Applying PBL in Teaching Programming: an Experience Report

Simone C. dos Santos
Information and Systems Department
Centro de Informática - UFPE
Recife, Brazil
scs@cin.ufpe.br

Éden Santana, Lucas Santana, Pedro Rossi, Lucas Cardoso,
Ullayne Fernandes, Cláudio Carvalho and Pedro Tôrres
Computer Science Department
Centro de Informática - UFPE
Recife, Brazil
{eeas, lss5, pgrr, lccao, uffl, cco2, phts}@cin.ufpe.br

***Abstract*— Tutorial Education Program (TEP) is an initiative from Brazilian government to develop education quality in different areas of knowledge, as its main aim to decrease students' evasion and promotion of academic success. PET-Informática group from Federal University of Pernambuco (UFPE) has invested in these activities, in particular, to teach programming, one of the disciplines in which students have lower grades in technology courses. In this context, this paper describes an experience of an extension course in Python, using the active learning approach Problem-Based Learning (PBL). This course, ministered in two weeks, made possible the construction of a PBL model to teach Python, based in the *xPBL methodology* described by Santos & Rodrigues (2016) and the lessons learned with this practical experiment. As the main contribution, this experience stands out the possibility to apply this course to different audiences with different levels of education, and also the dissemination of PBL approach in teaching programming.**

Keywords—Problem-Based Learning, Software Programming, Computing Education

I. INTRODUCTION

Brazilian government is continuously investing into initiatives to promote the development of teaching quality in their universities distributed across the country. Among these initiatives, it highlights the Tutorial Education Program (TEP), with the aim of developing standards of quality and academic excellence, highlighting the articulation of teaching, research and extension. This program aims to decrease the dropout of students and promote academic success. In its implementation, with activities directed at different audiences, internal and external to the university. This program is conducted by groups of undergraduates in specific departments and centers of a public university, in which each group is guided by a tutor. In the Informatics Center of the Federal University of Pernambuco (UFPE), the activities of the TEP group (referred as TEPI) are performed by a group of 13 contributors in three major projects: 1) *Educational Innovations in Computing*; 2) *Technology Supporting Education*; 3) *Audience attraction to computing*. This article reports an experience of TEPI group in the project 1, focusing on educational programming, one of the biggest challenges of Computer Science [1], [2].

In [1], the authors point to learn programming is not an easy task. It requires a correct understanding of abstract concepts, logical reasoning, domain of structures and understanding of the syntax of complex languages.

The study [1] recommended various teaching strategies to tackle the challenges in teaching programming in five categories: unplugged type activities, contextualized tasks, collaborative learning, development of computational thinking and programming tasks supported by scaffolding techniques. All these strategies are related to active teaching and learning approaches [3], and can be combined in order to not only promote the involvement of students as to deepen their knowledge and understanding of the area's issues.

In this context, this study proposes a response to the following research question: *"how to improve teaching programming from the use of active learning approaches?*".

In response to this question was created and run an extension course Programming in Python, using the active approach "*Problem-Based Learning*" or simply PBL. First used in the medical field in the 60's, PBL is becoming increasingly effective in a variety of disciplines in higher education, with the emphasis in engineering and more recently in computing education [4]. PBL is defined as a student centered approach, which uses one or more problems as learning tool, promoting technical skills and nontechnical to solve them, through teamwork and collaboration. The motivation for the use of PBL departed from its alignment with education aimed at the practice, such as software programming, and successful results obtained in computer education, such as presented in [4], [5] [6], [7], [8]. In [4], the authors systematically map the use of PBL from 1997 to 2011 to teach computing, and the authors analyzed 52 studies in this field, taken from sources such as IEEE, Scopus, Science Direct and ACM databases. Despite the potential benefits of the PBL approach, it is also evident that its use implies a profound educational change and a new culture of teaching and learning, thus considered innovative by the most of educational institution when teaching computing.

The choice of Python is justified by its use in high visibility applications such as Google, Instagram and YouTube in addition to its growing popularity [9], [10]. The course is taught in two classes: the first, with high school students and; the second, with college students in different areas, technical and non-technical. The PBL approach used to refer to PBL methodology for computer education defined in [11], which

978-1-5386-1174-6/18/$31.00 ©2018 IEEE

proposes the application of PBL from five key elements: *Problems, Environment, Human Capital, Content* and assessment *Process*. The problems were brought by the students themselves, from experiences in their communities. Each group was supported by three teachers in the role of tutors, in which students were organized into teams 4-5 members. Content support was made available to students at the slideshow format. The results were collected based on diagnostic assessments, knowledge, personal characteristics and a final project developed by the students. Analyzing these results, the ownership of the process of solving problems was better on behalf of theof high school students than of the college students, mainly due to the technical profile of the first class was perceived. After analyzing the results, it was possible to define a model for programming course replication in the Python language in the PBL approach, based on lessons learned in the course, containing guidelines and templates for its re-application.

This paper is organized into six sections. After this introduction, Section II describes the main theoretical frameworks used in the design and execution of this work. Section III describes the method of action research that has been chosen to perform this research. Section IV reports all the work from planning to the analysis of results, including implementation and verification steps. Section V proposes a model for teaching Python programming and, finally, Section VI presents conclusions of this study.

II. THEORETICAL REFERENCES

A. Challenges in Programming Education

Currently, the programming teaching widely used in schools and universities is carried out by a conventional methodology involving about 6 hours of lessons, lists of exercises, helping tutors in related questions issues in class to solve exercises and summative tests as assessments. However, when using this method of teaching it is possible to observe some challenges, especially for students who are just learning programming logic [1]. Among the main challenges stand out the followings: Use of the programming development environment; Get access to computers / network; Understand the programming structures; Learn the syntax of the programming language; Create a program to solve a given task; Split functionality in procedures, and finally; Find errors in the program. Faced by these challenges, many students have learning difficulties on this matter. Moreover, the traditional teaching model with the presentation of a lot of content, low diversity of exercises and few relevant practices, often conducted by the teacher, eventually discourage students, resulting in high dropout rates of programming courses [1].

In [2], Sentance and Csizmadia (2017) investigate the challenges of education in programming through a survey with 300 teachers. From this study are pointed out two categories of challenges: *Intrinsic,* such as knowledge, skills and attitudes of students, in addition to, and the teaching and learning approach; *Extrinsic,* such as lack of resources, time management, the practice of programming and student assessment process.

The teaching of programming is naturally very challenging, overcoming technical skills, because it is necessary to teach skills which are not usually developed in conventional teaching methodology. It is challenging for both those who are teaching, it is necessary to stimulate the student to develop skills such as teamwork and problem solving, as for the student, when it is evaluated by skills that are not used to and in face of poor performance, the student gets discouraged.

In [12], the authors discusses some conficts: "*To make this concrete, consider the role of group and individual work in typical foundation courses. The courses are based upon individual typically work. By contrast, much of the workforce operates in programming teams, where the ability to work cooperatively and to communicate well are important. Students with aptitude and a preference for teamwork may give up on Computer Science if their first year experiences convince them that the discipline is individualistic*".

B. Learning Approaches in Teaching Programming.

The study described in [13] highlights diverse approaches to teach programming: seminars classes based on exhibitions and reading materials and/or slides, with practical activities inside laboratories to apply what has been learned previously; visualization software for mapping programming concepts into more abstract ideas, so they can be easily illustrated as running a program step by step; problem-based learning (PBL) and; other approaches that comes from different ways of visualizing content and changes in the current educational paradigms.

The use of practical seminars and classes in laboratories have as positive factors the fact that they promote a guided learning process with reinforcement for beginners. It also provides flexibility for the teacher, as the human cost for the execution of the classes is low compared to other approaches. But this method fails on the retention of long-term knowledge by students, when you need to develop skills of higher cognitive demand such as analysis, synthesis and systems evaluation, according to the reference [13] "*Traditional teaching approaches imposes the rule based approach on the natural learner and restrict their problem solving skills*".

Visualization software came as a solution to programming teaching based in the reference "*The Theory of Multiple Intelligences state que humans are born with a Certain amount of intelligence, with specifics intelligences being dominant and others recessive and the potential to Develop all intelligences being possible*" [13]. Researches show that spatial intelligence, the ability to form a mental model of the world around, objects or patterns and handle them, ends up being an important factor in predicting the successful execution of a program. Among the advantages of different categories of visualization software, it is possible to highlight: students engagement; availability of many tools on the internet for this purpose; students can easily interact with such tools and; the tools make it easy to debug and learn programming through a visual step by step [13].

C. The PBL approach and xPBL Methodology

The Problem-based Learning can be defined as an instructional teaching method focused on the student, where they are immersed in real projects practices, work in teams in order to solve problems systematically, being co-responsible for his learning [14]. PBL is based on principles [15], however, it does not have a single methodology for applying these principles in the educational context. Considering the context of programming teaching, the authors of this paper chose to use a PBL methodology created for computing education: "xPBL". The xPBL methodology [11] aims to align methods and management tools for PBL approach, so that the principles that characterize can be guaranteed during its adoption in educational computing. In [16], the authors set out 10 principles (see Table I) from a thorough study of principles and features about PBL found in Savery & Duffy (1995) [15], Barrows (2001) [17], Peterson (1997) [18] and Alessio (2004) [14]. To ensure the principles of PBL, the xPBL is based on five elements: (1) Problems involving the capture of real problems, and complexity as relevant aspect; (2) Learning Environment, to reflect the characteristics of the professional environment of software; (3) Content, flexible and aligned to problem solving; (4) Human Capital, involving teachers and tutors as experienced professionals and PBL mentors, students as the development team, and actual customers as stakeholders and; (5) Assessment Processes based on different dimensions and continuous feedback. Table I shows the mapping between the ten principles and elements of xPBL.

TABLE I. ELEMENTS OF THE XPBL AND PBL PRINCIPLES.

xPBL Elements	*PBL Principles*
Problem	1, 2, 3, and 6
Environment	4
Content	5 and 7
Human Capital	9
Process	8 and 10

Four of the principles defined in [16] are related to the element "Problem" of the xPBL. These principles reinforce that in the PBL, all tasks and activities are anchored in trouble, not simply content exposure. These problems need to be real and complex enough to stimulate interest, motivation and reasoning of the students in building their knowledge. In addition, the student needs to feel an owner of the problem, establishing the process of how to solve it.

Regarding the "Environment" element of the xPBL, the principle 4 highlights the need for closer ties between the learning environment and the working environment, so that students can experience real situations, immersed in performed collaboratively practices and activities.

The principles 5 and 7 refer to the element "Content" of the xPBL, reflecting the need to run the practices from content (either structure, theory, foundation, model or concept). Although learning is essentially functional, PBL methodology considers essential to support content and conceptual basis for solutions to problems and can not be mistaken for practical experiments which prevail low theoretical support and the lack of monitoring of resolution processes.

Regarding the "Human Capital", the principle 9 reinforces the idea of a collaborative and multilearning promoted by the constant engagement between students, teachers, tutors and clients during the process of resolution.

Finally, the principles 8 and 10 associated with the element process emphasize the need to plan and monitor continuously the processes of teaching and learning through reflection of learning by parts of students and conducting ongoing evaluations.

For the application of xPBL, the authors of [11] are based on the steps of PDCA cycle by Deming [19], enabling that the planning is carried aligned with its execution. With the use of the 5W2H management technique, guidelines are set for each element of xPBL, allowing decompose, analyze and summarize activities associated with each during planning. Objectively, information about what and how it should be done, or who and when will conduct a certain activity are considered in the definition of each element in these guidelines.

Despite the xPBL contribution and its guidelines, the authors of this work realized in their case studies that needed a tool that facilitates collaborative planning of the course in the PBL approach, allowing for better interaction between all involved, whether teachers, tutors or coordinators. In this context, the work in [20] proposed the construction of a tool based on canvas, called *PBL Planner Toolkit*. This toolkit consists of a canvas organized in 11 fields (as illustrated in Fig. 1), and 40 instructional cards that guide its fill.

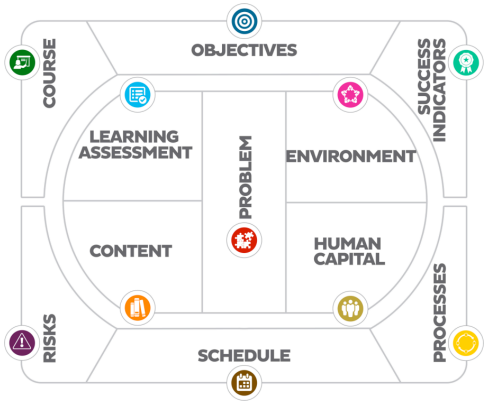

Fig. 1. PBL Planner Toolkit.

With this tool the pedagogical team, consisting of teachers, tutors and coordinators, elaborates the plan in PBL approach in three steps: 1) Planning, filling the fields with the help of letters 2) review, complementing the outstanding issues and resolving doubts and planning; 3) Sharing plan prepared with all involved.

III. RESEARCH METHOD

The authors of this study decided to adopt qualitative research, with an emphasis on action-research. According to Patton in [21], a research is said qualitative when aims to

investigate what people do, know, think and feel through data collection techniques such as observation, interviews, questionnaires, document analysis, among others. According to Merriam & Tisdell in [22], action-research is intended to solve a problem in practice, contributing to the research process itself. Its goal is to address a specific problem in a real environment such as a classroom, workplace, a program or an organization. Action-research is often conducted by people who are interested in facilitating change in their work, community or family. For this, you decide to "experiment" the situation, while documenting what happens and trying a new strategy or intervention. Typically, many interventions or strategies are implemented by the participants throughout the trial. The results and the unfolding process are continuously documented, making evident the process of finding most effective solutions in practice based problems.

This approach allowed the realization of an analysis of the study object based on the understanding of the perception of the actors involved in running a course of programming in Python, taught in two classes of different profiles: one with high school students and one with college students from different areas of knowledge. This research also had an exploratory character, when aimed discover the main challenges for the conduct teaching programming in real environment, from the perspective of students and tutors in order to find out what is happening and developing strategies to combat these challenges, and motivate future research. The case study was chosen as the appropriate research method for this research, because it is an empirical research that analyzes a contemporary phenomenon in its real life context, especially when this phenomenon is not clearly defined [23].

Fig.2 illustrates the steps of the research, given its methodological character.

Fig. 2. Steps of this research.

IV. CASE STUDY

This section describes the planning, execution, evaluation and identification of strengths and improvements in the programming courses conducted.

The aim of the summer school can be divided into two: disseminate knowledge of programming for people from various fields and introduce new potential IT students to practice.

With that in mind, it was decided to divide the classes, according to their purpose, in a class with high school students (T1) and a class with college students (T2), both with up to 20 students each. This course is historically performed in January by TEPI group, during the vacation period of the students school year, in order to have intensive classes with exclusive dedication.

The classes took place in two weeks, lasting four hours, totaling 40 hours. It was taught in the laboratories of the Informatics Center in the afternoon, by a team of six tutors, 3 tutors per class. Tutors are active members of the TEPI group and students of the graduation course of Computer Science from the periods of 3º to 7º from a total of 10 periods.

A. Plan

The planning of the course took place over the period from November 2017 to January 2018, and used as management tool *PBL Canvas Toolkit*. The tool helped in preparing the educational plan for the implementation of xPBL methodology in the course, the two groups T1 and T2. This course was conducted intensively over two weeks with 40 hours.

Due to the difference between the two groups proposed the planning was different in some points, the main difference was to decide which problem would be selected.

For the T2 class, due to a more specific demand of a programming knowledge, it was decided that students indicate problems of their areas in order to increase the ownership over them. The problems were indicated by each student, analyzed by the team of tutors, and voted in rounds, eliminating the least voted and forming groups of five. The analysis of the problems considered compliance with the planned course menu and its feasibility as first contact with the computer

Due to the lack of a specific area of expertise, for the T1 class in addition to the problems brought by the students, a pool of problems was created, considering the possible difficulty for the students to identify problems that could be solved computationally.

Due to the intensive nature of the course, students have no time to heavily research the issues. In this scenario, there is a need for early lectures in order to introduce basic concepts of programming. Lest there was a disconnection between solution ideation and the content necessary for its realization, it was decided that all the first week classes would be shared between: the introduction of a concept; a period of mentoring, with an analysis of the necessity of this concept for the proposed solution; the implementation of the solution. The second week would be dedicated exclusively to solve the problem and possible need for adjustments in the proposed solution.

The evaluation process was defined, considering three aspects: theoretical evaluation (content), evaluation of the proposed solution (result) and evaluation of the interpersonal skills (360 degree members of the performance of each group). The theoretical evaluation should be performed at the end of the course, the same day the solution presentation. The developed solution should be presented to the group of tutors who evaluate according to the technical criteria of implementation and have succeeded in solving the proposed problem. In a 360 degree assessment, all students evaluate the cooperation of his or her teammates in the project at the end of each week.

B. Do

the first day of the course was run a brief explanation of how the course would be conducted and what the teaching approach and learning used in addition of an introduction to the programming environment that would be used. This day was also asked students to indicate problems that would have desire to work.

In the class T1, where there was also a pool of support problems, many students had similar ideas and thus gathered easily with small interventions of tutors to make similar ideas converge. Some students also choose pool problems.

Students of the T2 class were from several areas, such as electrical engineering, medicine, biology and statistics. So the problems posed by they needed more intervention from the tutors, as some were too complex to be resolved within the scope of the course, and others were not challenging enough to stimulate the need for learning desired by the course.

With organized teams and the program in progress, some difficulties were observed. The first identified was the difficulty in writing a program. To encourage the practice of programming, it proposed a list of optional exercises.

During the first week, there have been difficulties in implementing one of the T2 class groups because of the choice of the theme. The problem selected did not fit in easily to the proposal of the course, which generated a unmotivation to group members, who wanted to change the subject. As the difficulty was also identified earlier in the week, after a meeting of the group members with the tutor, they agreed to use one of the problems posed in the pool problems presented to the T1 class, and the execution of the project continued without further problems.

In general, the execution of T1 class course was more aligned to their planning than the T2 class. In the T1 class also observed a better engagement and also a better dissemination of knowledge.

The main difficulty identified in the T2 class was the interaction between members of the same group, because of the great heterogeneity of knowledge of programming between them. The group that best exemplifies this challenge was composed of students of electrical engineering, computer engineering, medicine and biology. The engineering students were already familiar with programming therefore already had more ease with the subject and with the implementation of programs, while this course was the first contact of the students in the health field with computing. This meant that those who had wanted to ease move faster with the development of the solution than the others could, generating a sense of frustration on both sides. The team of tutors, to observe the situation, intervened so that everyone could participate in the solution, ensuring the generation of learning for all, explaining the importance of collaboration and teamwork. This intervention, performed well, despite the engagement fall with the course.

360 degree assessments evaluating the proximity of the tutors have proven being essential to correct the problems that arose. The theoretical evaluation and the project were successfully implemented.

C. Check

In assessing the *theoretical content* learned by students through test on programming content taught in the course, it was possible to obtain slightly satisfactory results as to class performance, reaching an average of 6.97 in the T1 class; and 7.47 in the T2 class.

In the evaluation of the *projects* developed by the students, it was obtained even better results, with an average of 9.4 in T1 and 8.6 in T2 class, very similar to *360 evalution* with an average of 9.2 in T1 and 9.0 in T2 class. It was also evaluated the individual performance of each student in the *presentation* of their projects, in order to observe its grip on the process of resolving the problem and understanding the knowledge not applied from this assessment, discrepancies were observed that could affect their significantly assessments. Fig. 3 (T1 class) and Fig. 4 (T2 class) show the students evaluations and their all grades, considering the four aspects mentioned above, into a 0 to 10 scale.

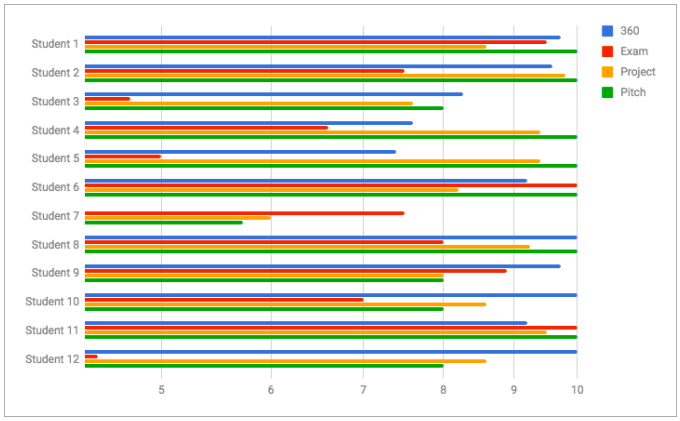

Fig. 3. Performance in T1 class (high school students).

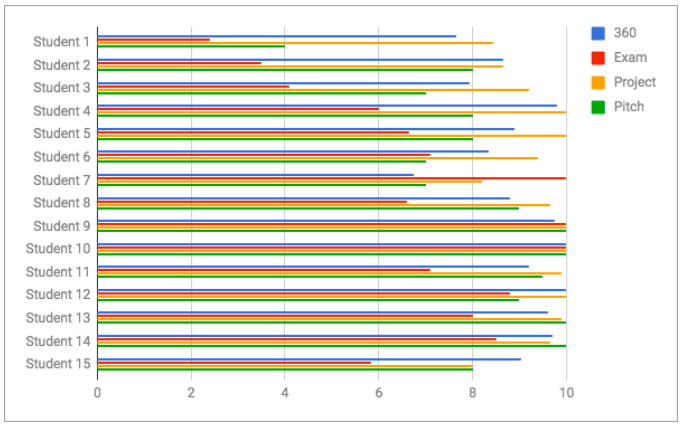

Fig. 4. Performance in T2 class (college students).

It was also carried out an evaluation with our students on PBL approach applied in the discipline and how they felt about the same, according to the proposal described in [16]. With the collection of data from this survey, it was able to conclude that the manner in which the model was applied

satisfied the goal of instigating learning through the integration of students into a real problem that captivated engagement of same in the task to be accomplished. It was perceived, however, that still fail to reach some goals in our PBL model. There is still reservations of students in relation to the learning environment, which did not resembled a real situation, and the choice of the theme for the project, which some found rather too influenced by the tutors.

In all aspects of our assessment of the applied teaching model, the T2 class demonstrated in virtually every aspect less immersion within the PBL methodology when compared to the T1 class. Possible reasons for this discrepancy will be discussed later.

Overall, the course was successful, with a total approved rate higher than 90% for graduates and delighted by the students, who were motivated to continue learning about the given subject, even after the end of the course. Some even contacted their guardians after months to take new doubts and ask for suggestions of materials to give continuity to their learning. Fig. 5 and Fig. 6 show the final results, considering the approval average was set to 7.0.

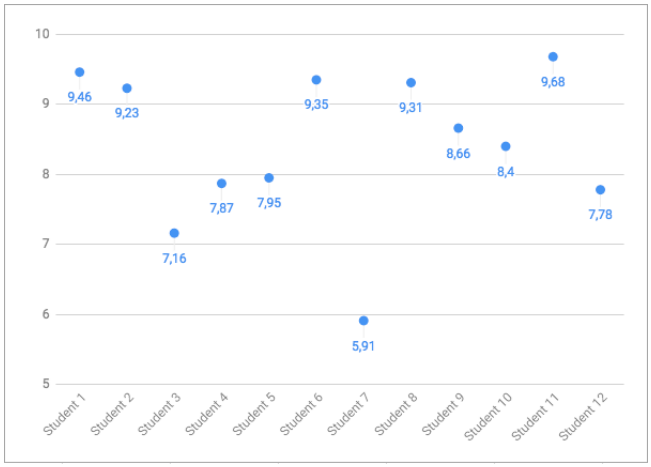

Fig. 5. Final grades in T1 class (high school students).

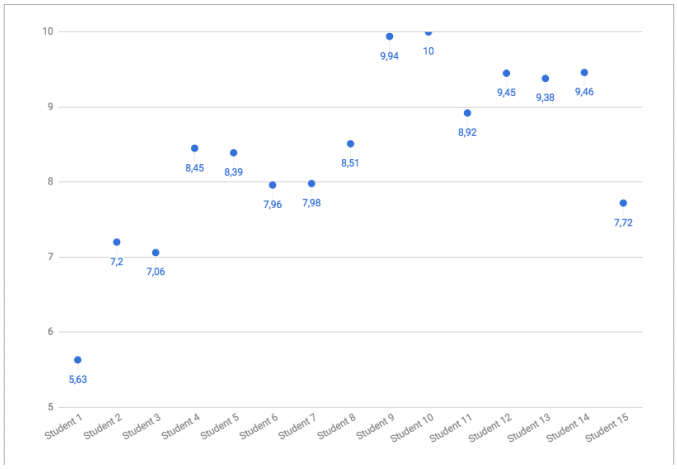

Fig. 6. Final grades in T2 class (college students).

D. Act

As a strong point, it was realized that the large exposure of students to practical and real problems in the programming area, allowed that class had a good performance, thanks to the implementation of the PBL approach. The theoretical aspect of this area, despite not having made a necessarily negative, it was shown as a knowledge less absorbed by the students. Importantly, by proposing practical aligned content it is common to identify the limitations in the assimilation of concepts by the students, leaving more evident a major extrinsic challenges in educational programming, as the balance between theory and time management. Moreover, it was also perceived the difficulty of transforming a teaching process conducted by the teacher, as in formal education, for a learning process focused on the student, who must be active in search and search for knowledge. During the course, some students expressed the desire for more content from classes and exhibition to remedy his or her needs.

A major difficulty to overcome is mainly presented in the T2 class, where there was a greater amount of students with little desire to actually learn the subject, merely present the prospect of obtaining certification of the course, in contrast to our T1 class where general students were more interested in the subject taught.

It was realized that without a great motivation for the subject and real willingness to learn, it is difficult to get students actually to enjoy the PBL characteristics that, often depend on proactivity. Such difficulties were managed with close monitoring of tutors in some groups and even changes in the compositions of some teams to adapt them to project interests of participants. Unfortunately, not all problems could be solved, which generated a T2 class dropout rate 8,97% higher than the T1 class.

V. Proposed Model for Programming Education in Python

In this section, based on program execution and analysis of the results obtained, a model is proposed for the intensive program teaching using xPBL split in five elements, namely: (A) room preparation, physical and virtual; (B) the selection of the problems to be worked along the course; (C) formation of the teams; (D) choice of content and; (E) choice of tools and application of the evaluation model.

A. Environment

Learning environment should be separated into two categories: online and offline. For the first, it is a virtual learning environment, which for this edition of the course, it was used Google Classroom. Through the virtual environment, the classes and tutors can exchange information about the issues and the availability of presentation slides used in class and lists of exercises. Each team should also create a virtual group, where team members can have a closer contact with their tutors and each other.

For classroom activities, laboratories are used where each member has access to a computer connected to the internet with the necessary tools for the development of activities. It was used the IDE JetBrains PyCharm. Groups should also have at their disposal whiteboard, whiteboard pencils, paper and pencil for discussion of ideas and design solutions.

For the Introduction of the course, presentation of the problem and presentation of final projects at the end of the course is interesting the use of auditoriums where all members can attend the presentations of other teams in their class. Due to the time of presentation of each project became unviable the junction of two groups at this time.

B. Problem

Given the intensive nature of the course, it is essential that the first day of class is asked to the students too seek in their daily lives or in their areas of interest, issues that are possible to solve with programming. Considering that the search time students do not bring many problems, it is suggested that tutors prepare together a pool of problems that students can choose from.

Problems can be of several types, provided they are feasible in a short time and have a difficulty level compatible with the team. In addition, tutors should jointly assess whether the problems brought by the students include the topics covered throughout the course and the application should contribute to the construction of knowledge.

If students bring problems that tutors do not cover the area, is asked the student to explain to the tutors in detail so that they can analyze its feasibility within the course scope. Otherwise, the body of tutors and students try to bring a similar problem or the student should look for another problem that he or she can also be interested in.

C. Human Capital

The formation of the teams should be based on the interest of each of the students for the problems that are brought or chosen (from the pool of problems for high school students) themselves, by a vote where each one choose one or two options, depending on the number of participants and tutors, labor interests. Options with a low number of votes are deleted and a new round of votes is done, until it reaches the number of options equal to number of teams. It is suggested that each team has, in average, four students, but small variations in the size of the team generated no significant difference in learning.

It was observed that, dividing using this method, improves the engagement of the participants, as problems in unattractive areas end up being chosen by few or no students and are excluded from the set of possibilities. For problems that end up attracting a lot of students, it is suggested that can be divided by the team of tutors in two or more smaller subproblems, since in general, the nature of these are complex and the intensive character of the course is not to end up with a complete solution, but the engagement in learning programming and algorithmic thinking to solve problems.

Moreover, it is noteworthy that the formation of teams that contain an equal division between participants who already have a good knowledge in programming and those who have no knowledge can result in difficulty and discouragement of the second group as the first group have not willing to help and this part of the group ends up doing all the work. This is a situation that occurred more with undergraduates and did not occur with the high school. It will be better worked on the development of this research, but what is proposed is that the tutor responsible for staff encourage participants who hold a more advanced knowledge can help those with fewer.

D. Content

For the times when it is necessary the presentation by the tutors of specific knowledge, such as the syntax of operations or statements, the use of slideshows for the best display in class is proposed, but the syntax of the application is made through exercises related to the problem that each team chose attack.

The fundamental reference for the course was the documentation of Python found in [24] for the preparation of slides, but it is noteworthy that given the technical writing of documentation of programming languages, the tutor should facilitate the understanding and interpretation of the team of what is described.

Naturally, students can look for solutions to issues related to the program that they are developing in the course elsewhere, such as Stack Overflow [25], and at that point it is up to the tutor evaluate the truthfulness and correctness of the information contained there.

E. Assessment Process

As the duration of the course is only two weeks, the dynamics of the interaction between the group occurs very quickly and this can cause the technical tutor accompanying the team may lose some detail to this, a 360 degree assessment (assessment of the conduct and performance of a member by everyone else including him or herself) to be held at the end of the first week, when it is expected that a part of the solution has already been developed. The criteria evaluated at that time are colaborativity and self-initiative, creativity, commitment and communication of the team member. This assessment gave us a base to better monitor some team members who had some difficulty in dealing with the team.

The technical evaluation of the knowledge acquired by students took place in two front, a theoretical (the concepts presented at each meeting, the Python syntax, loops, conditionals, functions) and practical (the project developed on the problem itself). The theoretical evaluation consists of a set of questions about each of the content presented in the course and a practical assessment consists of a final presentation of the developed solution. It is important that the whole team of technicians tutors are present for allevaluate the end product of each team, not just the tutor responsible for monitoring it.

In order to ensure the quality of the course, there must also be an evaluation of the tutors in order that they may not have such an advanced knowledge in the PBL methodology (despite receiving previous training, explained in IV Section) since the primary goal is to be facilitators in the technical obstacles of projects. The assessment of facilitators should be made at the end of the course, through an anonymous form where each student gives a score from zero to ten on five criteria: Subject Domain, incentive to the project, cordiality and ethical exposition of ideas and availability. This ensures that improvements can be made in the training of tutors.

Finally, this proposal recommends that an evaluation of the course alignment is made with the ten principles of PBL presented in section II D. For this evaluation, it was used the PBL-Test described in [16], which has ten questions that allows to verify the course adherence to the PBL principles. Students must respond to this test as well as the evaluation of tutors without the latter are present in the room, so there is less pressure and naturalness in the responses.

VI. CONCLUSIONS

The TEPI group, in line with its objectives stipulated by the Ministry of Education, seeking always apply innovative ways to improve the quality of the educational institutions. Considering the main challenges in computing education, it was proposed to implement a intensive course in programming in Python in the summer holidays, adopting the Problem-Based Learning approach (PBL) based on the xPBL methodology. This approach centered on the student makes the learning process occurs through the analysis, reflection and development solution of a problem that the student him or herself controls. This approach has been very efficient and our application and had a significant engagement of the participants in the course, considering the results obtained and the satisfaction of student in learning programming.

Some issues are still open and deserve an investigative deepening research to improve the model for teaching a programming language intensively, for example, the best team building when the group shows a wide variation in previous knowledge of each student; the development of other means for fixing the content; and the points to be modified to achieve better adherence to the ten principles of PBL.

REFERENCES

[1] S. Sentance and A. Csizmadia. "Computing in the curriculum: Challenges and strategies from a teacher's perspective". Education and Information Technologies March 2017, Volume 22, Issue 2, pp 469–495.

[2] E. Lahtinen, K. Ala-Mutka and H. Järvinen. "A Study of the Difficulties of Novice Programmers". ITiCSE'05, June 27–29, 2005, Monte de Caparica, Portugal.

[3] M. Ben-Ari. "Constructivism in computer science education". Proceedings of the twenty-ninth SIGCSE technical symposium on computer science education. Atlanta, Georgia, United States: ACM. 1998.

[4] A. M. C Oliveira., S. C. Santos and V.C. Garcia, "PBL in Teaching Computing: An overview of the last 15 years". In: Frontiers in Education 2013, Oklahoma IEE Education Society.

[5] W. Peng. "Practice and Experience in the Application of Problem-based Learning in Computer Programming Course". International Conference on Educational and Information Technology. 2010.

[6] P. Panwong, & K. Kemavuthanon., "Problem-Based Learning Framework for Junior Software Developer: Empirical Study for Computer Programming Students". Springer, Wireless Pers Commun (2014) 76: 603.

[7] C. O. Figuêredo., S. C. Santos, G. H. S. Alexandre, and P. H. M. Borba, "Using PBL to develop Software Test Engineering", CATE, Cambridge, UK, 2011.

[8] S. C. Santos and A. Pinto, "Assessing PBL with Software Factory and Agile Processes", CATE, Naples, Italy, 2012.

[9] JM Zelle, "Python programming: an introduction to computer science". Franklin, Beedle & Associates, 2004.

[10] W. J. Chun, "Core Python Programming". Prentice Hall PTR, 2001.

[11] S. C. Santos, and A. Rodrigues, "A Framework to Apply PBL in Computing Education", FIE, Erie, Pennsylvania, 2016.

[12] Barg M., Fekete A., Tony Greening, Owen Hollands, Judy Kay, Jeffrey H. Kingston Basser, and Kathryn Crawford. "Problem-Based Learning for Foundation Computer Science Courses". Department of Computer Science The University of Sydney, 1999.

[13] E. Costelloe. Teaching Programming The State of the Art. Dublin, Leinster: CRITE Technical Report, 2004.

[14] H. Alessio. "Student Perception about Performance in Problem Based Learning", Journal of Scholarship of Teaching and Learning, Vol. 4, N. 1, pp. 25 – 36, 2004.

[15] J.R. Savery,.; T. M. DUFFY. Problem Based Learning: An instructional model and its constructivist framework. Educational Technology, 1995, 35, p. 31-38.

[16] S. C. Santos, C. O. Figuerêdo, F. Wanderley, "PBL-Test: a Model to Evaluate the Maturity of Teaching Processes in a PBL Approach", FIE, Oklahoma, EUA, 2013.

[17] HS Barrows, & R. M. Tamblyn. Problem Based Learning: An Approach to Medical Education. Interdisciplinary Journal of Problem Based Learning. New York: Springer, 1980.

[18] M. Peterson. Skills to Enhance Problem-based Learning. Med Educ Online [serial online] 2.3, 1997.

[19] Andreas Holzinger. Successful management of research & development. BoD–Books on Demand, 2011.

[20] G. H. S. Alexandre, and S. C. Santos, "PBL Planner Toolkit: a Canvas-based Tool for Planning PBL in Software Engineering Education", CSEDU, Funchal, Madeira - Portugal, 2018.

[21] MQ Patton, "Qualitative Research & Evaluation Methods". 3. ed. California: Sage Publications, 2002.

[22] Sharan B. Merriam and Elizabeth J. Tisdell. "Qualitative Research: A Guide to Design and Implementation". Jossey-Bass, Fourth Edition, 2015.

[23] R. K. YIN, Case study: Design and Methods. 2nd ed, Porto Alegre. Bookman, 2010.

[24] Python 3.6.5 documentation. (nd). Retrieved from https://docs.python.org/3/

[25] Stack Overflow - Where Developers Learn, Share, & Build Careers. (nd). Retrieved from https://stackoverflow.com/