Scientific Paper

Revista sítio novo
Instituto Federal do Tocantins

Adaptive Testing for programming logic skills assessment

https://doi.org/10.47236/2594-7036.2026.v10.1905

Lucas Montagnani Calil Elias[1]
Francisco de Assis Zampirolli[2]

Submission date completed: November 5, 2025. Approval date: January 19, 2026. Publication date: February 2, 2026.

Abstract – Adaptive Testing (AT) enhances learning outcomes by adjusting assessments to students' proficiency levels. This paper presents adaptive methods for evaluating programming logic skills, implemented in an open-source system named MCTest. In this system, teachers create ATs tailored for their students. Three adaptive methods were developed: Semi-AT (SAT), Weighted Probability of Correction (WPC), and Maximum Likelihood Estimation (MLE). Six tests were designed, including a non-adaptive baseline, with multiple-choice questions classified according to Bloom's Taxonomy. These tests validated item calibrations using Item Response Theory. The method was applied in two classes with 72 students, and a final questionnaire with 17 respondents statistically confirmed its perceived effectiveness.
Keywords: Adaptive Testing. Education. Item response theory. Programming logic. Maximum likelihood estimation.

Avaliação Adaptativa de habilidades de lógica de programação

Resumo – A Avaliação Adaptativa (AA) aprimora os resultados de aprendizagem ajustando as avaliações à proficiência dos estudantes. Este artigo apresenta métodos adaptativos para a avaliação de habilidades da lógica de programação na educação, implementados em um sistema de código aberto denominado MCTest. Neste sistema, os professores criam AA personalizadas para seus estudantes. Três métodos adaptativos foram desenvolvidos: Testagem Semi-Adaptativa (SAT), Probabilidade Ponderada de Correção (WPC) e Estimativa de Máxima Verossimilhança (MLE). Seis testes foram concebidos, incluindo uma linha de base não adaptativa, com questões de múltipla escolha classificadas de acordo com a Taxonomia de Bloom. Esses testes validaram as calibrações de itens usando a Teoria de Resposta ao Item. O método foi aplicado em duas turmas com 72 estudantes, e um questionário final com 17 respondentes confirmou estatisticamente sua eficácia percebida.
Palavras-chave: Avaliação Adaptativa. Educação. Estimativa de máxima verossimilhança. Lógica de programação. Teoria de resposta ao item.

Evaluación Adaptativa de habilidades de lógica de programación

Resumen – La Evaluación Adaptativa (EA) mejora los resultados de aprendizaje al ajustar las evaluaciones al nivel de competencias de los estudiantes. Este artículo presenta métodos adaptativos para la evaluación de habilidades de lógica de programación en la educación, implementados en un sistema de código abierto

[1] Bachelor's Degree in Computer Science at the Federal University of ABC. Santo André, São Paulo, Brazil. lucas.montagnani@aluno.ufabc.edu.br https://orcid.org/0009-0006-4746-1551 http://lattes.cnpq.br/3276534519963546.
[2] Ph.D. in Electrical Engineering from State University of Campinas. Full Professor of Computer Science at the Federal University of ABC. Santo André, São Paulo, Brazil. fzampirolli@ufabc.edu.br https://orcid.org/0000-0002-7707-1793 http://lattes.cnpq.br/4127260763254001.

llamado MCTest. En este sistema, los profesores crean EAs personalizadas para sus estudiantes. Se desarrollaron tres métodos adaptativos: Evaluación Semi-Adaptativas (SAT), Probabilidad Ponderada de Corrección (WPC) y Estimación de Máxima Verosimilitud (MLE). Se diseñaron seis pruebas, incluida una línea de base no adaptativa, con preguntas de opción múltiple clasificadas según la Taxonomía de Bloom. Estas pruebas validaron las calibraciones de ítems utilizando la Teoría de Respuesta al Ítem. El método se aplicó en dos clases con 72 estudiantes, y un cuestionario final con 17 encuestados confirmó estadísticamente su eficacia percibida.
Palabras clave: Evaluación Adaptativa. Educación. Estimación de máxima verosimilitud. Lógica de programación. Teoría de respuesta al ítem.

Introduction

Education in recent years has undergone a significant evolution in how technology is integrated with teaching practices (Gros, 2016). This is driven by advancements that open new avenues to enhance learning and cater to individual student needs. Educators can now analyze student performance data to adapt their strategies, leading to the rise of adaptive learning — an approach that recognizes individual strengths, weaknesses, and interests (Becker *et al.* 2018). This allows for targeted interventions (Costa *et al.* 2022).

Works like Johnson *et al.* (2016) acknowledge technology as an educational tool, while Pellegrino and Quellmalz (2010) highlights its potential to enrich assessments. Research by Ghavifekr and Rosdy (2015) suggests technology based teaching surpasses traditional methods by creating engaging environments. In this context, recent studies illustrate diverse technological strategies: Pontes and Victor (2022) explored educational robotics for programming logic; Alves and Santos (2022) applied gamification in mathematics; and Oliveira et al. (2025) proposed software solutions to stimulate critical thinking. Expanding on these innovations, Soares et al. (2025) highlighted the potential of Generative AI for academic support. Other studies, like Moran (2015), emphasize integrating technology across all learning spaces, and Moreira and Schlemmer (2020) finds technological evolution fosters innovation and transformation in education. However, challenges remain. Limited access to technology and inadequate teacher training require attention, as noted in Alves *et al.* (2020). Addressing high failure rates in introductory STEM courses, Alves et al. (2022) emphasize that continuous assessment methodologies are effective in reducing student retention, a principle that aligns with the adaptive approach proposed herein.

Based on this context, this paper aims to enhance student motivation by offering teachers resources to provide adaptive tests aligned with students' abilities. Essentially, the difficulty of the tests will dynamically adjust to each student's proficiency level. Consequently, it is anticipated that this approach will foster greater student engagement. It is crucial to emphasize that these tests are exclusively formative, having no impact on the students' final grades, except for participation. By providing timely feedback and tailoring items to individual strengths and weaknesses, these formative assessments can cultivate a growth mindset. In essence, students with lower performance can remain motivated and avoid dropping the course, while those with higher performance can continue to receive challenging items that stimulate their intellectual curiosity, thereby maintaining their motivation to engage in the course.

This proposal differs from existing approaches by introducing a hybrid adaptive framework integrated into the open-source system MCTest (Zampirolli 2023). Unlike traditional Computerized Adaptive Testing (CAT) that relies on real-time computer

access, this method generates individualized hardcopy tests (referred to as exams in this paper). This design choice addresses infrastructure limitations and significantly reduces the potential for plagiarism, as each student takes a distinct test offline. With its capability to handle parametric items through integrating Python code and LaTeX editing, MCTest enables the generation of numerous test variations from a pool of items. By leveraging student performance data from previous tests, the system selects a variation tailored to each student's individual skill and knowledge levels.

The proposed workflow operates as follows: first, the instructor designs a pool of multiple-choice items classified according to the first three levels of Bloom's Taxonomy (Remembering, Understanding, Applying) across the six course topics (Krathwohl 2002). The assessment cycle begins with a non-adaptive (random) test to establish an initial proficiency baseline. From the following week onwards, assessments become adaptive, alternating between the three methods detailed in this study (SAT, WPC, and MLE). Finally, students complete these printed exams offline, which are subsequently scanned and automatically corrected by the system, ensuring a continuous loop of personalized feedback.

Background

Adaptive learning uses technology to monitor students' progress and dynamically adjust teaching methods based on collected data, personalizing the learning journey to individual skills and progress, as described by Becker (2018). This approach involves technologies that modify course content according to the student's abilities, improving performance through automated adjustments and instructor interventions (Pugliese 2016), resulting in a more effective learning process. Waters (2014) emphasizes that adaptive learning strategies adapt the student experience based on performance and interaction with course materials, creating a flexible and personalized learning environment. Paramythis and Loidl-Reisinger (2003) identifies four categories of adaptation in learning environments: (1) adaptive interaction (interface adjustments), (2) adaptive course delivery (personalized course content), (3) content discovery and assembly (selection of relevant learning material), and (4) support for adaptive collaboration (facilitating communication and collaboration). The ATs covered in this paper are more related to the second and third categories. Below is a summary of the theoretical foundation used.

Computerized Adaptive Testing (CAT)

CAT offers significant advancements in assessment by providing superior accuracy and efficiency. It tailors the test by selecting items based on difficulty and the examinee's performance, requiring fewer items to determine a score (Wainer *et al.* 1990; Lazarinis *et al.* 2010). This allows for shorter tests with immediate results (Meijer and Nering 1999). CAT's adaptability enables it to administer only essential items, overcoming item number limitations (Hammond *et al.* 2014). However, implementing CAT can be expensive and requires pre-testing all items for stable statistics (Wainer *et al.* 1990; Meijer and Nering 1999).

Item Response Theory (IRT)

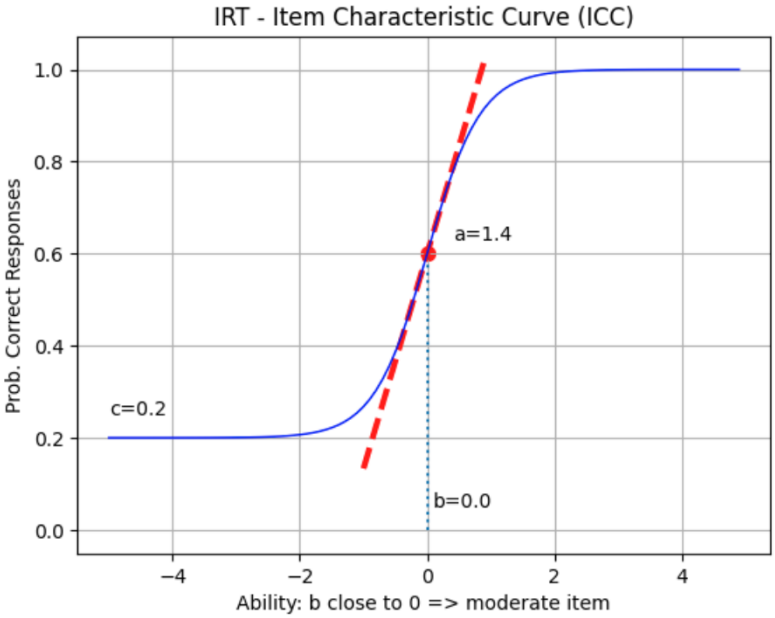
IRT is instrumental in various testing domains, including CAT, where it tailors the test to individual proficiency levels (Cai *et al.* 2016). By estimating the probability of a correct answer based on individuals' latent traits and test items, IRT models optimize information gain and reduce testing time (Yang *et al.* 2022). Despite the challenges of pre-calibration of test items, model sensitivity, strict assumptions, and sample size

requirements, IRT offers significant advantages such as increased accuracy, scale independence, and various reliability assessment methods (Hamdare 2014).

Ability estimation

IRT is crucial in AT, primarily for estimating the student's ability (θ), or proficiency, based on their correct and incorrect answers to the assessment items (Baylari and Montazer 2009; Wainer *et al.* 1990). The probability of a student correctly answering a test item, denoted by P(θ), varies with the student's ability and the item's difficulty (Baker *et al.* 2017). This relationship can be represented by the Item Characteristic Curve (ICC), a smooth S-shaped curve seen in Figure 1[3] (Wang 2006). Each item has its own ICC, making it the fundamental principle behind IRT (Baker *et al.* 2017).

Figure 1 – Example of an Item Characteristic Curve for $a = 1.4$, $b = 0$ and $c = 0.2$.



There are three IRT/ICC calculation models, known as 1PL (Parameter Logistic), 2PL, and 3PL, based on the number of parameters in their mathematical formula (Karino and Souza 2012; Galvao *et al.* 2013; Baker *et al.* 2017; Binh and Duy 2016):

1PL (one parameter): Also known as the Rasch model, it is represented by $P(\theta) = \frac{1}{1+e^{-(\theta-b)}}$, where θ is the student's ability estimate, b is the item difficulty parameter, which is expressed on the same scale as θ;

2PL (two parameters): This model maintains almost the same configuration as the previous one, with the only difference being the addition of the a parameter in its equation, which represents the item discrimination value by $P(\theta) = \frac{1}{1+e^{-a(\theta-b)}}$;

3PL (three parameters): This model complements the two-parameter model by adding a third parameter known as the guessing parameter, represented by the letter c, which represents the lower asymptote of the curve by $P(\theta) = c + (1-c)\frac{1}{1+e^{-a(\theta-b)}}$.

Item selection

[3]Figure created in colab.research.google.com/drive/1ka7_SR_QB4G7ZPVvH3p_E0bZEbOH1vhK.

To estimate examinees' ability (θ) more accurately, procedures were created to determine item parameters and ability θ as participants respond to each item (Binh and Duy 2016). These procedures are based on statistical algorithms, with one of the most common being **Maximum Likelihood Estimation (MLE):** The MLE is considered the most efficient approach and is the most widely used currently, but it has some limitations (Chen 2019). The MLE cannot be used when examinees answer all items correctly or incorrectly. This is because the MLE depends on the maximization of the likelihood function based on the item parameters and the specific response pattern (correct/incorrect) provided by the examinee. In these cases, the ability estimate becomes positive infinity (+∞) and negative infinity (-∞), respectively (Baker *et al.* 2017). Other algorithms are **Ability Estimation:** IRT uses maximum likelihood procedures to estimate an examinee's ability θ iteratively until the variation is negligible (Baker *et al.* 2017); **Item Information Function (IIF):** The final step in AT is adaptive item selection, mimicking an experienced examiner's approach (Baker *et al.* 2017). This avoids redundancy by selecting items based on the candidate's ability θ and difficulty level, aligning with the core principle of CAT (Zheng 2014). The most common method, the maximum Fisher information method, selects the item from the bank that maximizes information gain at the current ability level, similar to ability estimation using the likelihood function (Lord 1980 and Zheng 2014); **Test Information Function (TIF):** The TIF extends the concept of IIF to the entire test. The TIF assesses the accuracy of the ability estimates throughout the ability range, providing a broader picture compared to the individual analysis of the items through IIF (Baker *et al.* 2017). It is calculated by summing the information from each item's IIF at a given ability level.

This paper will present a method for selecting a variation of an exam using the Test Information Function (TIF). The proposed method, described in the next section, selects a test variation based on the TIFs of the available versions and the students' abilities in previous tests.

AT personalizes the exam experience by dynamically adjusting item difficulty based on student performance, relying on a step called calibration (Baker *et al.* 2017). Calibration determines item parameters (difficulty, discrimination) and student abilities beforehand by administering the test to a representative group and analyzing their responses using IRT to create a single ability scale for both test items and examinees (Chen 2019), establishing a reference point for interpreting future test results.

Related works

Several studies have explored the development of adaptive testing systems. One such system, Computerized Formative Adaptive Testing (CAFT) by Choi and McClenen (2020), utilizes a combination of CAT and Dynamic Bayesian Networks (DBNs) for e-learning platforms. CAFT personalizes formative assessments by dynamically selecting test items and tests based on student abilities. This approach, validated through empirical studies, offers a tailored and efficient diagnostic learning experience.

Binh and Duy (2016) introduced a study on student ability estimation using IRT and clustering via k-Means. They addressed the limitations of Classical Test Theory (CTT) in accurately assessing student abilities due to its reliance on simple scoring methods. Their approach utilized various IRT models, including 1PL, 2PL, and 3PL, to estimate both student abilities and item difficulties. The study applied MLE to estimate student abilities and employed k-Means clustering to categorize students into groups based on their abilities. The results suggested significant improvements over traditional

methods, demonstrating the potential for broader application in educational assessment systems.

Lazarinis *et al.* (2010) proposed an adaptive web-based testing system. This system personalizes tests based on a participant's performance, prior knowledge, objectives, and preferences. It utilizes student profiles to create customized assessments and deliver progress reports. This system enhances flexibility for both educators and students, particularly in formative assessments with immediate feedback.

Another relevant work by Baylari and Montazer (2009) introduces a personalized multi-agent e-learning system integrating IRT for learner ability estimation and Artificial Neural Networks (ANNs) for tailored recommendations. Utilizing IRT, the system administers ATs aligned with learner proficiency levels and employs ANNs to personalize learning material suggestions. The network architecture involves 1-2 hidden layers with sigmoid activations, trained using the Levenberg-Marquardt algorithm with early stopping mechanisms to prevent overfitting. Experimental findings indicate the system accurately recommends learning materials akin to human instructors in 83.3% of cases, showcasing the efficacy of neural networks in personalized educational contexts.

Although these and related works employ CAT for student assessment, they differ from the approach proposed in this paper, which will be detailed in the next section. This approach focuses on selecting test variations based on student abilities, rather than varying individual items within the tests themselves. Only Choi and McClenen (2020) mentioned the use of adaptive test selection based on student ability, but did not provide sufficient details about the process, and the system was not found for further analysis, making it difficult to replicate the applied method. Table 1 summarizes these related works and this approach. The column "Quest" indicates whether the paper applied a questionnaire to students to evaluate the proposed methods. In Lazarinis *et al.* (2010), a questionnaire was distributed to ten educators, but the results lacked statistical significance.

Table 1 – Comparative analysis between related works and the proposed adaptive testing approach.

Paper	**Method Used**	**Quest**	**Open-source**	**Personalization Approach**	**Additional Features**
Choi and McClenen (2020)	CAT, DBNs	No	No	Dynamic selection of items and tests based on student abilities	Tailored formative assessments
Binh and Duy (2016)	IRT (1PL, 2PL, 3PL), MLE, k-Means	No	No	Categorizes students using clustering	Improvement over CTT
Lazarinis *et al.* (2010)	Adaptive Web-based Testing	No*	No	Customizes assessments based on profiles	Immediate feedback, reports
Baylari and Montazer (2009)	IRT, ANNs	No	No	ATs and personalized learning material recommendations	Multi-agent e-learning system
Approach of this paper	IRT (3PL), Test Variation, applied on hardcopy	Yes	Yes	Selects test variations based on student abilities	Open-source implementation for broader use

Materials and method

The current MCTest project available on GitHub (github.com/fzampirolli/mctest) was developed in Django (djangoproject.com) for essential functionalities, with HTML and CSS for the web interface and MySQL for the database, deployed on Linux Ubuntu 22.04. For ATs, libraries such as NumPy and Pandas were used. Integration of the R MIRT library required using Python's RPY2 library. Each item has parameters such as Topic, Description, and Answers, with the "Bloom's Taxonomy" (revised) field being crucial for ATs with dimensions Remember, Understand, Apply, Analyze, Evaluate, and Create (Krathwohl 2002). For more details, see Zampirolli (2023).

Implemented adaptive methods

Three adaptive methods were implemented in MCTest, summarized as follows:

Semi-Adaptive Test (SAT) – The difficulty parameter b is determined by the item's Bloom taxonomy index, ranging between -2 and 3. SAT involves the manual definition of Bloom taxonomy levels by the teacher for each item;

Weighted Probability of Correction (WPC) – The difficulty parameter b is calculated as the weighted fraction of correct answers out of the total number of items answered, normalized between -5 and 5;

Maximum Likelihood Estimation (MLE) – As defined previously, the difficulty parameter b represents the skill associated with the item, normalized between -5 and 5. MLE dynamically adjusts item difficulty based on previous responses, following the principles of IRT.X

In SAT and WPC, a student's ability is calculated by first computing the element-wise multiplication of the lists b_i and u_i (1 if the student answered correctly, or 0 otherwise) for each test answered by the student, where i is an index of items in these tests. The average score for each test is then obtained as

$$S_j = \frac{\sum_i b_i u_i}{} \quad (1)$$

where S_j is the student's score for test j, and N_j is the total number of items in test j. Finally, the student's ability is determined by averaging the scores across all tests taken:

$$\theta = \frac{\sum_j S_j}{} \quad (2)$$

where θ represents the student's ability, and M is the number of tests the student has completed. Proper item calibration for WPC and MLE types requires a minimum number of participants who have already responded to each item, with MLE typically needing at least $1{,}000$ participants in the 3PL model (Min and Aryadoust 2021). If any item has not been answered, Bloom's Taxonomy is used for classification. If the student has not responded to any test, -5 is assumed.

A new solution for AT

In MCTest, a novel AT approach has been devised to address functional issues while minimizing the impact on existing system resources. Instead of treating items individually, the focus has shifted towards considering a test as the primary unit of analysis. Thus, an AT format centered on **Exams** was implemented, diverging from conventional practices observed on other platforms, which typically use items as the unit of analysis based on student ability (as noted in Table 1). In this model, an exam consists of a set of **Variations**, each potentially having different difficulty levels as

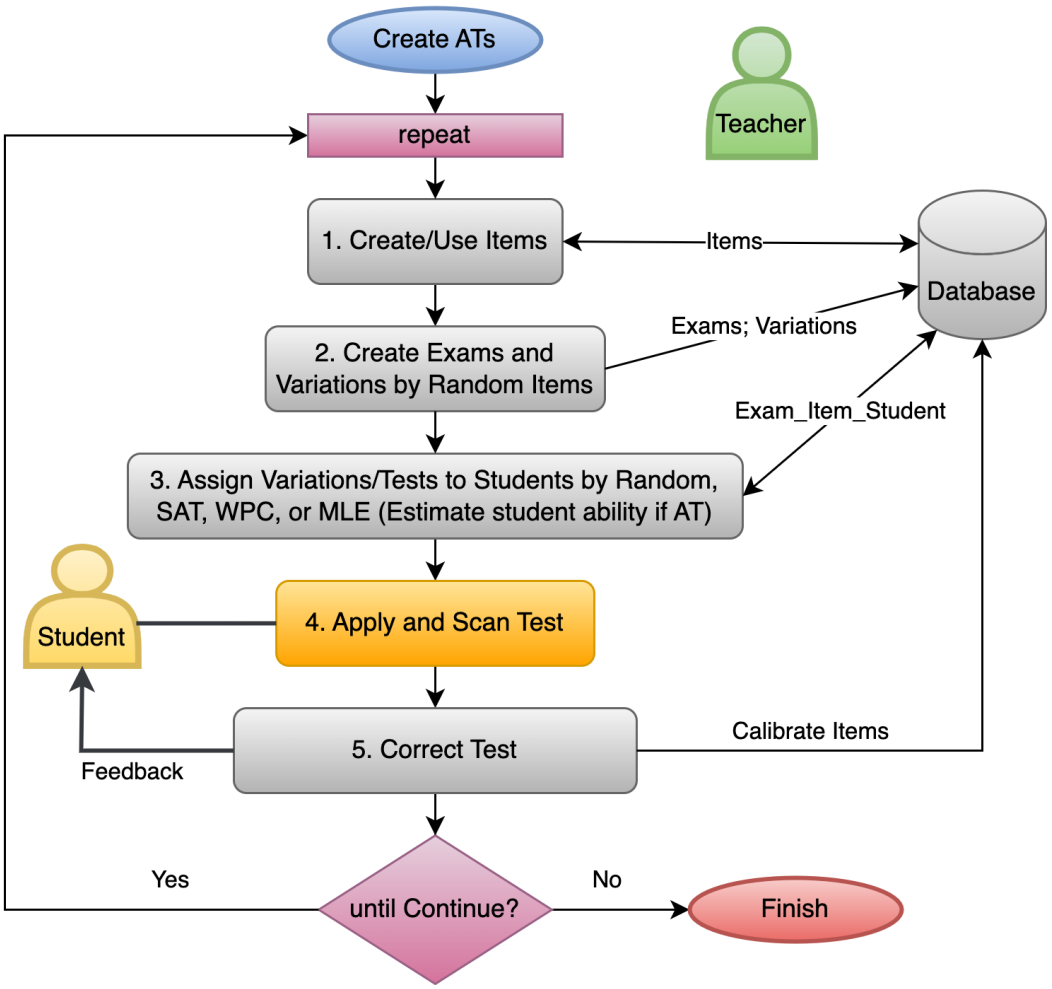
defined in the previously described TIF, and these are assigned to students based on criteria such as Random or AT.

The SAT, WPC, and MLE methods share a common CAT cycle, as illustrated in Figure 2. This process begins with (1) the creation of new items, which are initially assigned provisional parameters using IRT, or using the existing items. Then, (2) exams are constructed by creating variations through the random selection of items from the item bank. These variations/tests are then assigned to students (3) based on their proficiency levels or randomly in PDF format. After administering the test (4), it is scanned and sent for correction (5) and feedback to students, along with item calibrations. This iterative process continues until all planned exams are administered.

In the MCTest, once all course components (Institute, Course, Discipline, Topic, Class) are established, Figure 2 primarily utilizes the Exam screen to manage the entire testing process. New items are created using the Question (Item) screen in step (1). Detailed information on this process can be found in Zampirolli (2023).

The Generative AI tool Gemini Pro was utilized for grammatical review and style refinement, ensuring clarity and adherence to academic standards, as detailed in the Additional Information section.

Figure 2 – Flowchart of the Adaptive Testing (AT) process in the MCTest system, from item creation to student feedback.



Results and discussions

This section presents the results and discussion of the implementation of the method described in the previous section, beginning with the context of the

Interdisciplinary Bachelor's Program in Science and Technology, applied at the start of 2024 in two classes totaling 72 students, both taught by the same instructor. The course spanned 12 weeks, with four hours of instruction per week, held over two days in a laboratory setting.

Contextualization of programming logic course

Table 2 presents six tests designed to assess students' understanding of specific programming concepts. These tests are administered one week after covering the corresponding topic and last 30 minutes. To encourage participation, a 5% bonus is added to the final grade based on the number of completed tests rather than individual scores. The tests vary in difficulty to accommodate students' diverse abilities. Until 2023, the Programming Logic course transitioned from 5 weekly hours (3 theoretical and 2 practical) to 4 practical laboratory hours, with no changes to the curriculum. All teaching materials (available in Colab – colab.research.google.com) and assessments remained the same as in Zampirolli *et al.* (2021), except that the weekly exercise lists were replaced by the tests presented in this table. The assessments, including exercises and exams created by MCTest, were automatically graded using Moodle, a widely used learning management system (LMS) that facilitates course organization, assignment submissions, and grading automation. For programming tasks, the Virtual Programming Lab (VPL – vpl.dis.ulpgc.es) plugin was used within Moodle, allowing students to write, run, and test their code directly on the platform while enabling automated evaluation based on predefined criteria.

Table 2 – Tests and their respective topics and types.

Test	Topic	Type	Description
1	Sequential	Random	Sequential items presented randomly
2	Method	SAT	Semi-Adaptive Testing
3	Conditional	WPC	Weighted Probability of Correctness
4	Loop	WPC	Weighted Probability of Correctness
5	Array	MLE-v0	Maximum Likelihood Estimation-v0
6	Matrix	MLE-v1	Maximum Likelihood Estimation-v1

The first test is on the Sequential topic, of the Random type, indicating sequential instruction execution with randomly presented items. All variations have the same difficulty. The second test covers Methods and is of the SAT type, evaluating students' ability to work with methods and logic. Tests 3 and 4 deal with Conditionals and Loops, respectively, and are of the WPC type, focusing on the understanding and implementation of conditional and loop structures. Following these tests, the first evaluative exam (40% of the final grade) was held in week 5. Test 5 addresses the topic of Arrays and is of the MLE-v0 type, aimed at assessing knowledge of arrays and related operations. Test 6 deals with Matrices and is classified as MLE-v1, presenting a higher level of difficulty.

Each test consists of 200 variations, and one of them is assigned to the student, depending on the type chosen (Random, SAT, WPC, and MLE). All tests have 5 multiple-choice items, each with 5 alternatives, with only one correct answer. All items were created by the teacher and assigned to one of Bloom's Taxonomy, prioritizing the first three levels: Remembering, Understanding, and Applying, due to the course being an introductory programming. In week 11, the second evaluative exam (60% of the final grade) was administered, and in week 12, the recovery exam. All evaluative

exams were conducted integrating MCTest, Moodle, and VPL tools, as detailed in Zampirolli (2023). These three exams lasted two hours, and after 2023, they were conducted using the important Safe Exam Browser resource (safeexambrowser.org), without external consultation to the assessment activity in Moodle.

As all six tests were applied using the same method presented in the previous section, details of the last test will be shown in the next sections.

It is essential to note that the order of test types in Table 2 was carefully chosen to reflect the increasing complexity of their implementation within MCTest. These implementations were carried out throughout the course.

Test 5, focusing on Arrays, uses MLE-v0 in its first version. This method calculates the average of tests following Equations 1 and 2. It follows the principles used in SAT and WPC, utilizing the student's average ability in the four previous tests to determine variations proportional to the difficulty of Test 5 in a linear distribution. For example, the student with the lowest average ability receives a variation with the lowest average difficulty b_i among the 200 generated. The next section will compare this method with the classical form of TIF (Baker *et al.* 2017).

Test 6: Matrix – MLE-v1

In Test 6, on Matrices, a second version of the adaptive MLE method was used, employing the TIF concept to assign the most appropriate variation to each student. This test had a response rate of 76.4% (55/72 respondents). For more information about the corrections and the method used, see the function getHashVariationByCat() in the file UtilsLatex.py, accessible on GitHub (github.com/fzampirolli/mctest/blob/master/exam/UtilsLatex.py). In Test 6, only six variations were used out of a total of 200. With a simple code change, now, instead of taking the TIF related to the student's ability, it chooses a random test in the range ±0.05 of the TIF value. This adjustment increases the number of variations used to 15 within this range. In comparison, the WPC method used 54 distinct variations distributed linearly across students. Due to randomness, these numbers may change slightly. With a total enrollment of 72 students in both classes, the probability of two students receiving the same variance and sitting next to each other is minimal, not only in MLE-v1, but also in WPC and SAT.

Calibration of items

Table 3 presents the calibration and detailed statistics of the items answered in Test 6, grouped by Bloom's Taxonomy. Items are divided into three categories: Remember, Understand, and Apply. Each category includes a specific set of item keys in MySQL, with parameters in 1PL, 2PL, and 3PL models, and the percentage of correct answers (Mean) and standard deviation (SD).

Table 3 – IRT calibration parameters (1PL, 2PL, 3PL) and descriptive statistics (Mean/SD) for Test 6 items, grouped by Bloom's Taxonomy.

		1PL	2PL		3PL			Statistics	
	Keys	***b***	***a***	***b***	***a***	***b***	***c***	**Mean**	**SD**
Remember	264	-3.43	1.09	-3.16	1.02	-3.17	0.16	0.95	0.22
	3078	-2.22	1.00	-2.15	1.07	-1.90	0.13	0.85	0.36

	3079	0.80	1.13	0.73	1.11	0.79	0.02	0.33	0.47
Understand	3081	0.87	2.91	0.32	3.05	0.30	0.01	0.27	0.46
	3083	0.41	2.03	0.06	2.38	0.05	0.03	0.33	0.50
	3084	-1.31	-1.53	-0.56	-1.24	-0.73	0.11	0.60	0.55
	3085	0.17	0.37	0.63	0.64	1.97	0.27	0.43	0.51
Apply	3086	0.04	0.96	0.07	1.00	0.14	0.04	0.47	0.51
	3087	-0.82	1.18	-0.81	1.30	-0.61	0.11	0.57	0.51
RMSE		2.92		2.94		2.95			

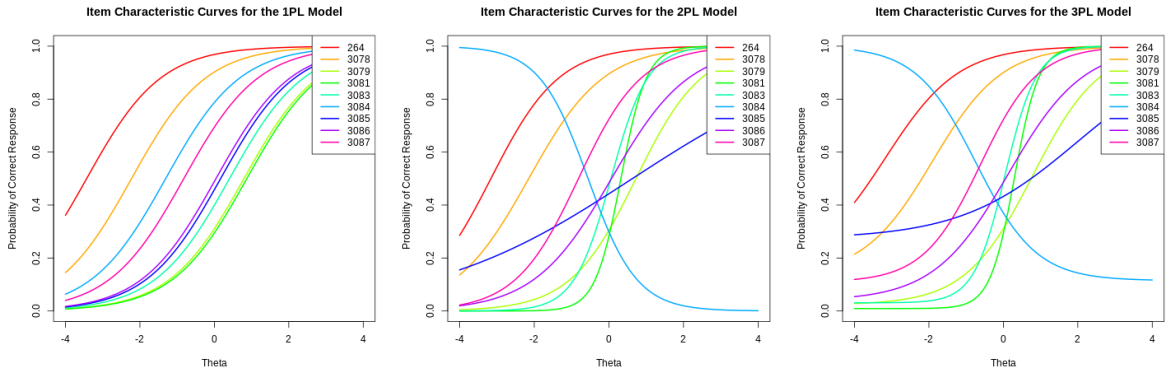
The IRT parameter estimation, or calibration, was implemented in Python in MCTest, using the 3PL model. However, this section presents analyses using the R language, which offers various graphical resources for better visualization and interpretation of results. These models were applied to the data using the mirt() function from the R MIRT library. Some important points to highlight in this table: (I) In models 1PL, 2PL, and 3PL, the difficulty values (b) are within the recommended range of –5 to 5, as implemented in MCTest; (II) The discrimination values (a) are also not within the ideal range of 0.5 to 1.5. These values were empirically estimated after some tests on Colab, see Figure 1. Some items have a low capacity to discriminate between individuals with different proficiency levels; (III) The inadequacy of the parameters to the expected ranges suggests that the items were not well calibrated, probably due to the low number of respondents in this initial analysis. Therefore, further adjustments are needed, such as collecting more data and refining the item calibration process, to improve the psychometric quality of these instruments. In Table 3, the RMSE (Root Mean Squared Error) values for the 1PL, 2PL, and 3PL models are 2.92, 2.94, and 2.95, respectively. Although RMSE values near 2.9 suggest a deviation between the model and observed data (considering the scale range), this is acceptable for an initial calibration with a limited sample size (N=72). As established in IRT literature, parameter stability increases significantly with larger samples (Baker et al., 2017). These values serve as a baseline for the continuous calibration process of the MCTest system.

Item Characteristic Curves (ICC)

Figure 3 illustrates the ICC for the 9 test items. The left panel shows the ICC under the 1PL model, the center panel displays the ICC under the 2PL model, and the right panel presents the ICC under the 3PL model. For comparison, Figure 1 depicts the ICC generated using only the Python language. By examining this figure, distinct differences in the shapes of the curves are observed. In the 1PL model, the curves are smooth and monotonically increasing, reflecting a uniform discrimination parameter across all items. The 2PL model introduces variability in the slopes of the curves due to different discrimination parameters for each item, allowing for a more nuanced understanding of item difficulty. The 3PL model further complicates the curve shapes by including a guessing parameter, which accounts for the possibility of random

guessing on easier items. This addition helps to better model the probability of a correct response, particularly for lower-ability students.

Figure 3 – Comparison of Item Characteristic Curves (ICC) for Test 6 items across.

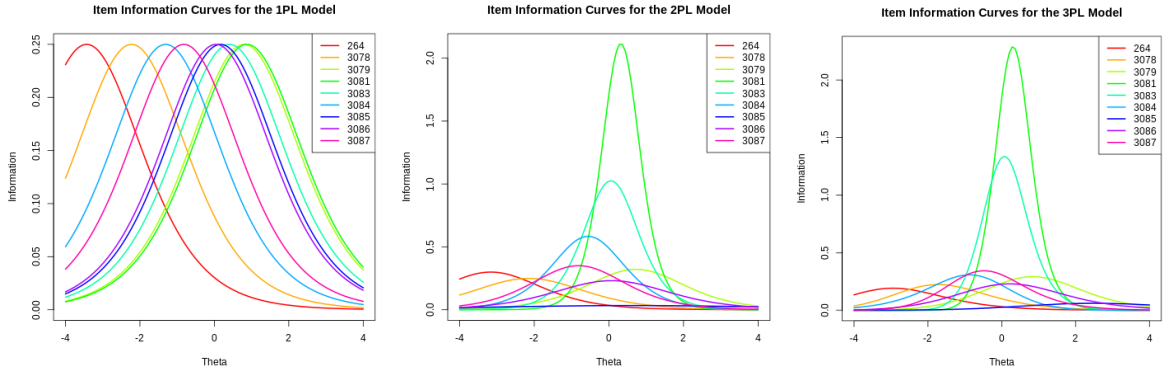


Based on the analysis of these ICCs, it is possible to conclude, for example, that item 264 is an easy question, with b = –3.17 in the 3PL model and Mean = 0.95, and does not effectively discriminate between candidates.

Item Information Curves (IIC)

Figure 4 shows the IIC for 1PL, 2PL, and 3PL models. In 2PL and 3PL, items 3081 and 3083 exhibit prominent peaks of information, indicating that they provide substantial information about individuals' abilities within a narrow range of the latent trait. This behavior is characteristic of highly discriminative items, which are better at distinguishing individuals with abilities close to the item's maximum information point but less effective for those with abilities further away from this point. In the 1PL model, the curves for each item primarily reflect the difficulty parameter b, as the discrimination parameter a is assumed to be equal across all items and does not contribute to the shape of the IIC beyond the difficulty level.

Figure 4 – Item Information Curves (IIC) demonstrating information gain per item in1PL (left), 2PL (center) and 3PL (right).



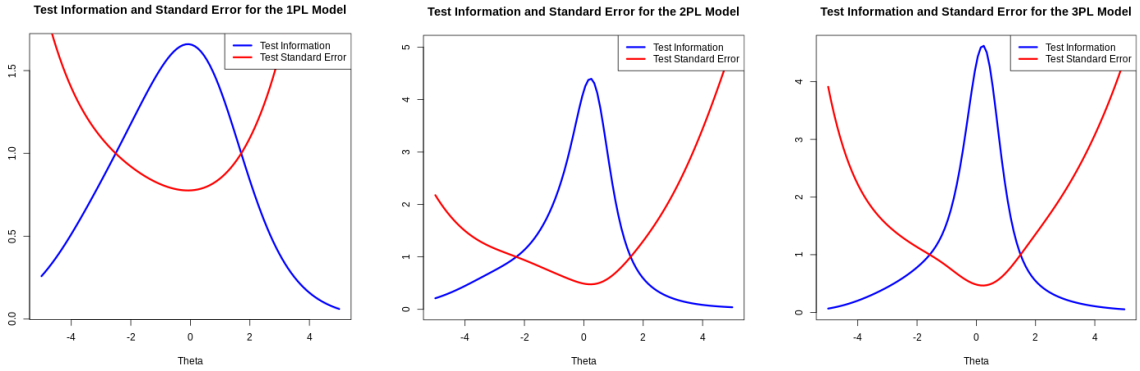
Test Information Curves (TIC) vs Standard Error of Measurement (SEM)

Figure 5 displays the TIC and SEM for the 1PL, 2PL, and 3PL models. The comparative analysis of these models reveals interesting characteristics, with maximum information around θ = 0, in addition to intersections between the TIC and SEM curves near –2 and 2. Furthermore, the bell-shaped TIC exhibits steeper decreases for 2PL and 3PL as one moves away from θ = 0, while the SEM increases

about the minimum point at θ = 0, indicating greater measurement precision near the minimum and lower precision at extreme latent traits.

By analyzing these figures, it is possible to adjust each item to better classify candidates in future exams.

Figure 5 – Test Information Curves (TIC) versus Standard Error of Measurement (SEM) for1PL (left), 2PL (center) and 3PL (right).



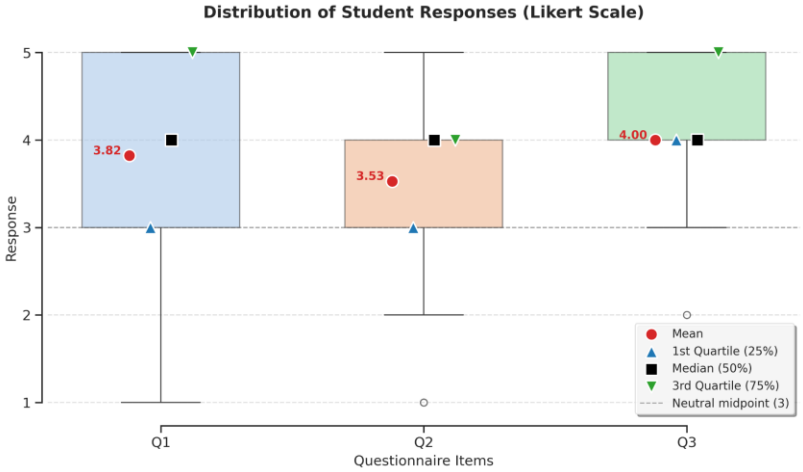
Evaluation questionnaire

For the analysis of this paper, seventy-two students were enrolled in two classes: the failure rate in class A was 55.3% (21/38), while in class B it was 32.4% (11/34). Class B follows the historical result of around 32% failure rate between 2009 and 2024, as detailed in the following section. Although these classes had the same teacher and teaching/assessment materials, this difference is beyond the scope of this work, as it may be related to the class formation criteria defined by the higher levels of the institution. As of early 2024, there were 1, 287 students enrolled in Programming Logic across 34 classes, taught by 24 professors. After the 11th week of the course, a questionnaire was made available to all enrolled students; however, only 17 of them (23.6%) responded, 6 in class A and 11 in class B.

Applied questions

The questionnaire consisted of 20 questions, with results similar to those of previous publications Zampirolli (2023), except for the ATs, as it is a new resource. The Likert scale used ranged from 1 – Strongly Disagree to 5 – Strongly Agree. This section focuses on presenting the results of three AT-related questions: **Q1 – I consider weekly individual tests important; Q2 – There was an improvement in confidence and understanding of programming logic concepts; Q3 –- The adaptive tests were challenging.** Figure 6 explores students' perceptions of these questions using BoxPlot (Tukey 1977), with means of 3.8, 3.5, and 4.0, respectively.

Figure 6 – BoxPlot distribution of student perceptions regarding Adaptive Testing (AT) on a 5-point Likert scale. The items represent: Q1 (Importance of tests), Q2 (Confidence improvement), and Q3 (Level of challenge). Markers indicate the Mean (red circle), Median/2nd Quartile (black square), 1st Quartile (blue triangle), and 3rd Quartile (green triangle). The horizontal dashed line marks the neutral midpoint (3).



Statistical analysis of results

Table 4 presents the results of the statistical analysis conducted on the responses of 17 students regarding Adaptive Tests (AT). The objective of the analysis was to examine whether students' perceptions were significantly positive, that is, greater than the neutral reference value of 3 on a five-point Likert scale.

Given the ordinal nature of Likert-type data and the relatively small sample size, a non-parametric inferential approach was adopted. Although data normality was assessed using the Shapiro-Wilk test (Shapiro and Wilk, 1965), this evaluation was performed for completeness, as Likert-scale responses are discrete and often deviate from normality by construction. The results indicated non-normal distributions for all items ($p < 0.05$). Consequently, the one-sample Wilcoxon Signed-Rank Test (Wilcoxon, 1945) was employed to evaluate whether the median response for each item was significantly greater than the neutral value. To control the family-wise error rate arising from multiple hypothesis testing, the Holm-Bonferroni correction was applied to the p-values (Holm, 1979). The hypotheses tested were defined as follows:

H_0**: The method had a neutral or negative effect on students' learning perception (median ≤ 3);**

H_1**: The method had a positive effect on students' learning perception (median > 3).**

Effect sizes were computed using the r statistic, defined as Z divided by the square root of the sample size ($r = Z / \sqrt{N}$), where Z is the standardized test statistic obtained from the Wilcoxon test and N is the number of observations. Effect sizes were interpreted according to Cohen's conventional benchmarks: 0.1 (small), 0.3 (medium), and 0.5 (large) (Cohen, 1988; Rosenthal, 1984). Mean and standard deviation values are reported solely for descriptive purposes, whereas all inferential conclusions are based on median-centered non-parametric tests.

Table 4 – Statistical results using Wilcoxon Signed-Rank Test with Holm-Bonferroni correction (N = 17). Mean (SD) and Median are descriptive. S-W p denotes the Shapiro-Wilk normality test p-value. W-stat

is the Wilcoxon test statistic. Effect Size (r) is calculated as Z/√N. Bold values indicate the most favorable results or statistical significance (p ≤ 0.05).

Item	Mean (SD)	Median	S-W p	W-stat	Effect Size (r)	p-val	p-adj	Significant
Q1	3.824 (1.131)	4	0.012	89	0.573	0.009	0.018	Yes
Q2	3.529 (1.328)	4	0.013	85	0.356	0.071	0.071	No
Q3	4.000 (0.866)	4	0.013	100	0.750	< 0.001	0.003	Yes

As shown in Table 4, items Q1 and Q3 exhibited statistically significant positive effects even after applying the Holm-Bonferroni correction. The adjusted p-values for Q1 (p-adj = 0.018) and Q3 (p-adj = 0.003) are well below the 0.05 significance threshold, and both items demonstrate large effect sizes (r = 0.573 and r = 0.750, respectively). These results support the rejection of the null hypothesis for these items, indicating that students perceived the weekly individual tests as important and considered the Adaptive Tests to provide an appropriately challenging learning environment.

For item Q2, which assessed students' confidence improvement, the adjusted p-value (p = 0.071) does not allow rejection of the null hypothesis at the 5% significance level. Nevertheless, the observed medium effect size (r = 0.356) and a mean score above the neutral point (mean = 3.53) suggest a potentially meaningful positive trend that did not reach statistical significance under the current sample size. This finding indicates that the absence of statistical significance may be related to limited statistical power rather than a lack of practical relevance. Future studies with larger samples are recommended to further investigate the impact of Adaptive Tests on students' confidence.

Historical analysis of programming logic failures

Table 5 presents a historical overview of failure rates in the Programming Logic course at the institution associated with this study, which operates on a three-quarter academic calendar. The table includes information on failure rates (Failures %) and class sizes (Classes). The average failure rate (AVER) ranges from 21% to 58%, with an overall average of 32%. Notably, classes with failure rates as high as 94% were recorded in 2018.2. The number of classes also varies considerably, with some cells highlighted in pink to indicate the ideal period for students to take the course. In the other periods (Not Ideal), the students are typically those retaking the course after previous failures, and the average failure rate during these periods is 43%.

Table 5 – Historical series of failure rates and number of classes in the Programming Logic course (2009–2024).

Term	AVER	MAX	MIN	STD	Classes	Students	Stud. Min
2009.3	25	55	0	15	34	856	14
2010.1	31	48	16	15	4	107	25
2010.2	37	62	11	16	9	217	17
2011.1	29	88	0	22	47	1216	15

2011.3	42	69	24	14	11	262	13
2012.1	31	57	7	12	45	1206	18
2012.3	57	84	10	25	11	214	10
2013.1	30	83	4	16	39	1082	12
2013.2	56	92	14	29	12	215	12
2014.2	31	78	0	21	37	999	9
2014.3	36	61	8	18	10	340	17
2015.1	24	54	0	14	33	1085	17
2015.2	34	55	13	12	16	401	18
2015.3	45	58	29	11	6	145	23
2016.1	21	83	0	20	38	1104	15
2016.2	36	70	19	15	11	402	15
2016.3	43	45	41	3	2	243	55
2017.1	24	61	0	16	42	1298	12
2017.3	58	60	56	2	6	522	45
2018.1	26	79	4	16	69	2365	17
2018.2	52	94	25	17	18	466	16
2018.3	43	52	34	10	4	282	35
2019.1	37	81	0	21	84	2472	3
2019.2	43	49	31	11	3	122	36
2019.3	48	72	30	19	8	384	25
2020.1	39	93	3	25	75	2838	10
2021.3	36	61	19	11	34	1528	37
2022.2	24	48	0	12	44	1161	6
2022.3	42	78	15	23	9	229	15
2023.2	23	63	2	16	32	1247	29
2023.3	21	32	14	8	6	252	37
2024.1	26	76	3	20	34	1287	29
Ideal	**29**			**17**	**687**	**21744**	
Not Ideal	**43**			**14**	**146**	**4803**	
Total	**32**			**20**	**833**	**26547**	

The rows between 2020.1 and 2022.2 are highlighted in orange, suggesting a period of special interest or relevance that coincided with the COVID-19 pandemic. During this time, instruction was entirely remote, and assessments were conducted remotely, allowing students up to 72 hours to complete assignments. This remote learning environment and the extended deadlines likely influenced the observed learning and failure rates.

This table complements the results of Zampirolli *et al.* (2018), which compared CS1 (Programming Logic) in a blended learning environment with face-to-face assessments between 2016.3 and 2017.3.

The paper by Zampirolli *et al.* (2021) highlights the importance of using programming exercises with automatic grading, integrating Moodle, VPL, and MCTest, which was applied in the 2019.1 period. This material, along with the Colab notebooks, produced based on the book by Neves and Zampirolli (2017), proved to be very useful during the pandemic and continues to be utilized by many professors at the institution associated with this paper.

Finally, another important observation in this table is that the two ideal periods after the pandemic (2023.2 and 2024.1) had lower average failure rates (23% and 26%, respectively) compared to the overall average for the ideal period (29%). There is a general perception among professors, although this needs to be validated through a survey, for example, that instructors have become less demanding in their

assessments due to the learning gaps that arose during the pandemic. This hypothesis should be further investigated in future studies.

Threats to validity

This study is subject to certain limitations. Although the course enrollment in 2024.1 was 1,287 students, the proposed method was applied to only 72 students in two classes taught by the same instructor. Even using the same teaching materials as in previous editions (Zampirolli *et al.* 2021), a larger sample size could potentially yield more robust results.

Furthermore, the study is confined to the context of the institution associated with this paper, specifically within an interdisciplinary undergraduate program that includes students from various academic backgrounds in each class. This limitation affects the generalizability of the findings due to variations between classes in terms of programming languages, teaching styles, and the weight and difficulty of the academic activities assigned to students.

Since all items used in ATs were specifically designed for these two classes in the 2024.1 edition, the automatic calibration analysis conducted after test corrections and the examination of items that deviate from the established standard, as illustrated in this paper, necessitate a thorough review or elimination from the database.

Another limitation is the low response rate to the evaluation questionnaire (N=17, approximately 23.6% of the 72 students). This can be attributed to the fact that the survey was optional and administered at the very end of the academic quarter. At this stage, many students had already completed their course requirements and were on recess, while others were focused on final recovery exams, reducing engagement with voluntary activities. Consequently, the results reflect the perceptions of the most engaged students and may not fully represent the entire class.

Conclusion and future works

This study conducted during the first semester of 2024 in Programming Logic (CS1) involved two classes with a total of 72 enrolled students. Six tests were administered, five of which were ATs used as formative assessments in the open-source system MCTest. AT, including SAT, WPC, and MLE, provided personalized assessments based on student performance.

A final questionnaire with 17 respondents indicated that most students found ATs beneficial, enhancing their confidence and understanding of programming concepts. The results showed that adaptive methods offered personalized assessments, challenging students according to their skill levels, suggesting potential benefits for other educational contexts as an effective motivational strategy.

Further research with larger, diverse samples and control groups is essential for robust and generalizable conclusions about the effectiveness of ATs. With more data, it would also be important to analyze the differences between the three methods presented. It would be beneficial to develop a student module within MCTest to enable students to complete ATs directly within the system, thereby eliminating the need for printing and scanning the tests.

References

ALVES, Laura Filállepe et al. Continuous assessment, a teaching methodology for reducing retention and dropout rates in higher education calculus courses. **Revista Sítio Novo**, Palmas, v. 6, n. 4, p. 51-60, 2022. DOI: 10.47236/2594-7036.2022.v6.i4.51-60p.

ALVES, Lynn *et al.* Remote education: between illusion and reality. **Interfaces Científicas-Educação**, v. 8, n. 3, p. 348-365, 2020.

ALVES, Welington Domingos; SANTOS, Luiz Gustavo Fernandes dos. Playing with mathematics: using games to mediate the teaching and learning of mathematical content. **Revista Sítio Novo**, Palmas, v. 6, n. 4, p. 84-93, 2022. DOI: 10.47236/2594-7036.2022.v6.i4.84-93p.

BAKER, Frank B. *et al.* **The basics of item response theory using R**. v. 969. Springer, 2017.

BAYLARI, Ahmad; MONTAZER, Gh A. Design a personalized e-learning system based on item response theory and artificial neural network approach. **Expert Systems with Applications**, v. 36, n. 4, p. 8013-8021, 2009.

BECKER, Samantha Adams *et al.* **NMC horizon report: 2018 higher education edition**. Louisville, CO: Educause, 2018.

BINH, Hoang Tieu; DUY, Bui The. Student ability estimation based on IRT. In: **National Foundation For Science And Technology Development Conference On Information And Computer Science** (NICS), 3., 2016. [S. l.], 2016. p. 56-61.

CAI, Li *et al.* Item response theory. **Annual Review of Statistics and Its Application**, v. 3, p. 297-321, 2016.

CHEN, Keyu. **A comparison of fixed item parameter calibration methods and reporting score scales in the development of an item pool**. 2019. Tese (PhD) - University of Iowa, [S. l.], 2019.

CHOI, Younyoung; MCCLENEN, Cayce. **Development of adaptive formative assessment system using computerized adaptive testing and dynamic bayesian networks**. Applied Sciences, v. 10, n. 22, p. 8196, 2020.

COHEN, Jacob. **Statistical power analysis for the behavioral sciences**. 2. ed. Hillsdale, NJ: Erlbaum, 1988.

COSTA, Rebeca Soler *et al.* **Personalized and adaptive learning: educational practice and technological impact**. Texto Livre, v. 14, p. e33445, 2022.

GALVAO, Ailton Fonseca *et al.* **An intelligent model for item selection in computerized adaptive testing**. *2013. Dissertation (Master's) - Federal University of Juiz de Fora* 2013. (UFJF), Juiz de Fora, MG, 2013.

GHAVIFEKR, Simin; ROSDY, Wan Athirah Wan. Teaching and learning with technology: Effectiveness of ICT integration in schools. **International journal of research in education and science**, v. 1, n. 2, p. 175-191, 2015.

GROS, Begoña. The design of smart educational environments. **Smart learning environments**, v. 3, p. 1-11, 2016.

HAMDARE, S. An adaptive evaluation system to test student caliber using item response theory. **International Journal of Modern Trends in Engineering and Research**, v. 1, n. 5, p. 329-333, 2014.

HAMMOND, Flora *et al.* **Handbook for clinical research: design, statistics, and implementation**. Demos Medical Publishing, 2014.

HOLM, Sture. **A simple sequentially rejective multiple test procedure**. Scandinavian Journal of Statistics, v. 6, n. 2, p. 65-70, 1979.

JOHNSON, Amy M. *et al.* Challenges and solutions when using technologies in the classroom. In: **Adaptive educational technologies for literacy instruction**. Routledge, 2016. p. 13-30.

KARINO, Camila Akemi; SOUSA, Eduardo Carvalho. **Understanding your ENEM score - Participant's Guide**. [S. l.: s. n.], 2012.

KRATHWOHL, David R. A Revision of Bloom's Taxonomy: An Overview. **Theory Into Practice**, v. 41, n. 4, p. 212-218, 2002.

LAZARINIS, Fotis *et al.* **Creating personalized assessments based on learner knowledge and objectives in a hypermedia Web testing application**. Computers & Education, v. 55, n. 4, p. 1732-1743, 2010.

LORD, Frederic M. **Applications of item response theory to practical testing problems**. Routledge, 1980.

MEIJER, Rob R.; NERING, Michael L. Computerized adaptive testing: Overview and introduction. **Applied psychological measurement**, 1999.

MIN, Shangchao; ARYADOUST, Vahid. **A systematic review of item response theory in language assessment: Implications for the dimensionality of language ability**. Studies in Educational Evaluation, v. 68, p. 100963, 2021.

MORAN, José. **Hybrid education: a key concept for education today**. In: Hybrid teaching: personalization and technology in education. Porto Alegre: Penso, 2015.

MOREIRA, José António; SCHLEMMER, Eliane.
Towards a new concept and paradigm for onlife digital education. Revista uFG, v. 20, 2020.

NEVES, Rogério; ZAMPIROLLI, Francisco de Assis. **Processing information: a practical book on language-independent programming.** São Bernardo do Campo: EdUFABC, 2017.

OLIVEIRA, Plínio Cardoso de; SOUZA, Wallysonn Alves de; ALVES, José Robson Mariano. **Contesting: software to stimulate critical thinking, engagement, and promote autonomy in professional and technological education**. Revista Sítio Novo, Palmas, v. 9, p. e1611, 2025.

PARAMYTHIS, Alexandros; LOIDL-REISINGER, Susanne. Adaptive learning environments and e-learning standards. In: **European Conference On E-learning**, 2., 2003. [S. l.], 2003. p. 369-379.

PELLEGRINO, James W.; QUELLMALZ, Edys S. Perspectives on the integration of technology and assessment. **Journal of Research on Technology in Education**, v. 43, n. 2, p. 119-134, 2010.

PONTES, Paulo Ricardo da Silva; VICTOR, Valci Ferreira. Educational robotics: a practical approach to teaching programming logic. **Revista Sítio Novo,** Palmas, v. 6, n. 1, p. 57-71, 2022. DOI: 10.47236/2594-7036.2022.v6.i1.57-71p.

PUGLIESE, Lou. Adaptive learning systems: Surviving the storm. **Educause review**, v. 10, n. 7, 2016.

ROSENTHAL, Robert. **Meta-analytic procedures for social research**. Beverly Hills: Sage, 1984.

SHAPIRO, Samuel S.; WILK, Martin B. An analysis of variance test for normality (complete samples). **Biometrika**, v. 52, n. 3-4, p. 591-611, 1965.

SOARES, Ronald Ruan Pereira et al. **Development of a virtual assistant as academic support for the bachelor's degree program in Computer Science using customized generative AI and RAG**. Revista Sítio Novo, Palmas, v. 9, p. e1757, 2025.

TUKEY, John W. Box-and-whisker plots. In: **Exploratory data analysis**. [S. l.: s. n.], 1977. p. 39-43.

WAINER, Howard *et al.* **Computerized adaptive testing: A primer**. Lawrence Erlbaum Associates, Inc, 1990.

WANG, Feng-Hsu. Application of componential IRT model for diagnostic test in a standard-conformant eLearning system. In: **IEEE International Conference On Advanced Learning Technologies** (ICALT'06), 6., 2006. [S. l.], 2006. p. 237-241.

WATERS, John K. The great adaptive learning experiment. **Campus Technology**, v. 16, 2014.

WILCOXON, Frank. Individual comparisons by ranking methods. **Biometrics Bulletin**, v. 1, n. 6, p. 80-83, 1945.

YANG, Albert C. M. *et al.* Adaptive formative assessment system based on computerized adaptive testing and the learning memory cycle for personalized learning. **Computers and Education: Artificial Intelligence**, v. 3, p. 100104, 2022.

ZAMPIROLLI, Francisco de Assis. **MCTest: How to create and correct automatically parameterized exams.** Brazil: Independently Published, 2023.

ZAMPIROLLI, Francisco de Assis et al. An experience of automated assessment in a large-scale introduction programming course. **Computer Applications in Engineering Education**, p. 1284-1299, 2021.

ZAMPIROLLI, Francisco de Assis et al. Evaluation process for an introductory programming course using blended learning in engineering education. **Computer Applications in Engineering Education**, 2018. p. 1-13.

ZHENG, Yi. **New methods of online calibration for item bank replenishment**. 2014. Tese (PhD) - University of Illinois at Urbana-Champaign, [S. l.], 2014.

Additional information

Description		Declaration
Funding		None.
Ethical approval		Not applicable.
Conflicts of interest		None.
Availability of underlying research data		The content underlying the text of the manuscript is contained in this article.
Use of Artificial Intelligence		Yes. The use of AI was restricted to assisting with linguistic revision (spelling, grammar, and style corrections) of texts in Portuguese, Spanish, and English. The tool used was Google Gemini Pro. The procedure adopted consisted of inserting excerpts from the manuscript into the tool, using the command (prompt): “correct formal and scientific text: [original text]”. All suggestions generated by AI were critically analyzed and validated by the authors, ensuring the preservation of the original meaning. (drive.google.com/drive/folders/13tcXrV5DJ_gfDllVoDD2zUBXbxAXhtoH and github.com/fzampirolli/mctest.)
CrediT	Lucas Montagnani Calil Elias	Functions: conceptualization, data curation, formal analysis, investigation, methodology, programs, validation, writing — original draft, writing—review and editing.
	Francisco de Assis Zampirolli	Functions: conceptualization, data curation, formal analysis, investigation, methodology, programs, validation, writing—original draft, writing—review, editing, supervision, and project management

Reviewers: The reviewers opted for a closed evaluation and anonymity.
Reviewer of Portuguese text: Patrícia Luciano de Farias Teixeira Vidal.
Reviewer of English text: Patrícia Luciano de Farias Teixeira Vidal.
Reviewer of Spanish text: Graziani França Claudino de Anicézio.

How to cite:
ELIAS, Lucas Montagnani Calil; ZAMPIROLLI, Francisco de Assis. Adaptive Testing for programming logic skills assessment. **Revista Sítio Novo**, Palmas, v. 10, 2026. DOI: 10.47236/2594-7036.2026.v10.1905. Disponível em: https://sitionovo.ifto.edu.br/index.php/sitionovo/article/view/e1905.