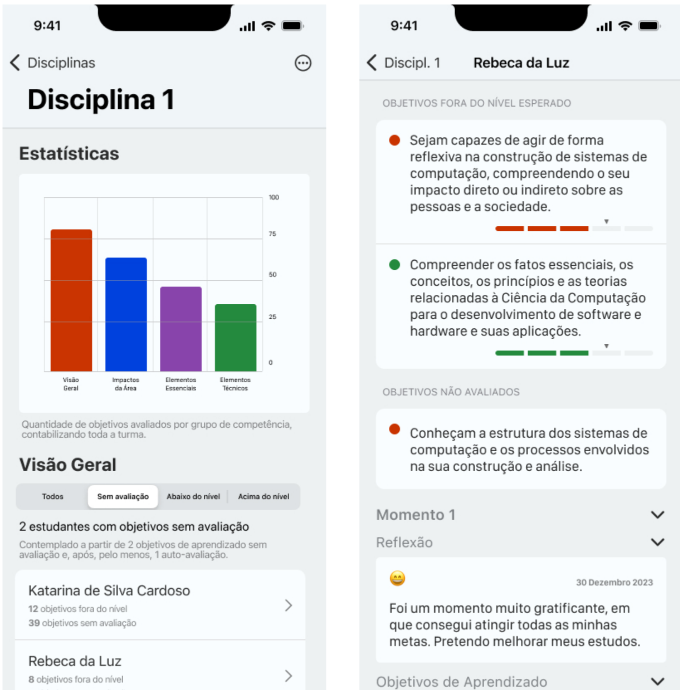

Figure 6. Second Version of Professors' Flow (Screenshots in Portuguese)

Although there are four categories of Lenses, at this moment, it was decided to focus on the 'Usability' Lens, aiming to improve the software interface. These Lenses are composed by a set of cards that have to be applied, each with their own questions. As such, all of the 'Usability' card were applied to the flows. After this definition, the application of the process was conducted, analyzing the three mentioned flows: student, teacher, and discipline creation. In each analysis, a sequential execution of the planned use was followed, resulting in tables that highlighted usability issues.

5. Tests and Users' Validation

Considering that the Design Lenses approach helped the continuous validation by students – mentioned previously, final users' validation was performed with CS professors. The validation phase was conducted to determine the acceptability of the proposed system within the context of the participating educators in the interview. This process was configured by a questionnaire that was followed by a semi-structured interview, previously approved by the Ethical Committee of the University.

In order to facilitate data analysis, the application of narrative analysis was chosen, a method that gained popularity in the 1990s but does not have a single form of implementation due to its interdisciplinary nature [Snyder 2019]. Essentially, the application of this method stands out in analyzing texts in their context as data or evaluations [Wang et al. 2020]. Therefore, this was deemed the best choice to implement, and this approach was maintained throughout the execution of the process.

The first step involved transcribing the audio recordings from the interviews, excluding the interviewer's comments, to ensure the integrity and reliability of the obtained testimonies. Subsequently, an initial reading of the texts was conducted to gain familiarity and establish connections with the material. In the third stage, the information presented by the participants was categorized. Finally, a synthesis of the collected information was carried out, transforming the data into contextualized relationships with the objectives and applicability of the research project.

The analysis and interpretation of ideas and opinions revealed a consensus regarding the feasibility of implementing the system in the daily routine of all interviewed teachers and educators in the context of CS-related courses.

6. Conclusions

The effective implementation of educational technologies requires consideration of the context in which they are applied. In this sense, the definition of the context limited to CS-related courses and their students and professors aimed to explore potential opportunities for integrating the proposed system into this specific educational environment. By understanding the key moments during the course where the application can be beneficial, its utility can be optimized, enhancing the learning experience for students.

The analysis of the impact of the application on self-regulated learning contributes to understanding how technologies can influence the teaching and learning processes, directing the focus to the intersection between educational technology and the student's ability to regulate their own knowledge acquisition technique.

Specific elements that comprise the solution and contribute to self-regulated learning are highlighted, such as the use of graphics, self-assessment, and the opportunity to express feelings associated with assessment. Lastly, fundamental points for the success of the proposal were identified, including clarity in explaining the system's functionality and the concept of self-regulation, as well as the implementation strategy in higher education institutions.

During the progression of the interviews, an analysis of the interaction between the system under study and the process of assessing student knowledge acquisition emerged, exploring the intersection between educational technology, and learning assessment.

The outcome of this research led to the proposal of a computational system designed to track students' knowledge acquisition in higher education, limited in this study to professors and students of CS-related courses. This solution was implemented to benefit both students and professors. Students have the option to conduct a self-assessment, where they can input visual reflections, textual reflections, and individual self-assessments for each of the objectives belonging to the evaluation moment.

Upon the conclusion of this study, it became evident that a system aiding in the monitoring of the learning journey that incorporates principles of self-regulation, yields positive results. The tests that were conducted with students and professors of CS-related courses contributed to this conclusion, highlighting that different learning and teaching contexts and realities could benefit from this approach. The system provides detailed insights for professors and encourages students to reflect on their learning, promoting a more humanized approach and considering soft skills development. By achieving the proposed objectives, this work points to the relevance of a system applicable in the context of higher education, enriching the learning experience and contributing to the academic development of students.

As future works, it would be relevant to apply this system in classrooms. This would verify the true applicability and scalability of the system in real-life scenarios. With this, contributing to the learners' continuous learning journey.

References

Apple (2023). Human interface guidelines. Technical report, Apple. Acessado em 11/2023.

Bloom, B., S., Hastings, J., T., and Madaus, G., F. (1983). *Manual de Avaliação Formativa e Somativa do Aprendizado Escolar*.

Bransford, J. D., Brown, A. L., and Cocking, R. R. (2003). *How people learn: Brain, mind, experience, and school.*

Clark, K. (2018). Learning theories: Cognitivism. *Radiologic technology*, 90:176–179.

Dewaele, J.-M. and Li, C. (2021). Teacher enthusiasm and students' social-behavioral learning engagement: The mediating role of student enjoyment and boredom in chinese efl classes. *Language Teaching Research*.

Dias Canedo, E., Santos, G. A., and de Freitas, S. A. A. (2017). Analysis of the teaching-learning methodology adopted in the introduction to computer science classes. In *2017 IEEE Frontiers in Education Conference (FIE)*, page 1–8. IEEE Press.

Dimensions (2023). Technical report, https://www.dimensions.ai. Acessado em: 12/2023.

Dobrigkeit, F. and de Paula, D. (2019). Design thinking in practice: Understanding manifestations of design thinking in software engineering. In *Proceedings of the 2019 27th ACM Joint Meeting on European Software Engineering Conference and Symposium on the Foundations of Software Engineering*, ESEC/FSE 2019, page 1059–1069, New York, NY, USA. Association for Computing Machinery.

d.school Stanford (2010). An introduction to design thinking - process guide. Technical report.

Hartikainen, S., Rintala, H., Pylväs, L., and Nokelainen, P. (2019). The concept of active learning and the measurement of learning outcomes: A review of research in engineering higher education. *Education Sciences*, 9(4).

Hartwig, A., Silveira, M., Fronza, L., Mattos, M., and Kohler, L. (2019). Metodologias ativas para o ensino da computação: uma revisão sistemática e um estudo prático. *Anais do Workshop de Informática na Escola*, 25(1):1139–1143.

Hulland, J. and Houston, M. B. (2020). Why systematic review papers and meta-analyses matter: An introduction to the special issue on generalizations in marketing. *Journal of the Academy of Marketing Science*, 48:351–359.

Johnson, L. F., Smith, R., Smythe, J. T., and Varon, R. (2009). Challenge-based learning: An approach for our time.

Lai, C.-L., Hwang, G.-J., and Tu, Y.-H. (2018). The effects of computer-supported self-regulation in science inquiry on learning outcomes, learning processes, and self-efficacy. *Educational Technology Research and Development*, 66:863–892.

Paiva, J. C., Leal, J. P., and Figueira, A. (2022). Automated assessment in computer science education: A state-of-the-art review. *ACM Trans. Comput. Educ.*, 22(3).

Pimentel, E. (2006). *Um modelo para Avaliação e Acompanhamento Contínuo do Nível de Aquisição de Conhecimentos do Aprendiz.* PhD thesis, Instituto Tecnológico de Aeronáutica.

Pintrich, P. (2003). A motivational science perspective on the role of student motivation in learning and teaching contexts. *Journal of Educational Psychology - J EDUC PSYCHOL*, 95:667–686.

Prather, J., Pettit, R., Becker, B. A., Denny, P., Loksa, D., Peters, A., Albrecht, Z., and Masci, K. (2019). First things first: Providing metacognitive scaffolding for interpreting problem prompts. In *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, SIGCSE '19, page 531–537, New York, NY, USA. Association for Computing Machinery.

Preece, J., Rogers, Y., and Sharp, H. (2023). *Interaction Design: Beyond Human-Computer Interaction*. John Wiley & Sons, Inc., EUA.

Pressman, R. S. and Maxim, B. R. (2021). *Engenharia de Software: Uma Abordagem Profissional*. AMGH, 9ed edition.

Ramos, V. A. B. (2019). A motivação e o sucesso escolar. Technical report. Psicologia.pt - O Portal dos Psicólogos.

Saadati, Z., Zeki, C. P., and Barenji, R. V. (2021). On the development of blockchain-based learning management system as a metacognitive tool to support self-regulation learning in online higher education. *Interactive Learning Environments*, 0(0):1–24.

Sahli, A., Pei, E., Manohar, A., and Evans, R. (2022). Knowledge visualization: A design centered framework. *Procedia CIRP*, 109:629–634. 32nd CIRP Design Conference (CIRP Design 2022) - Design in a changing world.

Santos, K. S. d. P. (2021). Processo de avaliação de interfaces para dispositivos móveis: uma proposta. Master's thesis, Universidade Presbiteriana Mackenzie.

Snyder, H. (2019). Literature review as a research methodology: An overview and guidelines. *Journal of Business Research*, 104:333–339.

Ulbricht, K. (1982). Summative evaluation of a course of study in computer science. *Higher Education*, 11.

Veiga, A. V. L. d. (2017). A participação dos alunos na regulação dos processos de aprendizagem. Master's thesis, Instituto Politécnico de Lisboa. Acessado em: 10/10/2022.

Wang, L., Ang, L. H., and Abdul Halim, H. (2020). A systematic literature review of narrative analysis in recent translation studies. 28:1–16.

Zheng, L., Li, X., Zhang, X., and Sun, W. (2019). The effects of group metacognitive scaffolding on group metacognitive behaviors, group performance, and cognitive load in computer-supported collaborative learning. *The Internet and Higher Education*, 42:13–24.