Active Methods in Teaching Frameworks for Mobile Application Development: An Experience with High School Students

Douglas Oliveira de Jesus[1]*, Pâmela Michele Cândida Cortez[1]
[1]State University of Feira de Santana (UEFS); Feira de Santana, Bahia, Brazil

This article reports an experience focused on applying active methods in teaching frameworks for mobile application development involving high school students from the metropolitan region of Feira de Santana. The project was divided into four modules, and the Django and Flutter frameworks were utilized to create an application for an environmental preservation NGO. The active approach combined video lessons with collaborative practices, challenging students to apply concepts of system modeling and user interface development. Periodic questionnaires were administered to measure the method's impact on learning progress and technical and social skills development. The results highlight the effectiveness of active methods in fostering greater engagement, autonomy, and creativity in the learning process, as well as significantly improving students' technical competencies in software development.
Keywords: Active Methods. Technology Education. Computer Science Education. Educational Engagement.

Integrating active methods into teaching Information and Communication Technologies (ICT) has proven to be an effective approach for developing practical and cognitive skills in educational contexts. Since the early discussions by Papert [1], who advocated for using computers as teaching tools and as mediums to promote active learning, Computational Thinking has emerged as an essential 21st-century skill for problem-solving and creativity. Papert emphasizes that learning with technologies should be a creative process rather than mere passive absorption of information. Today, Computational Thinking is formally incorporated into Brazilian educational guidelines through the National Common Curricular Base (BNCC) [2], underscoring the need to develop students' digital and technological skills in basic education. However, despite these directives, implementing these practices in classrooms remains challenging. According to the ICT Education Report [3], 61% of teachers in Brazil reported difficulties in using digital technologies for pedagogical activities, highlighting a significant gap between curricular needs and educators' technical capabilities.

In response to this scenario, we developed an outreach program utilizing active methods to teach programming to high school students from public schools, offering two courses. For the vocational course on application development, the program was structured around the Django (https://www.djangoproject.com) and Flutter (https://flutter.dev) frameworks, aiming to teach topics ranging from object-oriented programming concepts to mobile interface development. To consolidate their learning, students were challenged to develop an actual application for an NGO, applying acquired concepts in a collaborative, problem-based project.This article examines the impact of this method, investigating how active learning and the development of a real-world project contributed to the student's technical and social formation.

Additionally, we present the main challenges encountered during the course and the strategies adopted to promote student persistence and academic success.

Received on 24 September 2024; revised 26 October 2024.
Address for correspondence: Douglas Oliveira de Jesus. v. Transnordestina, S/N - Novo Horizonte. Zipcode: 44.036.900. Feira de Santana, Bahia, Brazil. E-mail: dojesus@ecomp.uefs.br.

J Bioeng. Tech. Health 2024;7(Suppl 2):23-30

Materials and Methods

The course initially involved 32 students, offering a hands-on experience aimed at developing technical skills and fostering social and teamwork abilities, which are crucial for professional growth in the technology field. Throughout the project, periodic questionnaires were administered to assess learning progress and the impact of active methods—particularly Problem-Based Learning (PBL), a method already utilized in the Computer Engineering program at UEFS [4]—on student engagement and motivation.

The problem posed to the students was to develop a mobile application for the Eco da Mata Institute (IEM). Using the requirements provided, we applied PBL to address a real-world challenge with the following functionalities:

- **Event Management**: Centralizing the creation and dissemination of events related to ecological tourism and community fairs;
- **Project and Partner Promotion**: Showcasing sustainable services and products from traditional communities;
- **Geolocation of Map Points**: Highlighting tourist spots along the northern coast of Bahia;
- **Community Interaction**: Facilitating local community engagement and strengthening the connection between environmental preservation and tourism.

Active methodos, such as flipped classrooms, have gained prominence in recent years as a way to foster greater student engagement and active learning [5]. By reversing the traditional logic of knowledge transmission, this method allows students to study the material in advance and use classroom time for practical and collaborative activities, promoting the construction of new knowledge and its application to real-world situations [6]. In this context, in addition to problem-based learning, we employed the flipped classroom method for activity development. Educational materials were made available through recorded lessons (https://sites.google.com/view/jeditemple/desenvolvimento-mobile), and students were required to watch these asynchronous lessons before synchronous classes. This approach made the sessions more interactive and focused on clarifying doubts and engaging in collaborative discussions rather than mere content delivery. These meetings, held via videoconference, facilitated direct interaction between participants and the instructor, eliminating the need for physical travel, particularly for students from the Feira de Santana region.

The course was divided into four modules and spanned 34 weeks. Each module was designed to balance theoretical foundations with practical activities, promoting knowledge construction progressively and collaboratively. Accordingly, evaluation was conducted continuously and practically, with each team developing specific functionalities of the requested application.

Module 1: Object-Oriented Programming (OOP) and Introduction to Software Engineering Concepts

The course required prior knowledge of structured programming in Python, which we also taught to high school students from public schools through another course: Algorithms and Programming in Python (https://sites.google.com/view/jeditemple/algoritmos-e-programação). Accordingly, the first module, spanning 10 weeks, focused on Object-Oriented Programming (OOP) concepts, Design Patterns, and UML Diagrams.

During this period, students conducted seminars on the topics covered and completed practical Python activities to reinforce their understanding of the concepts. Assessment was based on their participation in the seminars and weekly practical activities.

Module 2: Understanding Requirements and Introduction to System Development

The second module, lasting six weeks, introduced students to more specific software development concepts. The PBL problem was presented, and

students began working on the planning and design of the mobile application. The main activities and approaches adopted in Module 2 were as follows:

Theoretical Classes: Video lessons covered Project Architecture, Database Modeling, Version Control Concepts, and the distinctions between back-end and front-end. Additionally, the Django and Flutter frameworks—used in system development—were introduced.

PBL Problem: Starting with this module, students began working on solving the proposed problem, organized into teams responsible for different application functionalities (event management, partners, community, and projects).

Synchronous Meetings: Synchronous sessions were increased to two weekly, each lasting two hours. One session focused on resolving questions related to theoretical content, while the other was dedicated to practical team-based work.

Team Activities: Students were divided into groups, with each team responsible for developing a specific part of the application, strengthening collaborative learning among participants [7]. Teams worked together to ensure the integration of functionalities and the system's overall functioning. During practical sessions, students created class diagrams, use case diagrams, and prototypes for the application based on the identified requirements. The project supervisor and coordinator provided the requirements specification.

Module 3: Back-End Development and API Creation

The third module, lasting eight weeks, focused on building the back-end and creating the API using the Django and Django REST frameworks. The goal of this module was to enable students to apply concepts such as data modeling, the MVC (Model-View-Controller) architectural pattern adapted in Django as MTV (Model-Template-View), and the development of RESTful APIs. The core features of Module 3 included:

Theoretical Classes: Students studied pre-recorded video lessons covering advanced topics such as data modeling, Object-Relational Mapping (ORM), and using the Django REST Framework to create APIs. Additionally, they were introduced to concepts like authentication, access control, and best practices for API security.

Synchronous Meetings: Following the structure established in the previous module, students participated in two weekly meetings, each lasting two hours. The first session addressed questions related to the video lessons, organized team progress, and set weekly goals. During these discussions, students structured their ideas, identified issues and facts, and defined short-term objectives, adhering to the "divide and conquer" principle [8].

Practical Activities: During the second weekly session, students worked in teams to implement the application's back-end. Using Django, they modeled the database, created functionality for managing events, projects, communities, and NGO partners, and developed a RESTful API to enable interaction with the front end. By the end of this module, a functional and well-documented API was delivered.

Module 4: Prototyping and Graphical User Interface Development

The fourth and final module, lasting 10 weeks, focused on prototyping the application and implementing the graphical user interface using the Flutter framework. This module aimed to consolidate the knowledge acquired throughout the course and provide practical experience integrating the front-end and back-end.

The key features of Module 4 included:

Theoretical Lessons: Students watched video lessons on interface prototyping using Figma and

mobile development with Flutter. Key concepts covered included reactivity in Flutter, using widgets to build modular and reusable interfaces, and integrating the front end with the API developed in the previous module.

Synchronous Meetings: As in Module 3, students participated in two weekly meetings of two hours each. The first meeting was dedicated to reviewing the concepts presented in the video lessons and planning interface development. The second meeting focused on practical implementation. Teams discussed short-term goals, aligned interface functionalities with the back end, and engaged in collaborative programming.

Practical Activities: During practical sessions, students used Figma to create application interface prototypes, simulating the user experience. They then implemented these prototypes in Flutter, connecting the front-end functionalities to the previously developed API.

Integration and Project Completion

By the end of the course, all established goals were achieved. The final project was presented to the NGO, delivering a comprehensive solution with a robust back-end, a well-structured API, and a responsive, intuitive graphical user interface. The PBL method, including fli, piped classrooms, and collaborative learning, provided a practical and enriching educational experience. This approach culminated in delivering a real product that met the NGO's needs and engaged the students in all stages of software system development.

Results and Discussions

The results from the final questionnaire applied to the students provide a comprehensive view of the impact of the active methods used in the course. They address aspects such as the participants' technical development, the course's influence on motivation to pursue a career in software, and an analysis of the attrition rates. Below, we discuss in detail each of these aspects based on the students' responses.

Analysis of Results on the Effectiveness of Active Methods

To understand the impact of active methods on the student's learning process, a questionnaire assessed different aspects of the educational experience. The responses comprehensively show how Active Learning has influenced academic progress and student motivation.

Among the questions asked, the one that stands out is: "On a scale of 1 to 5, how much do you think Active Learning has benefited you in achieving the course goals?" Most students (12 out of 19) gave the highest rating (5), indicating a significant positive impact. However, one student gave the lowest rating (2) and explained that Active Learning was not as beneficial for them. They mentioned: "I searched for solutions in many different places, which were often contradictory, and I got confused". Also, "I understand of certain topics was quite shallow." Therefore, while most students found value in the approach, not all could overcome difficulties interpreting varied information, failing to achieve autonomy in their studies. To make Active Learning more inclusive, it is advisable to provide clear guidelines and diverse learning resources (video lessons, manuals, synchronous classes, guided practical activities), adjusting the method to better meet the needs of all students.

Regarding the learning encouragement provided by the course, 13 students rated it a 5, 4 gave it a 4, and only one gave a lower rating (2). These data suggest that the course effectively encouraged students to seek knowledge autonomously, reflecting the effectiveness of active methods in maintaining student motivation. Additionally, the students highly valued the practical challenges. Most participants (13 out of 19) gave a 5 for the impact of using challenges on learning, and 15 considered the course challenging. The fact that no student gave a rating below 3 indicates that practical

challenges were widely accepted as a fundamental tool for learning and engagement, reinforcing the relevance of active methodos in the educational process.

The pre-recorded lessons received an average score of 3.8 regarding their contribution to the course content, with ratings ranging from 2 to 5. Six students gave the maximum rating, another six gave a 4, four chose a 3, and three gave a 2. The responses revealed that some students preferred to search for information online due to time constraints or the convenience of finding specific content rather than watching lengthy videos. One student mentioned that the theoretical lessons became less valuable as the course progressed. To improve, it is important to make the recorded lessons more accessible and adaptable to students' needs. For future classes, there are plans to break the lessons into subtopics and introduce a reference manual to facilitate access to information.

Based on the responses to the questionnaire, the course method, grounded in Active Learning practices, practical challenges, and recorded lessons, was well received by most students. The encouragement for learning and the relevance of the practical challenges were particularly praised. In contrast, the recorded lessons and the level of challenge in the course may be areas that require adjustments to better address the diverse needs and expectations of the students. This analysis reflects the importance of flexible methods, which promote autonomy and practice and can be adjusted to meet the different profiles of students.

Impact on Technical Development and Challenges Faced

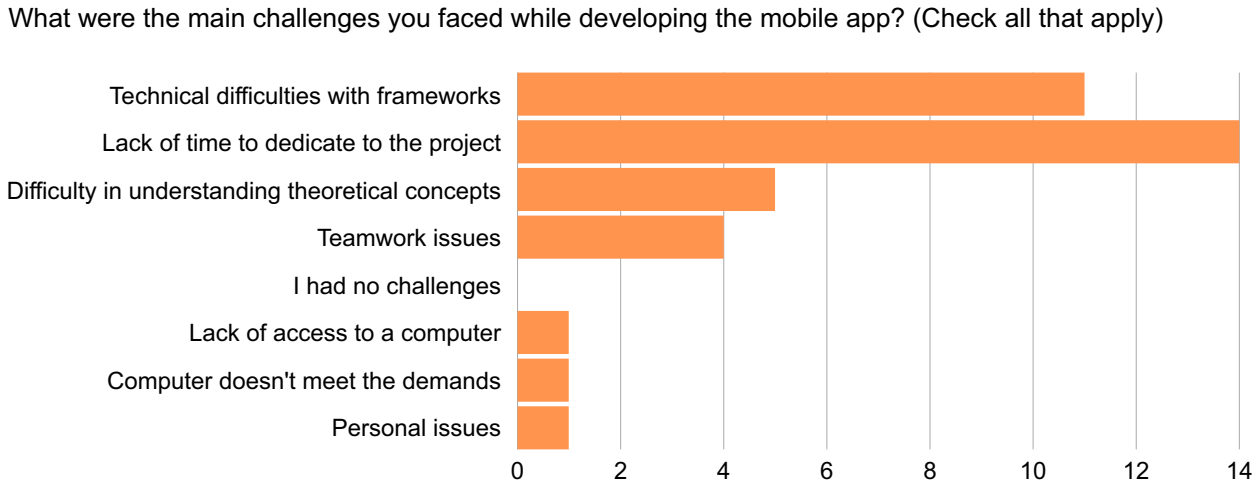
The results of the questionnaire on the use of the Django and Flutter frameworks and the development of a mobile application show that the course positively impacted the participants' technical knowledge. When asked about their understanding of these frameworks before and after the course, most participants (78.9%) stated that their understanding increased significantly. In comparison, 15.8% said their understanding increased moderately, and 5.3% reported a slight increase in knowledge. No student indicated that there was no change in their understanding, demonstrating that the course fulfilled its role in providing a solid foundation of technical knowledge.

Applying theoretical concepts was also highlighted as an important learning process. The question, "Did you manage to apply the theoretical concepts learned (such as object-oriented programming and system modeling) in the development of the application?" revealed that 31.6% of students were able to fully apply these concepts, while 52.6% applied them broadly. Only 10.5% said they applied them to a small extent, and 5.3% stated they applied them partially. These data suggest that the course provided an effective practical integration of theoretical concepts and the actual development of applications, which is crucial for the student's training.

However, when asked about the main challenges faced during application development, it became evident that technical and personal difficulties also influenced the process (Figure 1). 57.9% of the students mentioned technical difficulties with the frameworks, while 73.7% of participants identified the lack of time to dedicate to the project as the most significant obstacle. In addition, 26.3% of students mentioned difficulties in understanding theoretical concepts, and 21.1% reported problems with teamwork. No student stated that they did not face any challenges.

Two students also highlighted other factors that hindered their progress on the project: one mentioned the lack of access to a computer, and another reported that their equipment did not meet the minimum requirements necessary for application development. While the EAD modality has the advantage of opening opportunities for students who live far from the university, it simultaneously exposes social inequalities that could only be overcome with adequate infrastructure in computer labs in all public schools, accessible to students throughout the day, just like school libraries.

Figure 1. The main challenges encountered by the students.



Impact of the Course on Motivation for Careers in Technology

The survey results highlight the significant impact of the Mobile Development course on students' motivation to pursue careers in software development and, more broadly, in the technology field. The question "Did the development of this project increase your motivation to pursue a career in software development?" received overwhelmingly positive responses, with 16 students answering yes and 3 responding "maybe." No students expressed a lack of motivation.

Additionally, the question about the motivation of high school students who had not yet started higher education revealed that 87.5% expressed interest in pursuing a technology degree, while 12.5% answered "maybe." Several students provided detailed justifications for why they felt motivated to enter higher education in technology. One student stated that "the course sparked a desire to learn more about the field," mentioning plans to enroll in the Computer Engineering program at the State University of Feira de Santana (UEFS). Furthermore, the hands-on experiences provided by the course were highlighted as essential in shaping the students' aspirations. One participant commented, "The course made me realize that it is possible to acquire the necessary knowledge by practicing and studying, which motivated me to enter university."

In summary, the course developed technical skills and significantly influenced students' academic and professional aspirations. Exposure to real-world software development challenges and the collaborative resolution of problems motivated many students to consider continuing their studies in technology-related fields. These results highlight the effectiveness of vocational courses in inspiring young people to pursue higher education, further emphasizing the importance of practical learning experiences in shaping their careers.

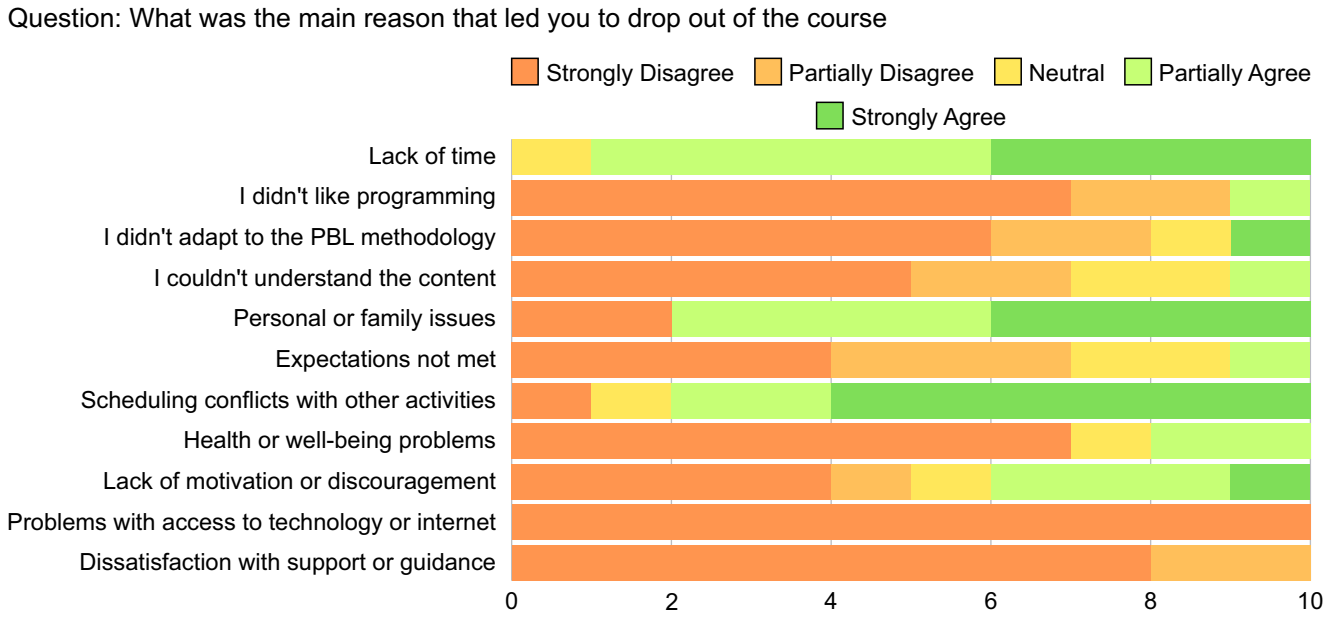
Contributing Factors to Course Dropout

The analysis of the responses from the questionnaire applied to students who dropped out of the programming course revealed a series of factors that contributed to their decision.

The primary reason identified was lack of time, mentioned by 90% of the participants, who agreed partially or entirely with this reason, as illustrated in Figure 2. Many students reported that professional and personal responsibilities, such as work and family commitments, made participating regularly in the course meetings and activities difficult.

In addition to the lack of time, difficulty understanding the content was another significant

Figure 2. Reasons for dropping out of the course.



factor for the dropout. Some students mentioned that they could not keep up with the course's pace, with one commenting that "it felt like everyone was already ahead." This sense of inadequacy may have been exacerbated by the active learning method, which requires greater autonomy from students and may not have been well-received by those with little prior experience in programming.

Another recurring factor was schedule incompatibility. One student reported that, although the meeting times were collectively defined, "changes in personal life" made it difficult to participate in synchronous activities. The lack of flexibility in the schedule was seen as an obstacle for those facing unforeseen circumstances or external commitments, suggesting the need to offer more scheduling options or alternative meetings. Despite the challenges, some students praised the course structure, describing it as "great" or "very good." This indicates that the dropouts were more related to external and contextual factors than dissatisfaction with the course content or method.

Some strategies could be implemented to address these issues and reduce dropout rates. Creating more accessible initial modules aimed at leveling students' knowledge with varying levels of experience could help improve course engagement. Offering alternative synchronous meetings or flexible schedules could better accommodate students' external demands. Additionally, more monitors could provide personalized support, addressing students' diverse needs and learning paces. These measures could significantly increase retention and student success in the course.

Conclusion

This paper presented the results of an educational experiment focused on applying active methods in teaching frameworks for mobile app development, emphasizing the use of Django and Flutter. The initiative involved high school students from the metropolitan area of Feira de Santana, aiming to develop technical skills and promote greater engagement with the technology field. The experiment also evaluated the impact of these methods on students' motivation toward higher education.

The results demonstrated that the adopted approach effectively enhanced the students' technical knowledge. They reported significant improvement in understanding the frameworks and their practical

application in software development. Furthermore, the surveys indicated that the course stimulated motivation for careers in the technology sector, with many students considering pursuing higher education at local institutions such as UEFS and UFBA.

However, technical difficulties, lack of time to dedicate to the project, personal issues, and lack of access to adequate equipment were mentioned. These factors highlight the need for more robust infrastructure and technical guidance support to ensure that all students can fully participate in the learning process.

We concluded that applying active methods, such as project-based learning and integrating video lessons and practical activities, is beneficial in the context of technological education. The course not only improved the students' technical knowledge but also fostered interest in careers in the software field, demonstrating the potential of these initiatives to train and guide young talent. Finally, it is recommended that future projects consider expanding technical support and flexibility schedules to better accommodate students' needs, ensuring that all can overcome the identified challenges and make the most of the opportunities offered by the course.

References

1. Papert SM. Mindstorms: Children, Computers and Powerful Ideas, Basic Books, New York, 1980.
2. Brazil. National Common Curricular Base, Ministry of Education, Brasília, 2018.
3. Núcleo de Informação e Coordenação do Ponto BR (NIC.br), Research on the Use of Information and Communication Technologies in Brazilian Schools, 2020. [Online]. Available at: https://cetic.br/media/docs/publicacoes/2/20211124200326/tic_educacao_2020_livro_eletronico.pdf. [Accessed: January 20, 2025].
4. Barros FLF, Bittencourt RA. Evaluating the influence of PBL on the development of soft skills in a computer engineering undergraduate program. Proceedings of the 48th Annual Frontiers in Education Conference, IEEE 2018.
5. DeLozier SJ, Rhodes MG. Flipped classrooms: A review of key ideas and recommendations for practice. Educational Psychology Review 2017;29(1):141–151.
6. Zhang L. Blended pedagogy for computer programming language. Pedagogical Research 2024;9(3), em0218. https://doi.org/10.29333/pr/14744
7. Hawlitschek A, Berndt S, Schulz S. Empirical research on pair programming in higher education: A literature review. Computer Science Education 2022;33(3):400–428.
8. Paula MC, Viali L, Rizzon EC. Historical context of the divide-and-conquer method and an approach to studying a self-problem in a real-world issue. Vetor - Journal of Exact and Engineering Sciences 2021.