



EasyCoding - Methodology to Support Programming Learning

Marcela Viana Pereira de Almeida - a42929

Thesis presented to the School of Technology and Management to obtain the Master's Degree in Information Systems, within the scope of a double diploma with the Federal Center for Technological Education of Minas Gerais.

Work carried out under the guidance of:

Professor Maria João Varanda Pereira

Professor Luís Alves

Professor Glívia Barbosa

Bragança

2019-2020



EasyCoding - Methodology to Support Programming Learning

Marcela Viana Pereira de Almeida - a42929

Thesis presented to the School of Technology and Management to obtain the Master's Degree in Information Systems, within the scope of a double diploma with the Federal Center for Technological Education of Minas Gerais.

Work carried out under the guidance of:

Professor Maria João Varanda Pereira

Professor Luís Alves

Professor Glívia Barbosa

Bragança

2019-2020

Dedication

To my parents, my grandparents, my uncles and my dear sister. I owe them everything.

Acknowledgment

Firstly, I would like to thank God for giving me strength to persist on my dreams and for guiding me through all these years.

To my family and friends, I would like to express my sincere acknowledgments for all the support and love.

To my advisors, Professor Maria João Varanda, Professor Luís Alves and Professor Glívia Barbosa, I would like to thank for all the support and patience through the development of this work. I really appreciate all the advices, suggestions and the words of encouragement given. They contributed directly to the achievement of the proposed objectives in this work.

Finally, I would like to thank the Polytechnic Institute of Bragança and the Federal Technology Education Center of Minas Gerais for this great opportunity to accomplish my master's degree.

Abstract

The increase in failure rate in the first programming curricular units, in engineering courses, and the school dropout growth by students shows the importance of further study on the real reasons for these episodes.

When considering that this situation is amplified, in large part, by the lack of motivation and the lack of student autonomy to study programming outside the classroom context, this work proposes a methodology based on activity guides, using innovative and attractive web platforms. The proposed methodology aims to facilitate the planning of teachers' activities and to increase students' autonomy and motivation.

For the execution of this study, we evaluated the proposed methodology with programming professors and students at the Polytechnic Institute of Bragança and the Federal Center for Technological Education of Minas Gerais.

In addition, we developed a system for automatic generation of activity guides which aimed assisting teachers in the creation of exercises with innovative web platforms to motivate students to study programming beyond the classroom.

The obtained results allow us to demonstrate the importance of using innovative teaching methodologies, inside and outside the classroom, to encourage students to practice more programming exercises. Thus, as contributions, the proposed methodology can help combat school dropout in higher education in engineering courses, once it can help to keep the students motivated during the first programming subjects.

Keywords: School Dropout; Higher Education; Programming Teaching; Student Motivation; Activity Guides Automation

Resumo

O aumento na taxa de insucesso nas primeiras unidades curriculares de programação, nos cursos de engenharia, e o crescimento do abandono escolar por parte dos alunos mostra a importância de um estudo mais aprofundado sobre os reais motivos desses episódios.

Ao considerar que essa situação é ampliada, em grande parte, pela falta de motivação e falta de autonomia do aluno para estudar programação fora do contexto da sala de aula, este trabalho propõe uma metodologia baseada em guiões de atividades, utilizando plataformas web inovadoras e atrativas. A metodologia proposta visa facilitar o planejamento das atividades dos professores e aumentar a autonomia e a motivação dos alunos.

Para a execução deste estudo, avaliamos a metodologia proposta com professores e alunos de programação do Instituto Politécnico de Bragança e do Centro Federal de Educação Tecnológica de Minas Gerais.

Além disso, desenvolvemos um sistema para geração automática de guias de atividades, com o objetivo de auxiliar os professores na criação de exercícios com plataformas web inovadoras, para motivar os alunos a estudar programação além da sala de aula.

Os resultados obtidos nos permitem demonstrar a importância do uso de metodologias inovadoras de ensino, dentro e fora da sala de aula, para incentivar os alunos a praticar mais exercícios de programação. Assim, como contribuição, a metodologia proposta pode ajudar a combater o abandono escolar no ensino superior nos cursos de engenharia, uma vez que pode ajudar a manter os alunos motivados durante as primeiras disciplinas de programação.

Palavras-chave: Abandono Escolar; Ensino Superior; Ensino da Programação; Motivação aos Estudantes; Automatização de Guiões de Atividades

Contents

1	Introduction	1
1.1	Motivation	1
1.2	Objectives	2
2	Theoretical Foundation	4
2.1	School Dropout in Higher Education	4
2.2	Teaching Programming Today	6
2.2.1	Traditional Programming Teaching	7
2.2.2	Gamified Systems	8
2.2.3	Online Judges	11
2.3	Related Work	12
3	Collection of Platforms to Support Programming Teaching	14
3.1	<i>Codeboard</i> Platform	14
3.2	<i>URI Online Judge</i> Platform	18
3.3	<i>C Tutor</i> Platform	22
3.4	<i>CPuzzles</i> Platform	24
3.5	<i>Coderbyte</i> Platform	25
3.6	<i>CodinGame</i> Platform	26
4	Platform Classification Model	29
4.1	Systems Classification Models	29

4.1.1	General Classification	30
4.1.2	Content Classification	30
4.1.3	Features Classification	31
4.2	Classification of the Platforms Collected	32
5	<i>EasyCoding</i>: Automation System for Generating Activity Guides	36
5.1	System Proposal	37
5.2	System Implementation	39
5.2.1	System Architecture	39
5.2.2	System Structure	40
5.2.3	Technologies Used	50
5.3	Activity Guide Modeling	54
6	Application of the Methodology Proposed and Results	57
6.1	Application Proposal	57
6.2	Application of the Activity Guides	58
6.2.1	Results of the Research Applied on Programming I Course at IPB .	59
6.2.2	Results of the Research Applied on Programming II Course at IPB	65
6.2.3	Results of the Research with CEFET-MG Students	68
6.3	Analysis of Results	71
6.3.1	Evaluation Methods	72
6.3.2	Interview Results	73
6.3.3	Questionnaires Results	75
7	Conclusions	86
7.1	Final Considerations	86
7.2	Future Perspectives	87
A	Questionnaire Applied to Students from IPB	A1
B	Questionnaire Applied to Students from CEFET-MG	B1

C Questionnaire Applied to Teachers from IPB

C1

List of Tables

4.1	Platform general classification	30
4.2	Platform content classification	31
4.3	Platform features classification	32
4.4	Results of the platform general classification	33
4.5	Results of the platform content classification	34
4.6	Results of the platform features classification	35
6.1	Results of activity guides application during the first semester of academic year 2019/2020.	61
6.2	2nd part of the results of activity guides application during the first semester of academic year 2019/2020.	62
6.3	Results regarding the automatic assessment tests implemented in <i>Code-board</i> platform, in the first semester.	63
6.4	Results of activity guides application during the second semester of academic year 2019/2020.	66
6.5	Results regarding the automatic assessment tests implemented in <i>Code-board</i> platform, in the second semester.	68
6.6	Results of the research with the students from CEFET-MG.	70
6.7	Results of the questionnaire from <i>Appendix A</i> applied to IPB students, in the second semester of academic year 2019/2020.	76
6.8	2nd part of the results of the questionnaire from <i>Appendix A</i> applied to IPB students.	77

6.9	Results of the questionnaire from <i>Appendix C</i> applied to IPB teachers. . .	79
6.10	2nd part of the results of the questionnaire from <i>Appendix C</i> applied to IPB teachers.	80
6.11	Results of the questionnaire from <i>Appendix B</i> applied to CEFET-MG stu- dents.	83
6.12	2nd part of the results of the questionnaire from <i>Appendix B</i> applied to CEFET-MG students.	84

List of Figures

2.1	Structure of the traditional method of programming teaching.	7
2.2	Screenshot regarding <i>Codeacademy's</i> interface.	9
2.3	Screenshot regarding <i>CodeCombat's</i> interface.	10
3.1	Screenshot regarding <i>Codeboard's</i> IDE.	15
3.2	Screenshot regarding <i>Codeboard's</i> submission area.	16
3.3	Screenshot regarding an example of automatic grading on <i>Codeboard</i>	18
3.4	Screenshot regarding submission area on <i>URI Online Judge</i> platform. . . .	19
3.5	Screenshot regarding exercise categories available on <i>URI Online Judge</i> platform.	20
3.6	Screenshot regarding <i>URI Online Judge Academic's</i> interface.	21
3.7	Screenshot regarding <i>Python/C Tutor's</i> visualization tool.	23
3.8	Screenshot regarding <i>CPuzzles'</i> interface.	24
3.9	Screenshot regarding <i>Coderbyte's</i> IDE.	25
3.10	Screenshot regarding <i>CodinGame's</i> practice area, divided in categories. . .	27
3.11	Screenshot regarding <i>CodinGame's</i> IDE.	27
5.1	Architecture of <i>EasyCoding</i> system.	39
5.2	Use Case Diagram of <i>EasyCoding</i> system.	40
5.3	Activity Diagram of activity guides generation in <i>EasyCoding</i> system. . .	42
5.4	Screenshot regarding the main page of <i>EasyCoding</i> system.	43
5.5	Screenshot regarding the platforms classification inside <i>EasyCoding</i> system.	44
5.6	Screenshot regarding the full dynamic form of <i>EasyCoding</i> system.	45

5.7	Screenshot regarding the message about statistical analysis and the available buttons on <i>EasyCoding</i> system.	46
5.8	Students results in spreadsheet format generated in <i>URI Online Judge</i> platform.	46
5.9	Graphs generated regarding the <i>URI Online Judge</i> example.	47
5.10	Statistical results generated regarding the <i>URI Online Judge</i> example. . .	48
5.11	Screenshot regarding the source code and automatic assessment code generated by <i>EasyCoding</i> system.	49
5.12	Screenshot regarding the solution generation page on <i>EasyCoding</i> system.	50
5.13	Example of the header of an activity guide.	54
5.14	Example of the body of a teacher activity guide.	56

List of Abbreviations

ESTiG - Escola Superior de Tecnologia e Gestão

IPB - Instituto Politécnico de Bragança

CEFET-MG - Centro Federal de Educação Tecnológica de Minas Gerais

ACM - Association for Computing Machinery

MOOC - Massive Open Online Courses

IDE - Integrated Development Environment

UNIFEI - Federal University of Itajubá

SPOJ - Sphere Online Judge

ICM - International Collegiate Programming Contest

MOSS - Measure of Software Similarity

VBA - Visual Basic for Applications

HTTPS - Hypertext Transfer Protocol Secure

IBM - International Business Machines Corporation

SQL - Standard Query Language

CSV - Comma-separated values file

URL - Uniform Resource Locator

HTML - HyperText Markup Language

CSS - Cascading Style Sheets

Chapter 1

Introduction

In this chapter, we present an introduction of this work, seeking to describe the main motivations that supported this study and the objectives it aims to accomplish.

1.1 Motivation

School dropout in higher education has been a topic of great importance in current academic research. The high and recurring number of dropouts in the first semesters of technology courses shows the importance of further studies on this topic. This school dropout can happen due to several factors, such as the lack of motivation and the proposed challenges the students need to face inside and outside the classroom.

The students' performance and learning problems may also arise from the teaching methodologies implemented by teachers, with lack of innovation and without the use of attractive technologies as teaching means.

In this sense, a larger study on new methodologies to support programming learning in the first subjects of technology courses becomes essential in the search for better teaching quality and a greater captivation and motivation of students.

1.2 Objectives

This work seeks to propose a methodology based on activity guides elaborated using innovative and attractive web platforms. The study aims to improve the quantity and the quality of time that the student uses to program, by motivating them in the first programming subjects of technology courses. In addition, it seeks to improve the students' autonomy and learning outcomes, encouraging them to practice more programming beyond the classroom context.

This study also proposes a system for automating the construction of these activity guides, with the objective of assisting teachers' planning and supporting them on the application of the exercises. The system is capable of generating activity guides automatically according to the preferences of each teacher, using innovative web platforms to assist teacher's assessment and analysis of students performance.

In summary, this work seeks to fulfill the following specific objectives:

1. Collection of information on innovative and attractive web platforms, that help teaching C programming;
2. Creation of a classification system for the collected web platforms;
3. Classification of the platforms collected and creation of a website to make the study available to teachers at the Polytechnic Institute of Bragança and the Federal Center for Technological Education of Minas Gerais;
4. Development of activity guides, using the platforms classified, to be applied during the first and second semesters of the academic year 2019/2020, of engineering courses, with respective monitoring;
5. Development of a system that allows the teacher to automatically generate new activity guides, according to characteristics pre-defined by the classification model;
6. Validation of the methodology, based on monitoring results and on questionnaires for collecting opinions from the teachers and students participating in the study.

This work provides, as main results, an innovative teaching methodology, which can contribute to students' motivation. In addition, the study provides a system that automates the generation of activity guides, with the objective of assisting the teacher in the programming teaching process.

As practical contributions, this study contributes to stimulate the study of programming logic inside and outside the classroom environment. Moreover, the activity guide can be used as a model for creating teaching methodologies for other content within and outside the scope of Engineering. As a scientific contribution, other researchers can evaluate the model created with different user profiles and teaching contexts, to evaluate the advantages and disadvantages of this methodology.

Therefore, this document is structured in 6 main chapters. The first chapter shows an introduction to school dropout in higher education, present-day systems in programming teaching and related work; the second chapter identifies and characterizes the programming teaching platforms used in this study; the third chapter specifies the developed model for the classification of the platforms collected; the fourth chapter explains the system developed to automate the generation of the activity guides, describing its proposal, implementation and how the activity guides were modeled; the fifth chapter exposes the application of activity guides to students and the results obtained, with respective analysis of the study; finally, the last chapter presents the final considerations and perspectives for future works.

Chapter 2

Theoretical Foundation

In this chapter, we present a literature review in order to support the conception of this work. Section 2.1 analyzes school dropout in higher education, mainly in the scope of technology courses, observed in different educational institutions around the world. Section 2.2 presents teaching methodologies currently used to teach programming and some online tools available in computing area. Finally, Section 2.3 presents the works related to this project.

2.1 School Dropout in Higher Education

In recent years, higher education in Portugal has experienced numerous changes, mainly in the growth of the number of students, courses and also in the cultural diversity found in the student population. When analyzing the constant transformations in the academic environment, we can find some problems associated with the increase of students' failure and school dropout.

Student evasion in the academic environment is an international problem that can generate consequences for students and the country's educational systems. The search for the causes of this abandonment is starting to get more attention in the academic field and is increasingly becoming object of studies around the world.

According to a survey conducted by Lobo and Silva Filho [1], in Brazil, dropout rates

in the Academy have remained approximately constant over the past 15 years, with slight variations from year to year, staying at approximately 22%, lower for the public sector and higher for the private sector. This means a big loss in tuition revenue throughout the year.

Lobo and Silva Filho [1] explain that considering an evasion of 20% per year in classes with 100 students entering a 4-year course, admitting a constant drop-out in each grade, it will represent instead of 400 paying enrollments in the total of a four-year course, only 297. Which means a loss of revenue of 27% with monthly fees.

Still according to this research, computing courses continue to have one of the highest student dropout rates in the private sector (31%) and in the public sector (22%). This study could reveal that, as the dropout rates remain high throughout these years, there are still few regular institutional programs to combat evasion with action planning, results monitoring and collection of successful experiences.

Aiming at reducing this dropout rate, some studies focus on finding out about issues and motivations that lead students to drop out of higher education without first completing the chosen course. Almeida and Vasconcelos [2] describe some aspects that can motivate these issues, and can be found mainly in the first year of science and technology courses in Portugal, such as the transition from secondary education to higher education, which can be challenging and can group areas of new demands that students would not be familiar with.

According to this research, other issues would be: adaptations to the new academic rhythm, which involve different teaching methodologies and assessment systems; creation of new and more mature social bonds with teachers and colleagues; autonomy and personal development requirements, as well as commitment to the chosen course.

Another pertinent aspect investigated in Almeida and Vasconcelos research [2] was the teaching and learning methods applied by teachers, often with lack of academic innovation. According to them, only expository teaching methods, can favor passive attitudes by the students, generating superficial learning and reducing the students' motivation.

In view of the context highlighted above, one can infer that there is a need for a

further study on innovative teaching methodologies. Moreover, the use of innovative support systems, can help combat academic failure in higher education.

2.2 Teaching Programming Today

Over the past few decades, teaching methodologies have undergone several changes. The traditional way of teaching has been transformed, especially after the emergence of the Internet and didactics aimed at online teaching, which use technology as a means of providing more knowledge to students. In this sense, several online learning aid platforms have been developed over the years.

When we think about the scope of study of Computer Science, it is possible to identify the benefits of using these online systems in learning subjects such as Computer Programming.

As Nagai, et al. [3] state,

Computer Programming is a discipline in which students entering higher education generally have contact for the first time, unlike other subjects such as chemistry, physics or mathematics. Many do not have the knowledge of what it is to program and, as it is a skill that uses a lot of logical reasoning, mathematics and the ability to solve problems, they end up having a lot of difficulty, resulting in a high failure rate.

Considering this initial difficulty in computer courses, online systems to assist teaching programming can become important tools in helping students and in fighting high failure rates. These systems work with different methodologies, such as *Gamification* concepts - described in Section 2.2.2 - and *Online Judges* - described in Section 2.2.3. These tools can be used by programming students and almost all of them have some motivating features like animations, interactive debugging, collaborative programming, online chats, forums, programs to complete, games, submission platforms, rankings and so on.

As this work focuses on assisting methodologies in teaching programming, this section

will focus on reporting the most currently used methods to teach programming, in addition to presenting well-known tools that hold some of the motivating features mentioned above.

2.2.1 Traditional Programming Teaching

According to Borges [4], the programming teacher's didactic posture usually follows the "traditional way", shown in the Figure 2.1. This teaching methodology is common in all areas of knowledge, but especially in courses in the computer science area.

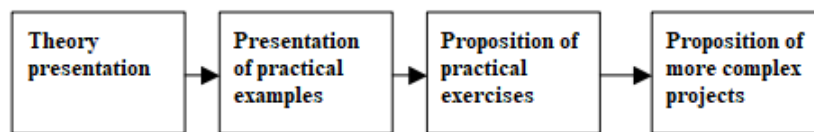


Figure 2.1: Structure of the traditional method of programming teaching.

When analyzing the traditional method, it is possible to infer that it may be not able to catch the attention of the students when it comes to being more interested in programming subjects. This is due to the practical part being normally based only on the application of exercises and projects proposition in the end, without any innovative approaches that could motivate the student.

Following this conception, Rapkiewicz et al. [5] explain that the exaggerated use of repetition of problems with textual statements passed in a printed list of exercises or on the board makes the teaching and learning process monotonous and tiring. The authors claim that this approach to teaching algorithms and programming does not do much to motivate students, as it postpones students' contact with the use of a practical resource (the computer) that is the very reason for the development of algorithms and programs.

Therefore, the traditional way of teaching programming can generate a learning barrier for the student. The use of an innovative didactic material can be an alternative to encourage students' in practicing more programming and consequently arouse a greater interest in the subject.

2.2.2 Gamified Systems

According to Kapp [6], gamification is the use of elements traditionally thought of as game-like of "fun" to promote learning and engagement. It is a teaching methodology that can motivate students and help solve problems, as the competitive nature of games encourages many to do their best to accomplish the goal of winning.

The mechanics of a gamified system are made up of a series of tools that, when used correctly, promise to yield a meaningful response (aesthetics) from the players [7]. Some of these elements are: Points, levels, leaderboards, badges, challenges/quests, onboarding and engagement loops. All of these tools have an important role on the gamification methodology to promote students motivation.

Following this concept, gamified online systems can be found on the internet and aim to engage and motivate more the programming students. Among them, we can mention *Codeacademy* [8] and *CodeCombat* [9] systems.

Codeacademy

Codeacademy provides free, online and interactive lessons for a variety of programming topics [10]. It was founded, in 2011, by Zach Sims and Ryan Bubinski. This tool is designed with gamification concepts and has courses organized in sections that have sequences of exercises. Each exercise has an introductory text, with instructions on what the student needs to implement, and an interactive IDE (Integrated Development Environment) for the student to solve the exercise within the system itself. In Figure 2.2, it is possible to observe the interface of *Codeacademy* environment.

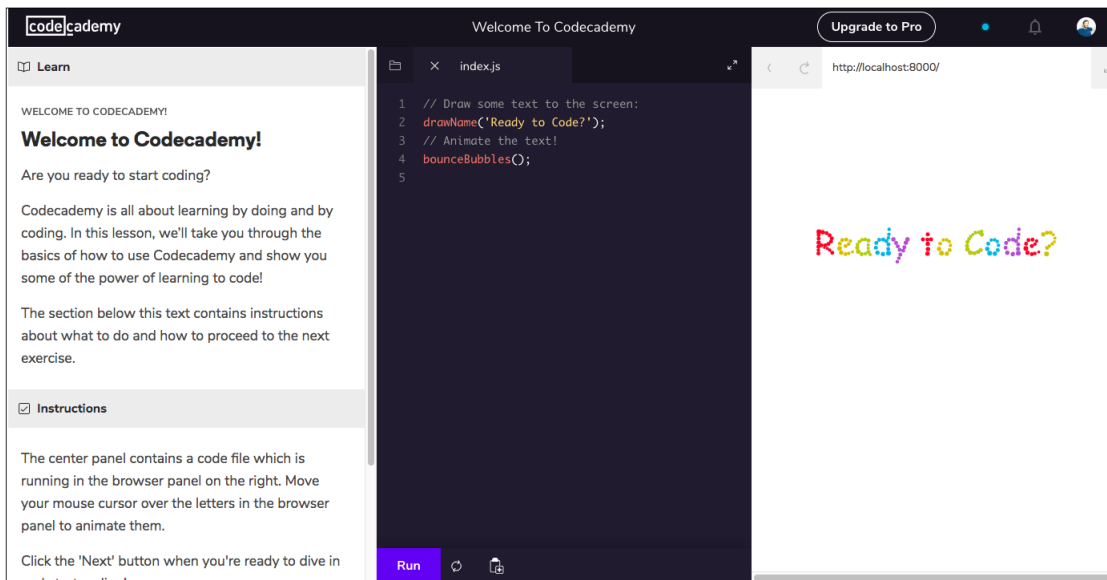


Figure 2.2: Screenshot regarding *Codecademy's* interface.

As stated above, the *Codecademy* platform is based on the concept of gamification. Swacha and Baszuro [11] explain that the system can provide students with points for each solved exercise and offer rewards such as medals and *badges* when completing lessons or performing more difficult tasks. According to the *Codecademy's* website [8], there are several languages available to practice and learn, such as: HTML & CSS, Python, JavaScript, Java, SQL, Bash/Shell and Ruby.

CodeCombat

CodeCombat is an educational adventure game system aimed primarily at learning computer programming. Unlike platforms like *Scratch* [12] and *Code.org* [13], in which code is produced using drag and drop blocks, the programs produced by students in *CodeCombat* are written in real programming languages (JavaScript and Python languages are available), as they usually are in a real context of use [14]. In Figure 2.3, it is possible to view the interface of the *CodeCombat* platform.

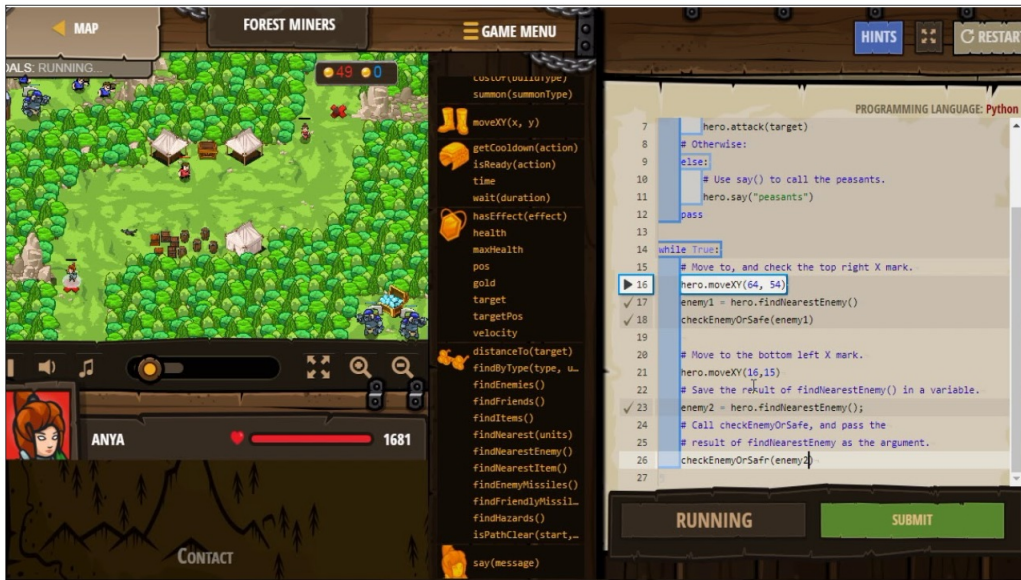


Figure 2.3: Screenshot regarding *CodeCombat*'s interface.

CodeCombat is a game-based program where the student can see in real time the characters react according to the implemented code and earn rewards after each step. Severgnini and Soares [14] emphasize that *CodeCombat* invites the player to think, recognize problems, formulate hypotheses, write possible solutions and see if they solve the problem. The system still seeks to deliver a fun experience, since it was created to “involve all students so that they believe that coding is for them.” [9].

It is important to point out that the platforms mentioned above were not applied in this work due to the lack of compatibility with the C language, an important requirement for this work to develop the intended methodology for the first programming courses. However, other gamified tools were used in this work, the *CodinGame* [15] and *Coderbyte* [16] systems, which will be better explained in Chapter 3 of this work.

2.2.3 Online Judges

Online Judges can be defined as systems that do automatic assessments by receiving the implemented source code sent by the user and subsequently evaluate this code. Chaves et al. [17] explain that during the execution of the program, Online Judges use formatted data as the program entry, process that data and compare the results obtained with the expected results.

According to Kurnia, et al. [18], most programs of an algorithmic nature only need to obtain a properly formatted data standard as input and, from that data, perform the due processing. After processing, the results are presented in a formatted template in standardized output. Following this sense, it is possible that the evaluation of programs is done automatically using a tool that generates the input data and another that obtains, verifies and compares the results obtained.

Chaves et al. [17] also describes that Online Judges are systems used in programming marathons and can be easily found on the Internet. As examples, SPOJ Brasil [19] and UVA Online Judge [20] systems can be mentioned.

SPOJ Brasil

According to Chaves et al. [21], *SPOJ Brasil* is the brazilian version of Sphere Online Judge (SPOJ) and its main features are:

1. It has a friendly interface;
2. Provides discussion forums, ranking, in addition to news and information links;
3. The problems support several programming languages, including C, C++ and Java.

SPOJ Brasil also allows advanced users to organize contests under their own rules and the tasks are prepared by its community of problem setters or are taken from previous programming contest. Besides having these benefits, *SPOJ Brasil* was not applied in this study because the exercises are not distributed on programming topics nor difficulty levels, which makes the system difficult to use for collecting exercises.

UVA Online Judge

Szymon et al. [22] describe that *UVA Online Judge* was founded in 1995 by Miguel Ángel Revilla, a mathematician who lectures on algorithms at the University of Valladolid in Spain. The platform provides an enormous archive of programming challenges originating from the ACM (Association for Computing Machinery) International Collegiate Programming Contest [23].

This system provide several programming exercises for students to solve and submit them on the platform itself. The user only needs to select the programming language to be used in the development of the activity and send the solution to be evaluated. The *UVA Online Judge* also has discussion forum, statistics on problems and a ranking of users.

Besides these benefits, *UVA Online Judge* was not applied in this study because of the lack of compatibility with the C language. Moreover, the exercises are not distributed on programming topics nor difficulty levels, which makes the system difficult to use for collecting exercises.

The *SPOJ Brasil* and *UVA Online Judges* were not used in this work due to the reasons mentioned above. However, another Online Judge system was applied in this study, the *URI Online Judge* [24], which will be better explained in the Chapter 3 of this work.

2.3 Related Work

The main objective of this work - the development of a methodology to support the teaching of programming, together with the implementation of a platform for automatically generating programming activity guides - has some topics already discussed and studied previously in different projects. It is possible to find some research with applications of different teaching methodologies, in an attempt to support the teaching of programming, in the early years of computer courses.

Combéfis, Beresnevičius and Dagiené [25] performed an overview, characterization and discussion of learning programming through online platforms that use games and contests as learning methodology. After the classification made in this study, it was concluded that these online platforms are useful to foster their users to learn and keep progressing, as well as making programming fun. Moreover, they verified some characteristics that make the online platform successful in motivating students, such as guidance through the system, engagement, feedback and (manual or automated) assessment.

Izeki, Nagai and Dias [3] demonstrate an experience in using gamified online tools in introductory computer programming. This study was applied among students of engineering courses at the Federal University of Itajubá (UNIFEI), using three online tools - *Code.org*, *Kahoot* and *Socrative*. The study showed satisfactory results, contributing to an improvement in the grades of the basic programming discipline.

Prados et al. [26] developed a e-learning environment for automatic generation and correction of technical/engineering exercises, in order to combat the alarming failure rate of the technological courses, aiming at mainly supporting continuous assessment and providing teachers with feedback of students weakness. For this objective, they created base exercises for mathematic, programming and database courses, that can be generated automatically, besides an exercise correction module that implements the automatic assessment of the exercises. The platform was tested in Girona Polytechnic University with very promising results.

Therefore, as it was observed in the works above, programming teaching using different methodologies has been a subject of great relevance in the academic environment. However, no work has been found that implies the automatic generation of programming activity guides - for students and teachers - using innovative online platforms with functionalities relevant to computer science learning. For this reason, the present work becomes more relevant in this aspect.

Chapter 3

Collection of Platforms to Support Programming Teaching

In this chapter, this work presents C programming teaching platforms in order to assist the methodology delivered in this study. The platforms' main characteristics and functionalities will be exposed to help understand why they were selected to be part of this research.

It is important to emphasize that only platforms that are compatible with teaching/learning C programming language were chosen. This is due to the common use of this language as a programming teaching method in the first semesters of technological courses.

3.1 *Codeboard* Platform

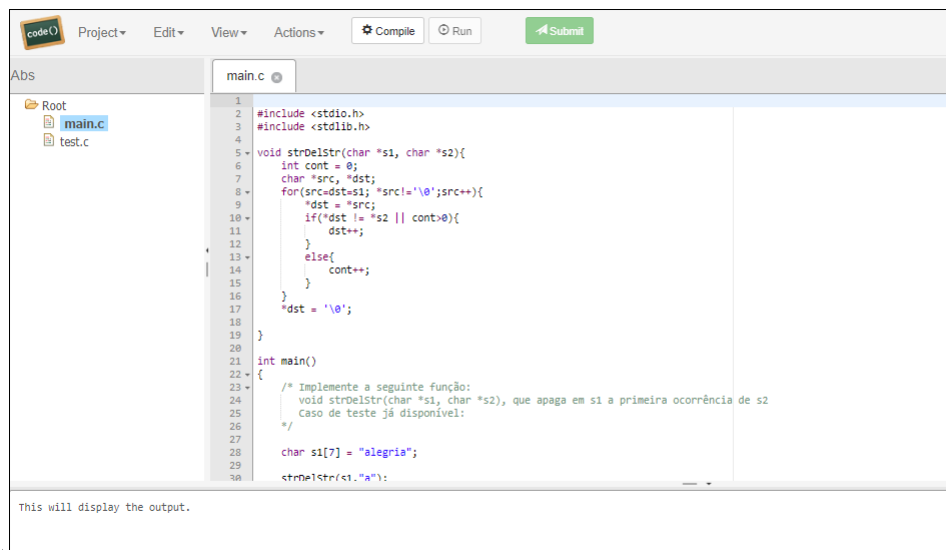
The *Codeboard* platform is a system that started as a research project, in 2014, in the ETH Zurich Software Engineering Chair, headed by Professor Bertrand Meyer. This web application has as main objective to assist teachers and students in teaching and learning programming.

According to Estler and Nordio [27], the *Codeboard* environment can be used in different scenarios such as supporting teachers who already use MOOC (Massive Open Online

Courses) platforms - open courses offered through virtual learning environments - in teaching, but do not provide the possibility of creating exercises with automatic assessments. In addition, the platform assists students in submitting their exercises and teachers in analyzing and evaluating them, by manipulating these submissions in a simplified way.

The *Codeboard* platform has some important features, such as:

1. Possibility of creating and solving exercises within the platform's IDE (Integrated Development Environment) itself, with code compilation and submission, which can be seen in Figure 3.1.



The screenshot shows the Codeboard IDE interface. At the top, there is a menu bar with 'Project', 'Edit', 'View', and 'Actions'. To the right of the menu are buttons for 'Compile', 'Run', and 'Submit'. Below the menu is a file explorer on the left showing a project structure with 'Root', 'main.c', and 'test.c'. The main editor area displays the code for 'main.c' with line numbers from 1 to 30. The code includes headers for `<stdio.h>` and `<stdlib.h>`, and defines a function `strDelStr` that removes the first occurrence of a character `s2` from a string `s1`. The `main` function implements this logic and tests it with the string "alegria". At the bottom of the IDE, there is a placeholder text: "This will display the output."

Figure 3.1: Screenshot regarding *Codeboard's* IDE.

2. The teacher has full control over the exercises submitted by each student and there is the possibility of exporting the results in spreadsheet format (CSV file). An example of the teacher's submission area can be seen in Figure 3.2.

Codeboard.io Explore Docs marcelavp Sign out

Submissions for project 131871 [Download all](#)

Filter by username

celaav (1)

Created	Result	Tests passed	Tests failed	LTI
Oct 4, 2019 11:07:19 AM	1.0000	3	0	no Open in IDE

marcelavp (10)

Created	Result	Tests passed	Tests failed	LTI
Oct 9, 2019 9:30:23 PM	1.0000	3	0	no Open in IDE
Oct 9, 2019 9:17:43 PM	0.6600	2	1	no Open in IDE
Oct 9, 2019 9:14:31 PM	0.6600	2	1	no Open in IDE
Oct 9, 2019 9:14:20 PM	0.6600	2	1	no Open in IDE
Oct 9, 2019 9:13:27 PM	0.6600	2	1	no Open in IDE
Oct 9, 2019 9:11:59 PM	0.6600	2	1	no Open in IDE

Figure 3.2: Screenshot regarding *Codeboard's* submission area.

3. Possibility of determination of valid date range for resolution and submission of exercises by students.
4. Different programming languages supported: C, C ++, Eiffel, Haskell, Java, Python, Python3 and Reason.
5. Automatic assessment system which can evaluate students submissions according to their results. According to *Codeboard's* website [27], the automatic assessment system currently supports two ways of grading the submissions:

(a) **Automatic grading using a result string:**

Automatic grading can be implemented by the teacher using every programming language available in *Codeboard* system, by simply printing a special string as the last output when the exercise is executed. This string must be as following:

```
<!--@test=the_grade_value;num_tests_passed;num_tests_failed;-->
```


On submission, *Codeboard* compiles and executes the submitted program and checks if the program output ends with a string satisfying the result string format. If a result string is found, the grade value (between 0 and 1), the number of tests passed and the number of tests failed are extracted from the result string and stored as part of the submission details [27].

(b) **Automatic grading using unit tests:**

Codeboard supports unit tests for a number of programming languages, which are *Java-JUnit*, *Haskell-HSpec* and *Python-UnitTest* project types.

On submission, *Codeboard* will run all unit tests available in the "submission test folder". The submission test folder is defined in the "codeboard.json" configuration file, in the property "DirectoryForTestSubmissionFiles". The grade (value between 0 and 1) is automatically calculated based on the number of passing and failing test cases in this folder [27].

In Figure 3.3, it is shown an example of an automatic grading code using a result string, implemented in C programming language for an exercise. It is important to elucidate that the automatic grading code can be hidden. By doing this, the students won't have access to this code, only the teacher.

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <math.h>
4
5 void testUnitario(char * y){
6
7     int i,cont=0,ok=0;
8     double grade = 0;
9     int testsPassed = 0;
10    int testsFailed = 1;
11
12    for(i=0;y[i]!='\0';i++){
13        if(y[i] == 'A')
14            ok = 1;
15        else
16            ok = 0;
17        cont++;
18    }
19    if(ok == 1 && cont == 2){
20        testsPassed++;
21        testsFailed--;
22    }
23
24    if(testsPassed == 1)
25        grade = 1;
26
27    printf("<!--@test=%.2lf;%d;%d;-->",grade, testsPassed, testsFailed);
28
29
30 }

```

Figure 3.3: Screenshot regarding an example of automatic grading on *Codeboard*.

The *Codeboard* platform is currently available only in the English version and does not provide exercises itself neither has debugging system. This platform was chosen to be part of the systems used in this study mainly because of its possibility to create exercises in C programming language, compile and submit solutions, automatic assessment easy to implement and teacher's access to the students submission and grades.

3.2 *URI Online Judge* Platform

URI Online Judge is a system that has been developed at URI - Integrated Regional University - Campus of Erechim (Brazil), since 2011. This platform provides programming exercises with the same aspects of the problems created in the ICM (International Collegiate Programming Contest) of ACM, an annual programming competition between universities around the world, sponsored by IBM (International Business Machines Corporation).

Moreover, the *URI Online Judge* platform offers an *online judge* that has the function

of testing students' solutions to the problems solved and displaying their scores in a ranking format. According to Bez, Tonin and Selivon [24], the tool has all the basic features necessary for a programming portal, such as:

1. Real-time correction.
2. Use of special judges who automatically correct the students' submitted solution.
3. Forum dedicated to the users.
4. Solutions accepted to be submitted in different programming languages, such as C, C++, Go, Haskell, Java 7, Java 8, C#, JavaScript. In Figure 3.4 it is possible to see its submission area.

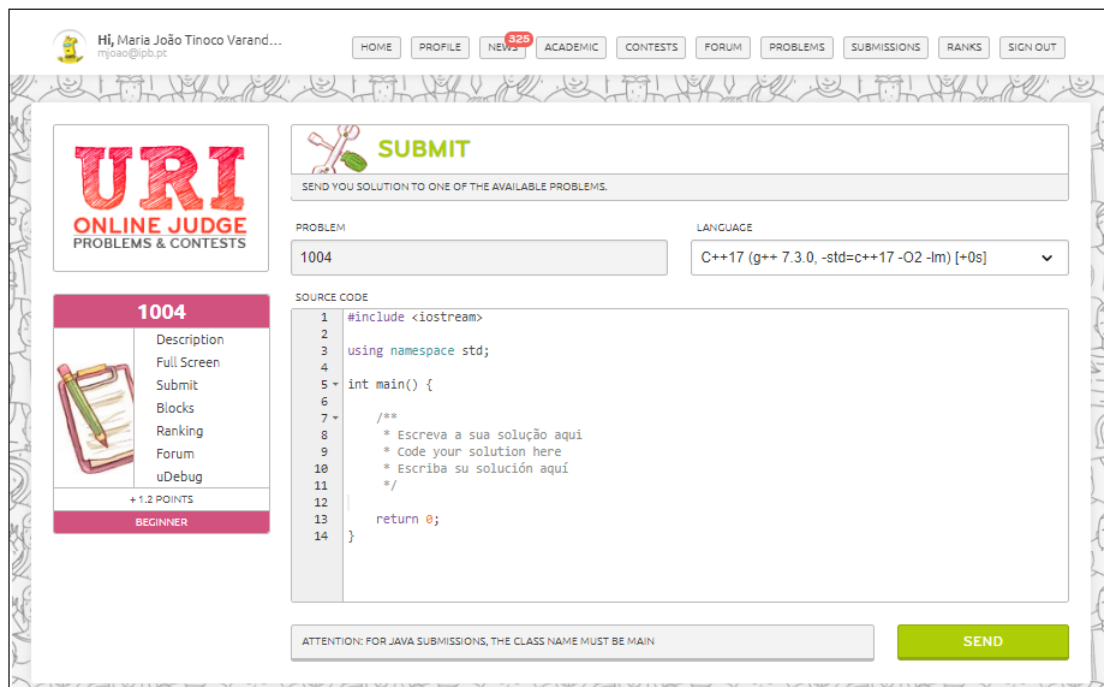


Figure 3.4: Screenshot regarding submission area on *URI Online Judge* platform.

5. Problems separated by categories and difficulty levels, which can be seen in Figure 3.5.

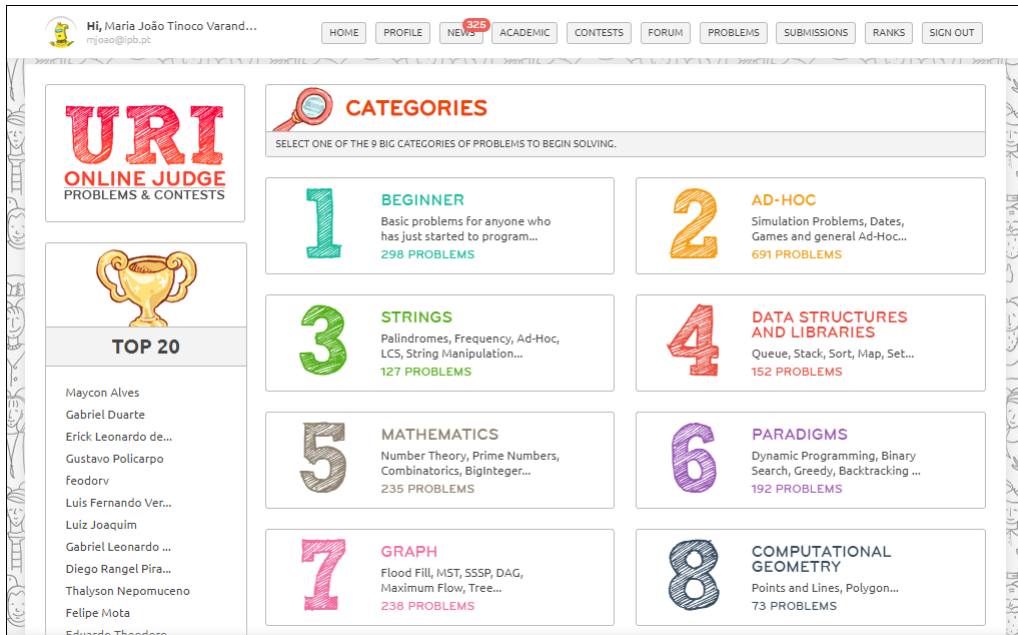


Figure 3.5: Screenshot regarding exercise categories available on *URI Online Judge* platform.

6. Use of gamification concepts, with the application of *badges* and *ranks* reward system, important for motivating students.
7. Exclusive ranking for each university registered on the platform, in addition to different rankings by problem and programming language.
8. Visualization of the source code lines with error.

In addition to the characteristics above, the platform developed, in 2013, an exclusive module for teachers and coaches of programming teams. The *URI Online Judge Academic* can help teachers to follow in real time the evolution of students and separately evaluate each solution. Its interface can be seen in Figure 3.6.

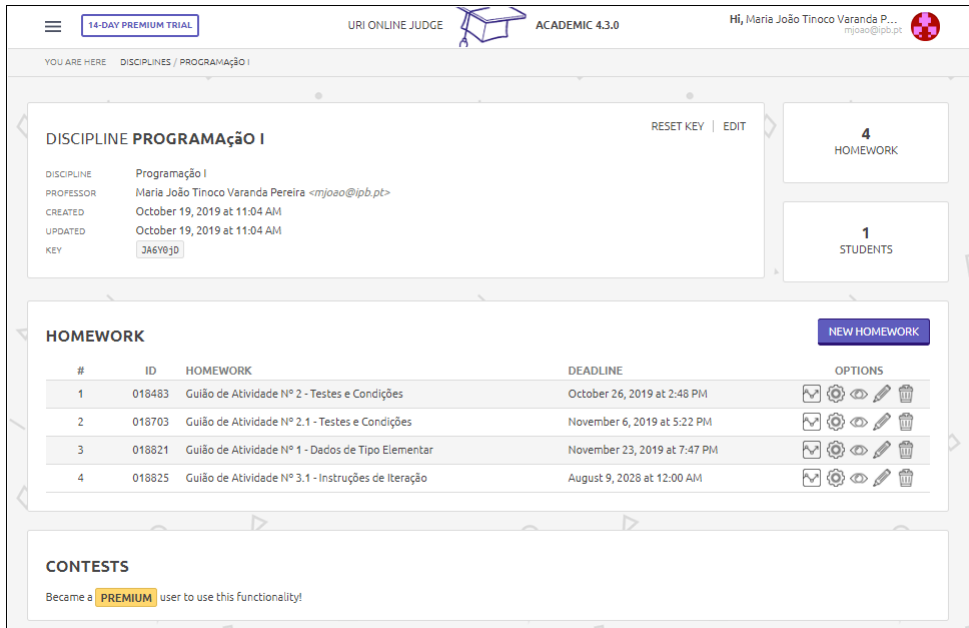


Figure 3.6: Screenshot regarding *URI Online Judge Academic's* interface.

The *URI Online Judge Academic* is a separate module and is intended only for teachers. In order for them to be registered, the registration process as a teacher obeys an assessment of the teacher's status by the URI support team. This Academic module was presented in its first version in 2013 [28] and, in 2014, an update was released with some new solution acceptance restrictions and integration with MOSS (Measure Of Software Similarity) [29] - an automatic similarity determination system software.

The *URI Online Judge Academic* has some important features in assisting the teacher such as:

1. Possibility of creation of *Homeworks* by subject, containing the exercises available on the *URI Online Judge* platform.
2. Possibility of determination of valid date range for the resolution of *Homeworks*.
3. Possibility of viewing the submission history of each student for each problem solved, as well as their source code.

4. Possibility of exporting students' results, with number of tests passed and failed, in spreadsheet format (CSV file).

Currently, all *URI Online Judge* problems are available on the platform in Portuguese and in English. The platform interface is available, in addition to the languages above, also in Spanish. *URI Online Judge* does not have compilation feature or debugging system. This platform was chosen to be part of the systems used in this study mainly because of its automatic correction by *online judges*, library of programming problems available in C programming language and separated in categories and difficulty levels, possibility of submitting exercises and full control of the students results by the teacher within the Academic module.

3.3 *C Tutor* Platform

The *Online C Tutor* is a free and open-source project that provides a web tool for compiling and viewing programs, helping to teach computer science content. The original name of the platform is *Python Tutor*, however the system created different services for other types of programming languages like *C Tutor* that supports C programming language.

According to Philip J. [30], there are some characteristics that make *Python Tutor* a modern, unique and effective tool:

1. *Web-based*: *Python Tutor* is the only code visualization tool that runs in a web browser with no software or plugin installation requirements. Some of the main features of this web tool are described below:
 - (a) View the line of code currently running.
 - (b) Possibility to move forward and backward in execution using a slider and buttons.
 - (c) Visualization of the variables and the current state of the stack.
 - (d) Visualization of the contents present in the heap and the pointers.

- (e) Visualization of the program output.
- (f) Generation of a shareable URL of the current view at an exact point of execution.
- (g) Collaborative work available when users enter in the same URL.

All these characteristics mentioned above can be seen in Figure 3.7.

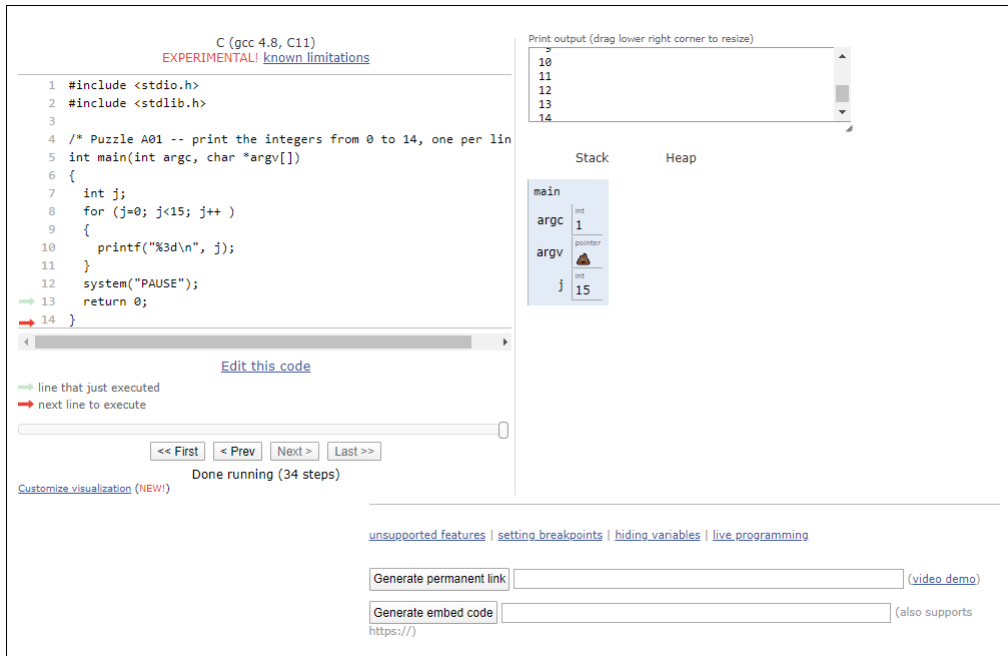


Figure 3.7: Screenshot regarding *Python/C Tutor's* visualization tool.

2. *Embeddable*: *Python Tutor* is the only program viewer that can be seamlessly embedded in web pages. Examples of use are in interactive digital textbooks that can incorporate *Python Tutor* into lessons (inserting a line of JavaScript code per view). Students can read lessons and interact with code views on the same page [30].

In addition to these features, the *Python Tutor* platform accepts different programming languages: Python, Java, C, C++, JavaScript and Ruby. This platform doesn't provide exercises. *C Tutor* was chosen to be part of the systems used in this study mainly because of its possibility to visualize each line of the code with errors, the contents present inside

the programming structures created and variables' current state, besides its collaborative work feature.

3.4 CPuzzles Platform

CPuzzles platform is a website that contains a collection of C programming exercises, called puzzles, and their solutions. As each puzzle has available its solution, CPuzzles is recommended only for teachers, in order to help them gather exercises for the students.

According to its website [31], CPuzzles' purpose is to supplement a C textbook and to provide a way to review what students already learned. The Figure 3.8 shows the platform's interface.

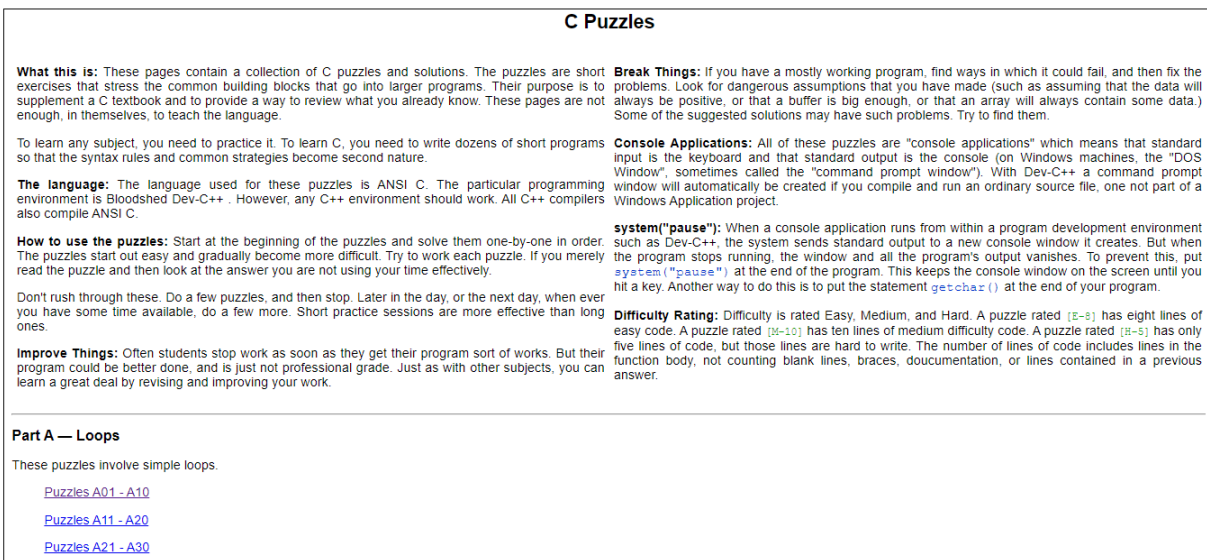


Figure 3.8: Screenshot regarding CPuzzles' interface.

The puzzles inside the platform are divided in eight groups, each one regarding a C programming subject. Each puzzle has next to its statement a difficulty rating, in order to help the teacher choose which exercise better meets students needs. Besides that, the teacher can find an example of solution and a *Skeleton* (a base code), which helps the student to start the exercise, for each puzzle in the platform.

CPuzzles is currently only available in English version and works with one type of programming language: C. The platform doesn't have compilation or debugging features. *CPuzzles* was chosen to be part of the systems used in this study mainly because of its library of C programming exercises that proved to be useful in the formulation of the activities.

3.5 *Coderbyte* Platform

Coderbyte is a web system developed to offer a collection of programming challenges and courses with the aim to help users to prepare for upcoming job interviews, as well as practicing more programming. The code challenges range in difficulty and *Coderbyte* has its own IDE (Integrated Development Environment) inside the platform, which can be seen in Figure 3.9. Therefore, the user can solve the problem inside its online editor and see, in real time, if the solution is correctly implemented.

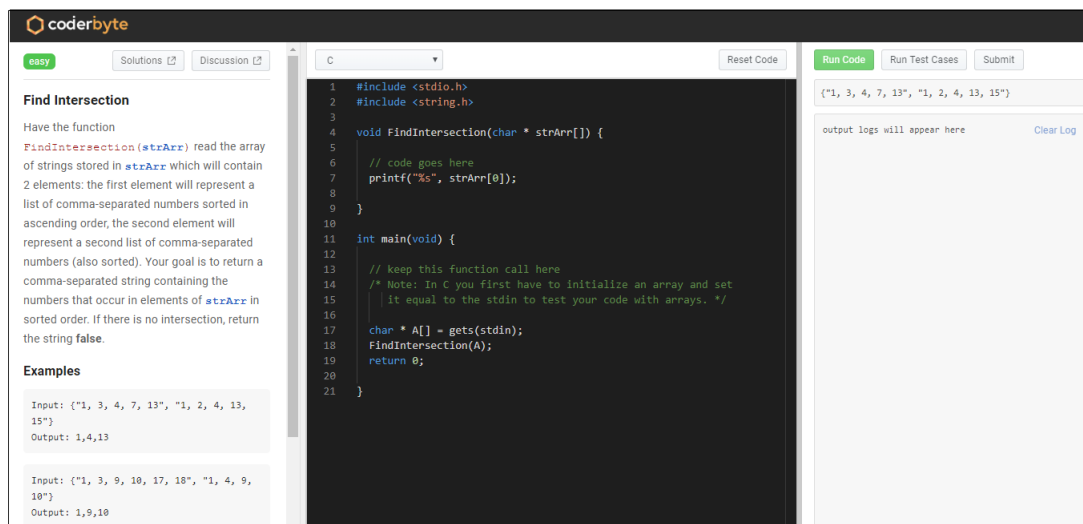


Figure 3.9: Screenshot regarding *Coderbyte's* IDE.

According to the *Coderbyte* website [16], the platform supports different programming languages such as: C, C++, Go, Java, JavaScript, C#, PHP, Python, Python3, Ruby

and Swift. The platform has compilation and automatic assessment features, but doesn't provide debugging system. This system was chosen to be part of the platforms used in this study mainly because of its library of exercises, automatic assessment and possibility of compilation and submission of the exercises.

3.6 *CodinGame* Platform

CodinGame is a gamified platform for teaching programming languages. Its methodology is based on puzzles with high level animations with characters and playful scenes where the students can see after solving the problem. According to *CodinGame's* website [15], their goal is to let programmers keep on improving their coding skills by solving problems and learning new concepts.

There are some characteristics that make *CodinGame* an important feature to learn programming, such as:

1. Real-time correction with automatic evaluation.
2. Use of gamification concepts, with the application of a *badges* and *ranks* reward system, important in motivating users.
3. Forum dedicated to the users.
4. Problems separated by categories and difficulty levels, which can be seen in Figure 3.10.

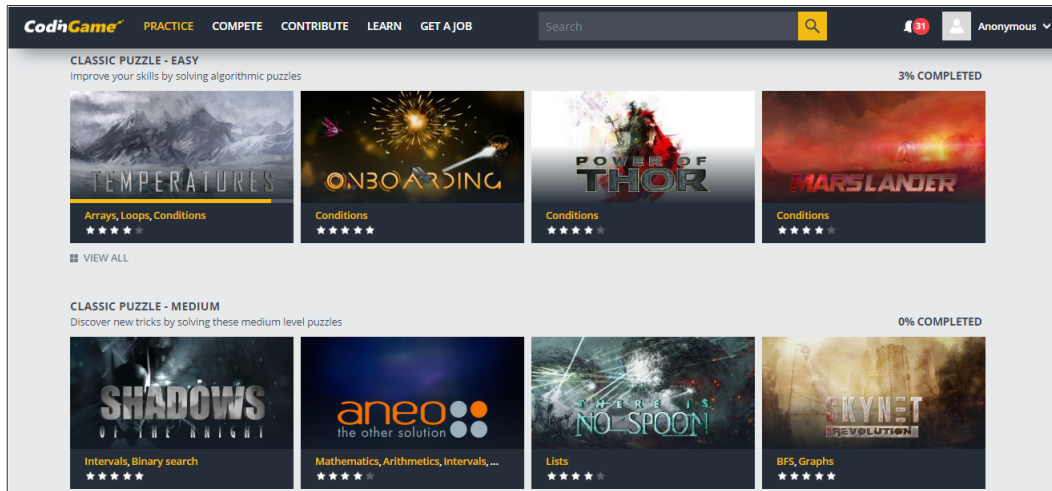


Figure 3.10: Screenshot regarding *CodinGame*'s practice area, divided in categories.

5. Possibility of solving the problem within the platform's IDE itself, with code compilation, error display and submission. *CodinGame*'s IDE can be observed in Figure 3.11.

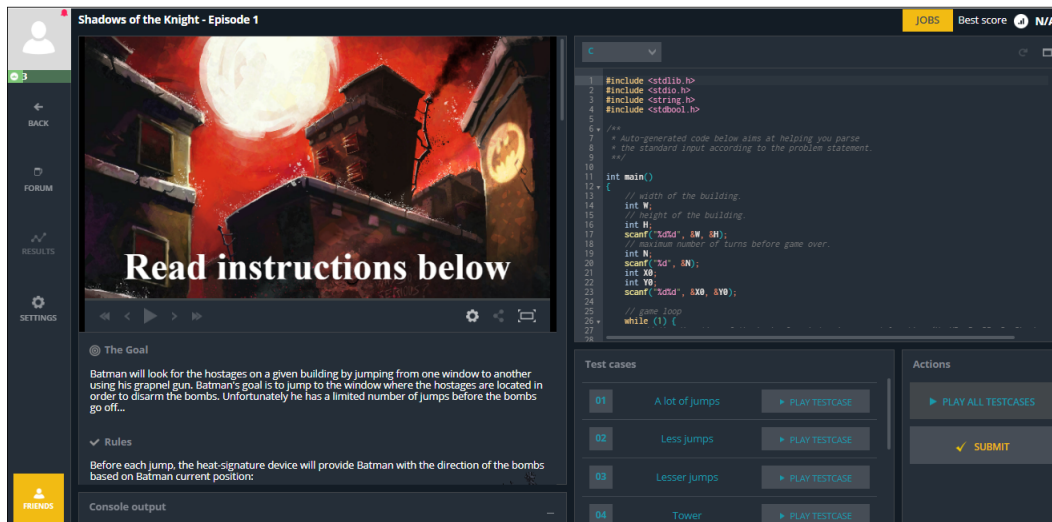


Figure 3.11: Screenshot regarding *CodinGame*'s IDE.

6. Solutions accepted in different programming languages: C, C++, Java, C#, JavaScript, Python3, F#, Go, Haskell, Lua, Objective C, Pascal, Perl, PHP, Ruby, Rust, Scala

and VB.NET.

The *CodinGame* platform is currently available in the English and French version. It doesn't have debugging system. This system was chosen to be part of the platforms used in this study mainly because of its gamification concepts with high level animations within the exercises, code compilation and submission, as well as automatic assessment.

Chapter 4

Platform Classification Model

After collecting the web platforms for learning C programming, it was possible to create a classification model to better analyze the main characteristics of each one. With these models, it was possible to classify the systems for the use in activity guides, to be created for the Programming I and Programming II syllabus in technological courses, at the Polytechnic Institute of Bragança (IPB) and Federal Technology Education Center of Minas Gerais (CEFET-MG). The Section 4.1 describes the classification models created for the systems collected and the Section 4.2 shows the classification of the platforms itself.

4.1 Systems Classification Models

To classify and analyze the characteristics of each collected platform in Chapter 3, we developed a classification model. As explained below, this model comprises three types of classification, namely, a general classification, a content classification and a features classification.

4.1.1 General Classification

Classification of the general characteristics of each platform collected, according to the languages allowed, free or paid access, application context as well as interface quality, elucidated in Table 4.1.

Table 4.1: Platform general classification

Classification	Characteristic
Language(s) supported	Which programming languages the platform supports.
Platform access	The platform is free and easy to access or it is paid and has restricted access.
Application context	The platform fits into the application method inside and/or outside the classroom.
Interface quality	The platform is flexible and easy to use or not very flexible and difficult to use.

4.1.2 Content Classification

Classification according to the syllabus of Programming I and Programming II disciplines, to which each online platform applies. When the platform has exercises available, the classification regards the existence or not of exercises about each content. When the platform doesn't have exercises available, the classification regards the system being able to compile or execute source codes about each content or not. This classification is elucidated in Table 4.2 below.

Table 4.2: Platform content classification

Content	Characteristic
Introductory Concepts	The platform is appropriate or not for the Introductory Concepts content.
Tests and Conditions	The platform is appropriate or not for the Tests and Conditions content.
Iteration Instructions	The platform is appropriate or not for the Iteration Instructions content.
Functions	The platform is appropriate or not for the Functions content.
Arrays	The platform is appropriate or not for the Arrays content.
String	The platform is appropriate or not for the Strings content.
Pointers	The platform is appropriate or not for the Pointers content.
Recursion	The platform is appropriate or not for the content of Recursion.
Files	The platform is appropriate or not for the Files content.
Structs	The platform is appropriate or not for the Structs content.

4.1.3 Features Classification

Classification according to the features found on each online platform, such as debugging, low level and high level animation, compilation, automatic evaluation and others, elucidated in Table 4.3.

Table 4.3: Platform features classification

Feature	Characteristic
Debugging	The platform may or may not have Debugging functionality.
Animation	The platform may or may not have High Level or Low Level Animation functionality.
Compilation	The platform may or may not have Compilation functionality.
Automatic Evaluation	The platform may or may not have Automatic Evaluation functionality.
Gamification	The platform may or may not have Gamification concepts.
Colaborative Work	The platform may or may not have Colaborative Work functionality.
Explanatory Tutorials	The platform may or may not have Explanatory Tutorials functionality about the system's usage.
Variable Values	The platform may or may not have functionality to show Variable Values.
Exercises	The platform may or may not have Exercises in its system.

4.2 Classification of the Platforms Collected

The classifications of the platforms were performed through qualitative research on each collected platform, which involved tests and observation, made by the author, to get the results seen in this section. The tables below show the classifications performed for each platform, according to their characteristics and available features, in accordance with the classification model proposed in Section 4.1.

Table 4.4: Results of the platform general classification

Classification	Codeboard	URI	Python / C Tutor	CPuzzles	Coderbyte	CodinGame
Language(s) Supported	C, C++, Eiffel, Haskell, Java, Python, Python3, Reason	C, C++, Go, Haskell, Java 7, Java 8, C#, Javascript	Python, Java, C, C ++, Javascript and Ruby	C	C, C++, Go, Java, Javascript, C#, PHP, Python, Ruby, Swift	C, C++, Java, Javascript, Python3, C#, F#, Go, Haskell, Lua, Objective C, Pascal, Perl, PHP, Ruby, Rust, Scala, VB.NET
Free Access	Yes	Yes	Yes	Yes	In Parts	Yes
Context	Outside the classroom	Inside and Outside the classroom	Inside and Outside the classroom	Inside the classroom	Outside the classroom	Outside the classroom
Quality of the Interface	High	High	Medium	Low	High	High

In general, the platforms collected are multi-language, have free access and reveal interfaces with high quality. Most of them have features considered more important to be explored outside the classroom context.

Table 4.5: Results of the platform content classification

Content	Codeboard	URI	Python/ C Tutor	CPuzzles	Coderbyte	CodinGame
Introductory Concepts	Yes	Yes	No	No	No	No
Tests and Conditions	Yes	Yes	No	No	No	No
Iteration Instructions	Yes	Yes	Yes	Yes	No	Yes
Functions	Yes	No	Yes	No	Yes	Yes
Arrays	Yes	Yes	Yes	Yes	No	Yes
String	Yes	Yes	Yes	Yes	Yes	Yes
Pointers	Yes	Yes	Yes	Yes	No	No
Recursion	Yes	Yes	Yes	No	Yes	Yes
Files	No	No	No	No	No	No
Structs	Yes	Yes	Yes	Yes	No	No

None of the platforms collected cope with Files content. Only *Codeboard* and *URI Online Judge* systems work with Introductory Concepts and Tests and Conditions content. Moreover, some of the platforms don't work with Functions, Pointers and Structs.

Table 4.6: Results of the platform features classification

Feature	Codeboard	URI	Python/ C Tutor	CPuzzles	Coderbyte	CodinGame
Debugger	No	No	Yes	No	No	No
Animation	No	No	Low Level	No	No	High Level
Compilation	Yes	No	Yes	No	Yes	Yes
Automatic Evaluation	Yes	Yes	No	No	Yes	Yes
Gamification	No	Yes	No	No	Yes	Yes
Collaborative Work	Yes	No	Yes	No	No	No
Explanatory Tutorials	Yes	Yes	No	No	No	Yes
Variable Values	No	No	Yes	No	No	No
Exercises	No	Yes	No	Yes	Yes	Yes

According to the classification above, only *Python/C Tutor* includes Debugging system and shows Variable Values. *CodinGame* and *Python/C Tutor* have Animation feature and most of them have Exercises available.

The results of these classifications are available online [32]. Therefore, teachers can better visualize in this website each platform classified and their characteristics.

Chapter 5

EasyCoding: Automation System for Generating Activity Guides

Teaching programming can be a challenging and complex task. According to Moreno [33], in addition to knowing the basic instructions, the students need to know how to use them in an adequate and creative way to solve a specific problem.

In order to achieve this, students should practice more programming beyond the classroom context. Furthermore, teachers also need some support to be able to propose the appropriate set of motivational activities along the semester and get feedback from the students. To accomplish these objectives, activity guides using innovative web platforms as teaching/learning methodology can be a great help. Moreover, the exercises must be carefully chosen to better support students learning and motivate them in programming more. When searching the web, it is difficult to find these activity guides already constructed.

After collecting and classifying innovative online platforms that better suit the objectives of the methodology being created to support the programming teaching in Chapters 3 and 4, we observed the need to generate activity guides using a joint of them on a platform for teachers. Considering this need, we evaluated a way to automate the creation of these activity guides to simplify the search for more innovative activities by teachers

to help motivate their students and consequently combat the school dropout in the first semesters of the technological courses.

This chapter elucidates the system developed for automatic activity guide generation, by explaining its proposal and implementation. In the end of this chapter, it is exposed how the activity guides were modeled. The *EasyCoding* system is available online [34].

5.1 System Proposal

The automation system for generating activity guides developed - *EasyCoding* - has as main objective to assist teachers, that use C programming language as a teaching method in programming disciplines. The system also aims to assist students in the execution of the activities proposed within the online programming platforms used.

To achieve this goal, the system proposes the automatic generation of activity guides, by making the teacher choose his/her preferences inside the system through a dynamic form. The guides are separated into guides for students and guides for teachers, in order to assist them in the correct use of the online platforms chosen and classified in this study. All these web platforms have important and motivating features like animations, chats, gamification concepts, submission platforms and collaborative programming, that can turn the programming activity more attractive and intuitive if they are correctly explored.

Moreover, these tools can be combined to join the best of each one. As some of the collected platforms don't have their own set of exercises, it is possible to join them and create different types of activity guides where the teacher and the students can experience different features and methodologies to teach and learn programming.

When generating activity guides automatically, through the *EasyCoding* platform, the teacher will have access, within the teacher's guide, to explanatory instructions on how he/she should use the online platforms to correctly apply the exercises chosen to the students. Moreover, within the students' guide, there will be explanations on how the student should proceed to complete the proposed activities inside the platforms. In other

words, the activity guides will have all the necessary explanation, such as registration on the platforms, how to submit exercises and solutions and so on, for teachers and students.

In addition to the guides, the *EasyCoding* system is capable of generating examples of solutions for the exercises that are available in the platform. These solutions aim to further assist the teacher while correcting the exercises submitted by the students.

In the case of generating activity guides using the *Codeboard* or *URI Online Judge* platforms, the teacher will also have the possibility to download a VBA (Visual Basic for Applications language) algorithm, implemented by the author, that will enable him/her to carry out important information and statistical analysis about the students' results within the *Microsoft Excel* software. This is made possible by the ability of these two online platforms to export all student results in spreadsheet format (CSV file).

Therefore, the teacher can upload the document generated by these online platforms inside the *Microsoft Excel* software and add the VBA algorithm, downloaded on *EasyCoding* platform, inside the *Excel Visual Basic Editor* - which allows users to create and edit VBA codes that can automate tasks inside Office applications. After running the algorithm inside the editor, the teacher will be able to observe and analyze the students results and statistics generated by it.

In the case of generating activity guides using the *Codeboard* platform, the teacher will be able to access an automatic assessment code, implemented by the author using the *Automatic Grading by Result String* concept from *Codeboard* system, for each exercise available for this platform in the *EasyCoding* system. This code will assist the teacher in evaluating students according to their code submissions inside *Codeboard* platform and grading them automatically inside the *Codeboard* submission page.

In summary, with *EasyCoding* system, the teacher will be able to join some of the online platforms, classified in this work, within the activity guides and will be capable of generating them automatically for teachers and students to assist them in completing the activities. This system can improve the programming learning/teaching and help motivate the students in their first semesters of technology courses, as they will have access to innovative and attractive online programming tools.

5.2 System Implementation

In this section, we expose the implementation of *EasyCoding* system. For this objective, it will be explained the system's architecture, structure and technologies used.

5.2.1 System Architecture

In order to make it possible to implement the *EasyCoding* system, according to the objectives and features mentioned in the proposal above, we observed that it is a system with a non-dynamic content, as its proposal is to work in a form format, where the teacher is able to choose his/her preferences to generate the activity guides. Therefore, it was opted for the *EasyCoding* system to have a static architecture, which can be seen in Figure 5.1.

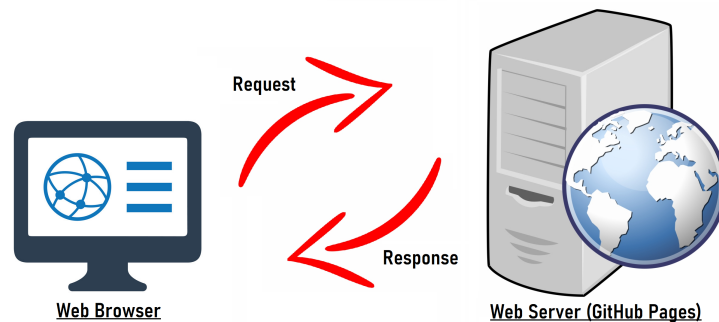


Figure 5.1: Architecture of *EasyCoding* system.

The *EasyCoding* system was implemented using mainly HTML and JavaScript files, which are made available on the user's web browser through the Web Server of *GitHub Pages* [35] - better explained in Section 5.2.3 - and published through *GitHub* - a source code host platform [36]. As a static website, *EasyCoding* system can be easily accessed online by any teacher through its link [34] in any browser. Moreover, the user will obtain a quick response from the Web Server while generating the activity guides, due to *GitHub Pages* velocity.

5.2.2 System Structure

The structure of the *EasyCoding* system was developed based on the needs exposed in the proposal above. To comply with the defined objectives, the developed structure can be seen in Figures 5.2 and 5.3, where the Use Case Diagram and the Activities Diagram of the *EasyCoding* system are shown, respectively.

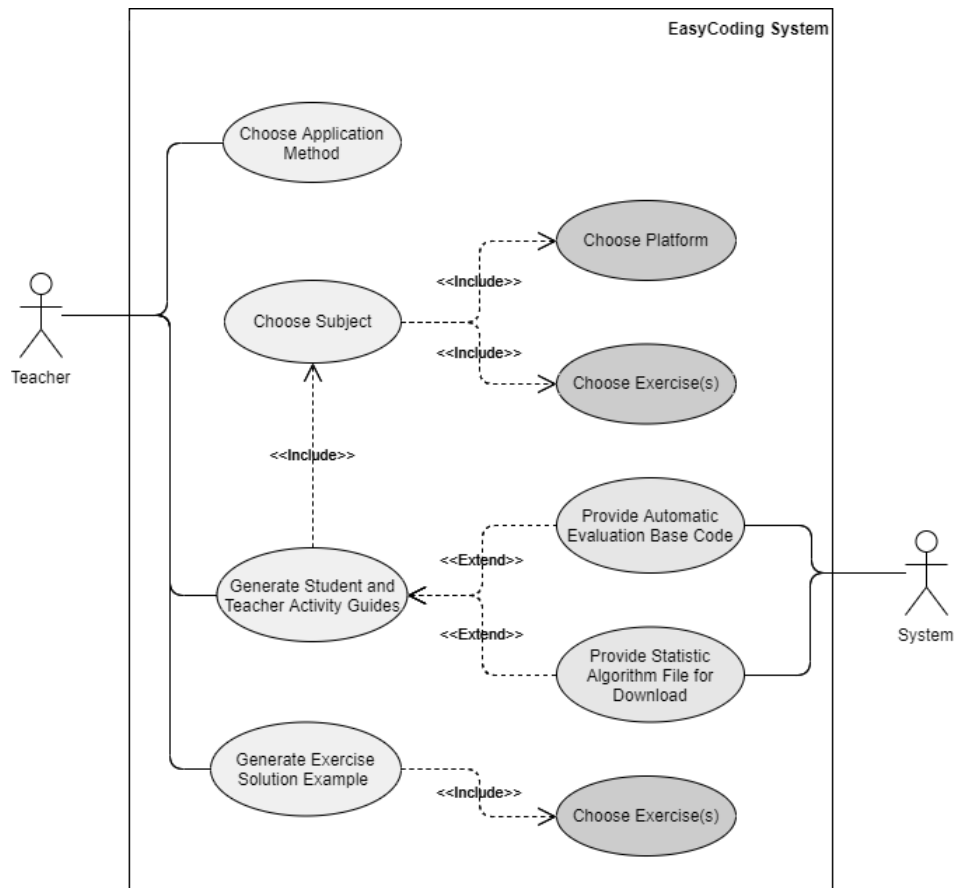


Figure 5.2: Use Case Diagram of *EasyCoding* system.

In the Use Case Diagram above, it is possible to visualize that the system has only one type of end user, the teacher. He/she will have available, within the system, the capability of:

1. Choosing the application method of the activity, which can be inside or outside the classroom;

2. Choosing the subject, according to Programming I and Programming II syllabus;
3. Choosing the platform where the activities will be done, according to a group of the platforms collected and classified in this study;
4. Choosing the exercises themselves;
5. Generating exercise solution examples;
6. Having access to automatic assessment code, when the platform chosen is *Codeboard*;
7. Having access to a VBA code file for statistical analysis of the students results when the platform chosen is *Codeboard* or *URI Online Judge*;
8. Generating student and teacher activity guides.

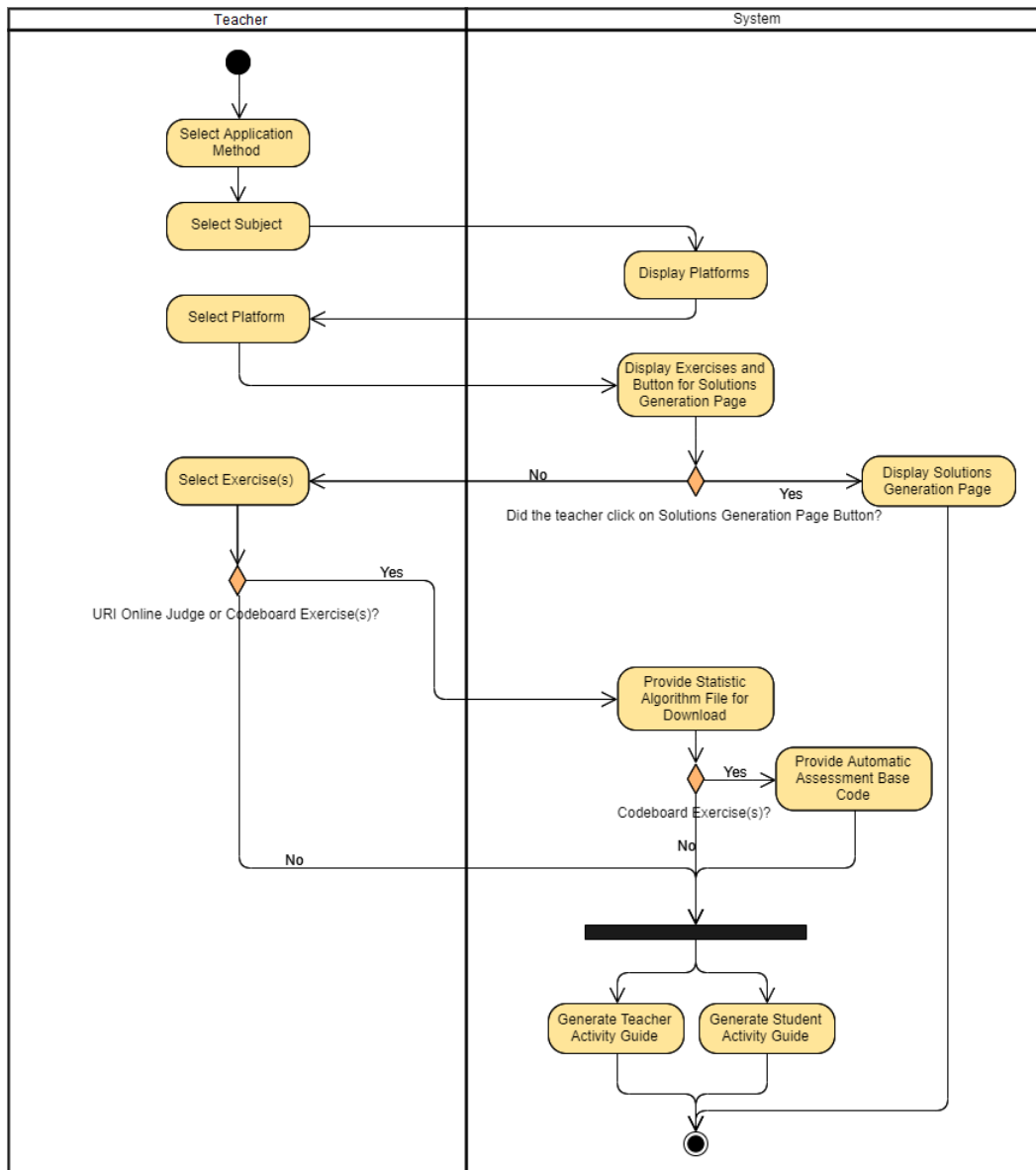


Figure 5.3: Activity Diagram of activity guides generation in *EasyCoding* system.

In the Activity Diagram above, it is possible to observe the activities' flow according to the teacher choices inside *EasyCoding* platform. This choices can be done inside the dynamic form available in *EasyCoding* system.

In order for the teacher to have control and autonomy over the activity guide that he/she pretends to generate, the main screen of the system consists of a brief explanation of the objectives and functionalities available. This explanation is followed by a dynamic

form, which changes according to the teacher's selections, such as chosen subject and chosen platform. In Figure 5.4, it is possible to see the main page of the *EasyCoding* system.

EasyCoding

Metodologia de Suporte à Aprendizagem de Programação

Plataforma criada para geração automática de Guiões de Atividades aos professores de Programação I e II, com ensino da Linguagem C.

Objetivo

O presente trabalho tem como principal objetivo propor uma metodologia de ensino, com a utilização de plataformas web inovadoras e atrativas, que permita melhorar os resultados de aprendizagem dos alunos e os motivem a praticar mais exercícios em suas primeiras disciplinas de programação, para além da sala de aula.

Para isso, esse sistema é capaz de gerar automaticamente guiões de atividade (aos alunos e professores), com o objetivo de explicar a utilização das plataformas web para os exercícios escolhidos, dentro ou fora de sala de aula.

Os exercícios propostos foram recolhidos das plataformas web estudadas e também do livro Linguagem C, de Luis Damas - 10ª Edição.

Preencha o formulário de acordo com o Guião de Atividades que deseja obter:
Para habilitar as opções 3 e 4, primeiro seleccione a Aplicação da Atividade e a Matéria.

1) Aplicação da atividade:

Escolha...

2) Matéria (Selecionar apenas uma opção):

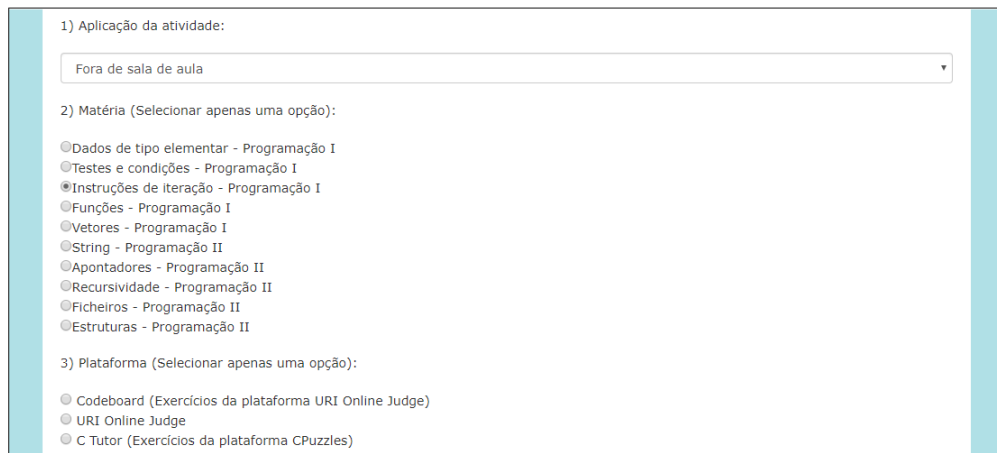
- Dados de tipo elementar - Programação I
- Testes e condições - Programação I
- Instruções de iteração - Programação I
- Funções - Programação I
- Vetores - Programação I
- String - Programação II

Figure 5.4: Screenshot regarding the main page of *EasyCoding* system.

The online platforms used in *EasyCoding* system are a group of the platforms collected and classified in Chapters 3 and 4. To better select which platforms were going to be used, an interview was done with some students from the Polytechnic Institute of Bragança (IPB), after applying these activity guides on the first semester of academic year 2019/2020. In this interview, better described in Section 6.3.2 of this work, it was possible to observe that the most accepted platforms were: *URI Online Judge*, *Codeboard* and *C Tutor*. Therefore, it was opted to add only these platforms, in addition to the *CPuzzles* platform for collecting exercises, in the *EasyCoding* system.

As a dynamic form, when choosing a subject, the form changes according to the platform classifications made in Table 4.5, in Chapter 4. In other words, for each chosen subject, the system will provide platforms (or sets of platforms) that are most suitable for

the selected content. In Figure 5.5, it is possible to see an example of this classification on the platform. In this example, when selecting "Iteration instructions" subject, the *EasyCoding* system shows the platforms *Codeboard*, *URI Online Judge*, *CPuzzles* and *C Tutor* as they were classified in this study as the appropriate ones for this subject.



The screenshot displays a web form with three sections for configuring an activity:

- 1) Aplicação da atividade:** A dropdown menu with the selected option "Fora de sala de aula".
- 2) Matéria (Selecionar apenas uma opção):** A list of radio buttons for subjects:
 - Dados de tipo elementar - Programação I
 - Testes e condições - Programação I
 - Instruções de iteração - Programação I
 - Funções - Programação I
 - Vetores - Programação I
 - String - Programação II
 - Apontadores - Programação II
 - Recursividade - Programação II
 - Ficheiros - Programação II
 - Estruturas - Programação II
- 3) Plataforma (Selecionar apenas uma opção):** A list of radio buttons for platforms:
 - Codeboard (Exercícios da plataforma URI Online Judge)
 - URI Online Judge
 - C Tutor (Exercícios da plataforma CPuzzles)

Figure 5.5: Screenshot regarding the platforms classification inside *EasyCoding* system.

The available platforms can be visualized in *EasyCoding* system alone or together with other platform (a set of platforms), so this additional platform can offer exercises. This happens when the main platform chosen doesn't have exercises available inside its system.

For each platform (or sets of platforms) chosen by the teacher, the system will provide suitable exercises - carefully selected by the author - which are available in the platforms chosen. Thus, the teacher can insert the exercises he/she wants in the activity guide to be generated. It is important to mention that the teacher can select the amount of exercises he/she prefers, from the available ones, to add in the activity guide to be generated. The exercises were taken from the platforms themselves, when they had exercises available, and from the book *Linguagem C*, by Luís Damas, 10th Edition [37].

When the *EasyCoding* system generates the activity guides, the exercises selected will be shown inside the teacher's activity guide together with their estimated time to conclude, as shown in Figure 5.14. The types of exercises available inside the platforms and in the book used in *EasyCoding* platform are:

1. Exercises that ask the student to develop the entire code for the problem;
2. Exercises that ask the student to implement just the function needed to solve the problem - a base code is already given;
3. Exercises that ask the student what does the code implemented write to the monitor;

In Figure 5.6 it is possible to observe each stage of the system's dynamic form. Also, one can see the "Visualize Solutions" button available. After pressing this button, the teacher will be redirected to the *EasyCoding* solutions generation page, which can be seen in Figure 5.12.

Preencha o formulário de acordo com o Guião de Atividades que deseja obter:
Para habilitar as opções 3 e 4, primeiro selecione a Aplicação da Atividade e a Matéria.

1) Aplicação da atividade:
Fora de sala de aula

2) Matéria (Selecionar apenas uma opção):
 Dados de tipo elementar - Programação I
 Testes e condições - Programação I
 Instruções de iteração - Programação I
 Funções - Programação I
 Vetores - Programação I
 String - Programação II
 Apontadores - Programação II
 Recursividade - Programação II
 Estruturas - Programação II

3) Plataforma (Selecionar apenas uma opção):
 URI Online Judge
 URI Online Judge e Codeboard

4) Exercícios (Para escolher mais de um exercício, pressione "Ctrl" enquanto os seleciona e depois clique em "Escolher"):
Visualizar Soluções

1002 - Área do Círculo - Estimativa: 10 minutos
1004 - Produto Simples - Estimativa: 10 minutos
1005 - Média 1 - Estimativa: 10 minutos
1006 - Média 2 - Estimativa: 10 minutos

Escolher

Figure 5.6: Screenshot regarding the full dynamic form of *EasyCoding* system.

After choosing the necessary exercises in the dynamic form, the system will behave in different ways depending on the platform (or sets of platforms) chosen. In the case of *URI Online Judge* or *Codeboard* platforms being chosen, in addition to being able to generate activity guides, the system will also provide the download of a VBA algorithm

for statistical analysis of student results inside the *Microsoft Excel* software. In Figure 5.7, one can see an explanatory message being shown in *EasyCoding* system about the correct use of this algorithm, in addition to the buttons to generate the activity guides and the link to download the VBA algorithm itself.

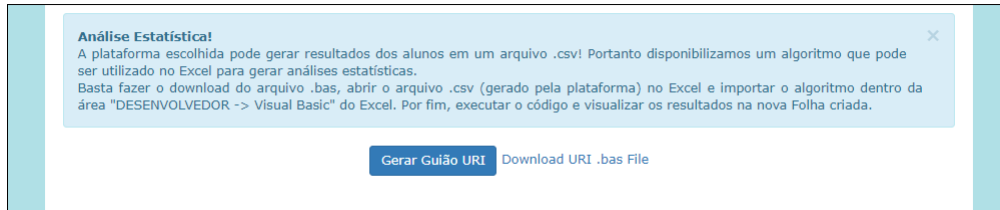


Figure 5.7: Screenshot regarding the message about statistical analysis and the available buttons on *EasyCoding* system.

After downloading the VBA algorithm and adding it inside the *Microsoft Excel's* Visual Basic Editor, the teacher can run the code together with the spreadsheet file acquired in *Codeboard* or *URI Online Judge* platforms that has the students results and visualize the statistical analysis generated. In Figure 5.8 it is possible to see an example of a spreadsheet file with students results generated by *URI Online Judge* platform, after the application of an activity guide to the students.

uri	students	exercises	solved	tried (but not solved)	percentage	score	total_score
405859	Aluno 1	6	4	2	66.666.666.666.667	400	600
405865	Aluno 2	6	6	0	100	600	600
405860	Aluno 3	6	6	0	100	600	600
405924	Aluno 4	4	4	0	66.666.666.666.667	400	600
405908	Aluno 5	4	3	1	50	300	600
405917	Aluno 6	2	2	0	33.333.333.333.333	200	600
405910	Aluno 7	5	5	0	83.333.333.333.333	500	600
405863	Aluno 8	4	4	0	66.666.666.666.667	400	600
405913	Aluno 9	2	2	0	33.333.333.333.333	200	600
405875	Aluno 10	6	5	1	83.333.333.333.333	500	600
405873	Aluno 11	2	2	0	33.333.333.333.333	200	600
405876	Aluno 12	4	4	0	66.666.666.666.667	400	600
405870	Aluno 13	6	6	0	100	600	600
405869	Aluno 14	6	6	0	100	600	600
405912	Aluno 15	3	2	1	33.333.333.333.333	200	600
405867	Aluno 16	3	1	2	16.666.666.666.667	100	600

Figure 5.8: Students results in spreadsheet format generated in *URI Online Judge* platform.

After adding the VBA algorithm downloaded in *EasyCoding* platform and compiling it inside the *Microsoft Excel* software together with the students results, the teacher will be able to visualize, in a new sheet, the statistical results, as shown in the Figures 5.9 and 5.10, regarding the example shown above.

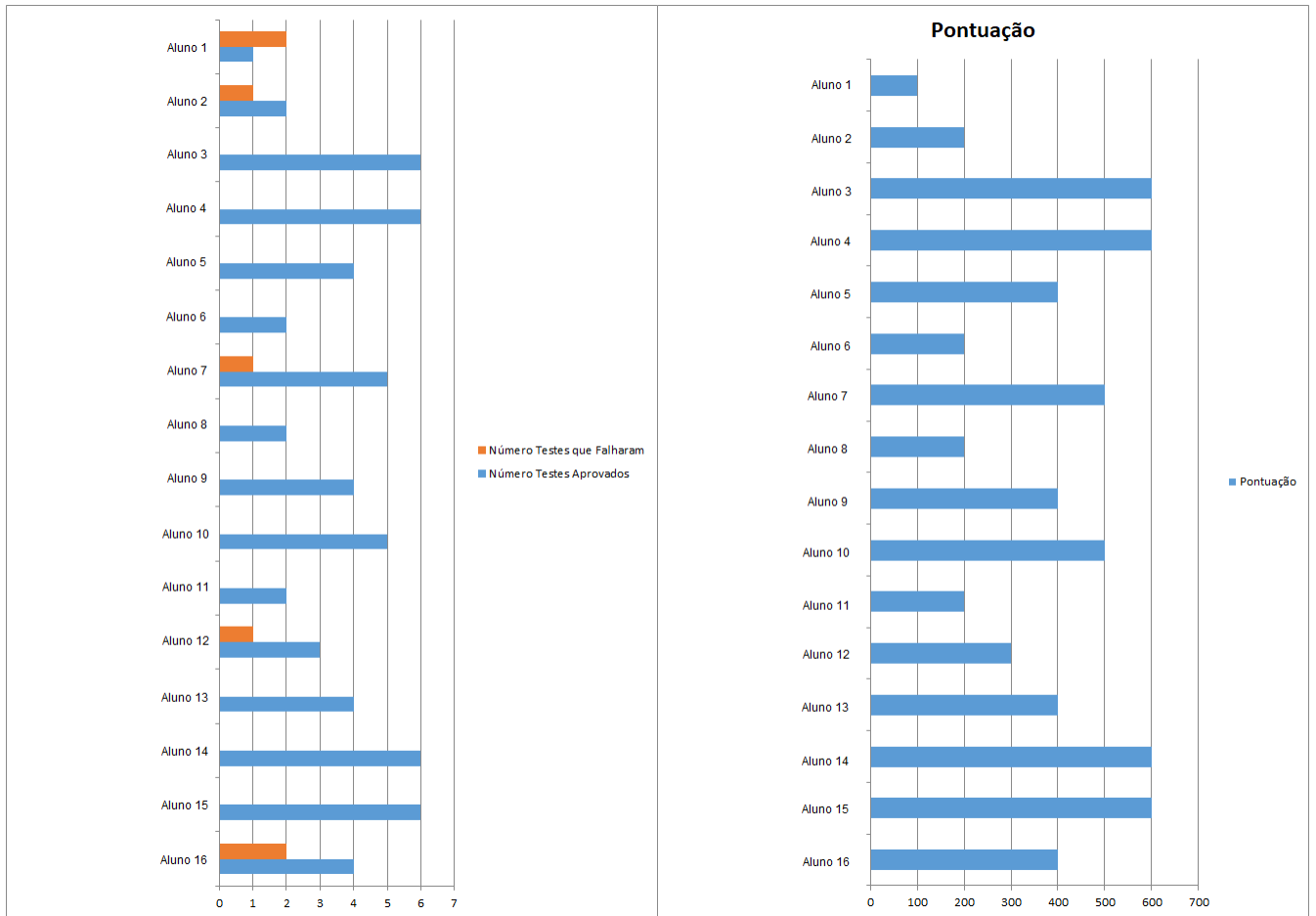


Figure 5.9: Graphs generated regarding the *URI Online Judge* example.

In Figure 5.9 above, it is possible to see graphics regarding the number of tests the students were able to do correctly or not as well as their final score in the activity. Each test concerns an exercise done by the student inside *URI Online Judge* platform. In Figure 5.10 below, it is possible to see students results regarding the number of approved tests and failed tests for each student, besides the average of approved tests, average of failed tests and the total number of students.

	A	B	C	D
1	Username Aluno	Número Testes Aprovados	Número Testes que Falharam	Pontuação
2	Aluno 1	4	2	400
3	Aluno 2	6	0	600
4	Aluno 3	6	0	600
5	Aluno 4	4	0	400
6	Aluno 5	3	1	300
7	Aluno 6	2	0	200
8	Aluno 7	5	0	500
9	Aluno 8	4	0	400
10	Aluno 9	2	0	200
11	Aluno 10	5	1	500
12	Aluno 11	2	0	200
13	Aluno 12	4	0	400
14	Aluno 13	6	0	600
15	Aluno 14	6	0	600
16	Aluno 15	2	1	200
17	Aluno 16	1	2	100
18				
19				
20	Media de Testes Aprovados	3,875		
21	Média de Testes Reprovados	0,4375		
22	Total Alunos	16		

Figure 5.10: Statistical results generated regarding the *URI Online Judge* example.

In the case of *Codeboard* platform being chosen by the teacher, while generating the activity guides for teachers and students, the *EasyCoding* platform will provide, for each exercise selected, a base code and an automatic assessment code to help the teacher implement the activity inside *Codeboard* platform. In Figure 5.11 it is possible to see an example of a base code with automatic assessment of an exercise, provided for the teacher after choosing this exercise and generating the activity guides, inside *EasyCoding* platform.


```
Códigos Base para a plataforma Codeboard

//-----Exercício 4 - Capítulo 5. Página 127 - Livro Linguagem C. Luis Damas. 10 Edição-----

//main.c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    /* Implemente a seguinte função:
    int Abs(int x), que devolve o valor absoluto de x.
    Casos de testes já disponíveis:
    */
    int x = 5, y = -5;

    x = Abs(x);
    y = Abs(y);

    testUnitario(x,y);
    return 0;
}

//test.c
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

void testUnitario(int x, int y){

    double grade = 0;
    int testsPassed = 0;
    int testsFailed = 1;

    if(x == 5 && y == 5){
        testsPassed++;
        testsFailed--;
    }

    if(testsPassed == 1)
        grade = 1;

    printf("<!--@test=%2.2f;%d;%d;-->", grade, testsPassed, testsFailed);
}
}
```

Figure 5.11: Screenshot regarding the source code and automatic assessment code generated by *EasyCoding* system.

On the *EasyCoding* solutions generation page, the teacher will be able to select any exercise available in the system to generate an example of solution. All the solutions were tested by the author. In the case of *CPuzzles* exercises, the solutions were taken from the *CPuzzles* platform itself. In case of *URI Online Judge* exercises, the solutions were done and tested inside the platform, by the author. Moreover, in the case of exercises from the book *Linguagem C* by Luís Damas, 10th Edition, the solutions were taken from the book itself. In Figure 5.12, it is possible to see the *EasyCoding* solution generation page, where the teacher can firstly select the system in which the exercise comes from and after the exercise itself, to generate an example of solution.

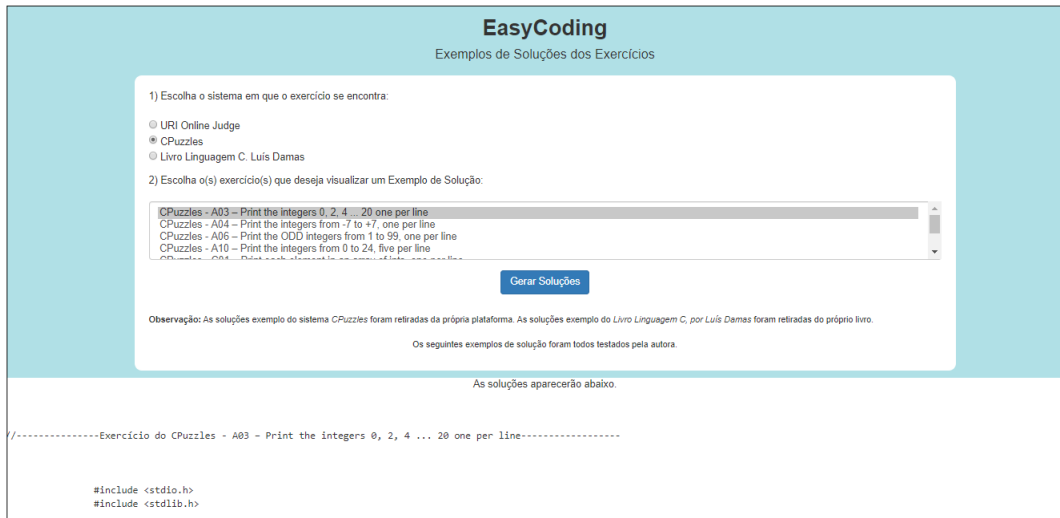


Figure 5.12: Screenshot regarding the solution generation page on *EasyCoding* system.

5.2.3 Technologies Used

For the complete development of the *EasyCoding* system, the following programming languages were used: HTML, JavaScript and Visual Basic for Applications (VBA). For the design and implementation of the main functionalities, the following libraries were used: Bootstrap, JQuery and JsPDF. To host the system, it was used the GitHub Pages, a service for hosting static websites that uses HTML, CSS and JavaScript files directly from the GitHub repository.

HTML

HTML is a standard markup language used for web pages. HTML means HyperText Markup Language and, according to Silva [38], Hypertext can be defined as all content inserted in a document for the web and whose main characteristic is the possibility of linking to other documents on the web.

This markup language made possible the generation of the main page in *EasyCoding* system, with the aim to offer the teacher an assistance inside and outside the classroom, providing the automatic generation of activity guides according to its choices filled in

the system's form. Moreover, HTML made it possible to provide the page for generating sample solutions for the exercises available in the platform.

JavaScript

JavaScript is a programming language for the web with the objective of specifying the behavior of pages, together with HTML - which specify content - and CSS - which define the presentation of websites. This language was developed to run on the client side due to a JavaScript interpreter hosted on client's browser.

According to Silva [39],

JavaScript was created by Netscape in partnership with Sun Microsystems, in order to provide a means of adding interactivity to a web page. The first version, called JavaScript 1.0, was launched in 1995 and implemented in March 1996 on the Netscape Navigator 2.0 browser when the market was dominated by Netscape.

For this project to be implemented, as a static system, JavaScript language was used to develop the main functionalities needed, like the generation of the teacher and student activity guide in PDF format, generation of automatic assessment base codes for *Codeboard* activity guides and to generate solution examples of each exercise available on the *EasyCoding* platform.

The JavaScript libraries used in *EasyCoding* system were *JsPDF* and *JQuery*, as described below:

1. *JsPDF*

To generate activity guides in PDF format, the library *JsPDF* was used on *EasyCoding* platform. According to its website [40], *JsPDF* is a JavaScript open source solution available on GitHub, with the aim of generating PDF's on the client side.

JsPDF can be used by taking its latest version via *Unpkg* link (a global content delivery network) and loading it on the HTML file, which can be seen below.

```
<script src="https://unpkg.com/jspdf@latest/dist/jspdf.min.js"></script>
```

This library can be installed by command-line using Yarn or Npm technologies, which are JavaScript package managers. In case of using frameworks like Angular or React, *JsPDF* can be imported as shown below.

```
import * as jsPDF from 'jspdf'
```

2. *JQuery*

Besides *JsPDF*, the *JQuery* library was used to simplify the scripts implemented in the *EasyCoding* system. This library has an *Event Handling* feature, which is an action the system should take over the users' actions. This concept was essential while implementing the system's logic.

According to *JQuery's* website [41], this JavaScript library is a fast, small, and feature-rich technology. It makes things like HTML document traversal and manipulation, event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.

JQuery can be used via Google Hosted Libraries [42], when loading the library on the HTML file, as shown below.

```
<script  
src="https://ajax.googleapis.com/ajax/libs/jquery/3.4.1/jquery.min.js">  
</script>
```

Visual Basic for Applications (VBA)

The Visual Basic for Applications is an implementation of the Visual Basic language in order to be incorporated on Microsoft Office systems like Excel. According to Microsoft website [43], VBA is effective and efficient when it comes to repetitive solutions to formatting or correcting problems. Thereby, VBA becomes an important language to help analyze all the students results in an optimized way and requiring less time.

The VBA language was used in this project to create algorithms that can be used by the teachers, when they choose a platform in *EasyCoding* system that has the functionality of generating Excel (.CSV) files that contain students data about their results in each activity accomplished. These algorithms contain the logic to generate inside the Excel software a new sheet with important information and statistics about the students results.

Github Pages

GitHub Pages is a hosting service to publish static sites that allows HTML, CSS, and JavaScript files, by taking them from a repository on GitHub. According to its website [35], the GitHub Pages sites created after June 15, 2016 and using "github.io" domains are served over HTTPS support for traffic, adding a layer of encryption that prevents others from snooping on or tampering with the traffic of the website.

GitHub Pages sites hold the following usage limits:

- GitHub Pages source repositories have a recommended limit of 1GB.
- GitHub Pages sites have a soft bandwidth limit of 100GB per month.
- GitHub Pages sites have a soft limit of 10 builds per hour.

The usage limits above were considered adequate for the implementation of *EasyCoding* system, as the tool is proposed for academic purposes, with just one type of end user, the teachers.

5.3 Activity Guide Modeling

The model for the activity guides to be generated automatically inside *EasyCoding* platform was proposed as follows:

1. Two types of activity guides have been specified:
 - (a) Student Activity Guides: Contains clear and illustrated instructions for the student to carry out the activities inside the platforms, without major difficulties.
 - (b) Teacher Activity Guides: Contains clear and illustrated instructions for the teacher to prepare the activities inside the platforms, without major difficulties.
2. Each activity guide contains a header on its first page with important information that describes them, which can be seen in Figure 5.13:

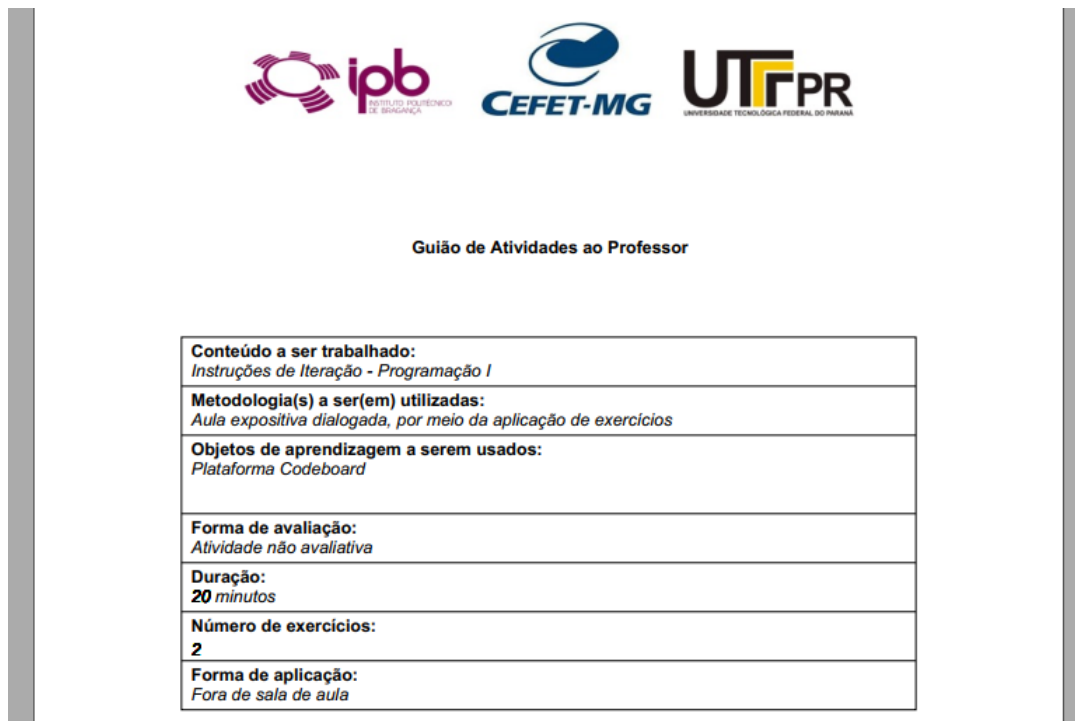


Figure 5.13: Example of the header of an activity guide.

- (a) *Content to be worked on*: Indicates the specific content of the activity, chosen by the teacher, regarding a Programming I or Programming II syllabus.
 - (b) *Methodology to be used*: Indicates the method of exposure of the activity to students.
 - (c) *Learning objects to be used*: Indicates the web platform (or set of platforms) applied to the activity, chosen by the teacher.
 - (d) *Evaluation form*: Indicates whether the activity is evaluative or not. For this study, all guides were considered as non-evaluative activities.
 - (e) *Activity duration*: Indicates the sum of each chosen exercise's duration, in minutes.
 - (f) *Number of exercises*: Indicates the number of exercises, chosen by the teacher.
 - (g) *Application method*: Indicates whether the activity is applied inside or outside the classroom. Information chosen by the teacher.
3. After the header, the body of the activity guide is composed of instructions to students and teachers, duly clarified and illustrated, on how to use the web platforms correctly to fulfill the objectives of the teacher and the student. Figure 5.14 illustrates an example of a teacher activity guide's body, with instructions on using the *URI Online Judge* platform.

1) Entrar na ferramenta URI Online Judge, link a seguir:

<https://www.urionlinejudge.com.br/judge/pt/login>

2) No menu superior da plataforma, clicar em "Problemas" e entrar na opção "Iniciante":



3) Recolher os problemas para os alunos. Sugestão de problemas:

1002 - Área do Círculo - Estimativa: 10 minutos

1005 - Média 1 - Estimativa: 10 minutos

4) Entrar na ferramenta Codeboard e criar uma conta na plataforma, link a seguir:

<https://codeboard.io/>

Figure 5.14: Example of the body of a teacher activity guide.

Besides the explanations on how to use the platforms to provide exercises for the students, the exercises themselves - chosen by the teacher - will be available on the body of teacher's activity guide, as shown in Figure 5.14 above. Thereby, the teacher and the students will be able to complete the activities proposed without major difficulties. The activity guides will provide all the necessary guidance to complete the activities.

It is important to mention that each platform used in the activity guides has its own methodology and different types of use. Some of them require previous registration from the users and each one has different ways of submitting code, for example. Therefore, the activity guides play an important role by explaining each step the teacher and the student should follow to accomplish the activities.

Chapter 6

Application of the Methodology Proposed and Results

6.1 Application Proposal

In order to accomplish the final objectives of this work, we considered a proposal for the application of the activity guides created, which can be seen below:

1. The teacher should be able to have in hands the activity guides (for himself/herself and for the students) by generating them on *EasyCoding* system, according to his/her needs.
2. By reading the teacher's activity guide, the teacher should be able to properly prepare the activity inside the platform chosen.
3. After this preparation (involving registration, submission of exercises and so on), the teacher should apply the exercises by sending the students' activity guide for the students enrolled in his/her class, by academic e-mail, and explaining the activity in the classes.
4. With the activity guide in hands, the students should be able to correctly accomplish the exercises by submitting their solution inside the platforms.

5. After having the students solutions, the teacher should be able to generate some statistics by using the VBA algorithm available in *EasyCoding* system.
6. In the end of this experiment, a questionnaire should be applied to students and teachers to better analyze their experience with this teaching/learning methodology.

6.2 Application of the Activity Guides

The present study considered introductory programming courses (Programming I and II), in the context of the first and second semesters of Higher Education, at the Polytechnic Institute of Bragança (IPB) and at the Federal Technology Education Center of Minas Gerais (CEFET-MG).

At IPB, Programming I is a subject taught in the first semester, and Programming II is taught in the second semester, covering the second part of the programming course syllabus. At the end of these course units, the student is expected to be able to develop medium/high complexity C programs and apply advanced programming knowledge in C language. At CEFET-MG, the C programming language is taught only in the first semester of academic year, in Programming I subject, covering also medium/high complexity C contents.

The IPB and CEFET-MG courses contain C programming language topics that are commonly taught in introductory programming at the university level. As well as other cases found in the literature, Programming I and II also reveal a low student success rate. This is because the teaching-learning programming process can face several challenges. One of the most important is to improve student motivation to enroll the students and make them work outside the context of the classroom.

In Sections 6.2.1 and 6.2.2 it is possible to see the results of the application of the activity guides in the first and second semester of academic year 2019/2020 at the Polytechnic Institute of Bragança - IPB.

In Section 6.2.3 it is shown the results of the application of the activity guides at the Federal Technology Education Center of Minas Gerais - CEFET-MG, in the second

semester of academic year 2019/2020.

6.2.1 Results of the Research Applied on Programming I Course at IPB

In the first semester of academic year 2019/2020 it was applied the proposed approach to hold the students involved in programming tasks. Thus, it was created a set of activity guides that cover the whole syllabus of Programming I.

During this first semester, the research was carried out with students enrolled in the Informatics Engineering and Management Informatics courses, at the Polytechnic Institute of Bragança - IPB. The syllabus of Programming I used in the activity guides, in the first semester of academic year 2019/2020 was:

1. Elementary data types;
2. Tests and conditions;
3. Iteration instructions;
4. Functions;
5. Arrays.

The activity guides held a set of web platforms classified in this study with exercises well contextualized for each syllabus. In the classes, the teacher was oriented to give a brief explanation of each of these guides, whose exercises were solved and submitted by the students inside and outside the classroom.

In the first semester of academic year 2019/2020, it was proposed the activity guides to 65 students that were divided into the two different classes mentioned above. It was decided to involve only these students for two reasons:

1. The teacher taught both classes, so he had full control of the students;

2. The idea was to start just as a pilot project and adjust the guides during the academic year.

However, regarding the students mentioned, only an average of 45 students attended classes, where the teacher emphasized the importance of carrying out the proposed activities.

In Tables 6.1 and 6.2 below, it is possible to see the results regarding the application of the activity guides during the first semester of academic year 2019/2020. On the first column, it is shown the activity guides applied, numbered according to each syllabus taught; On the second column, it is shown the online platforms used; On the third column, its possible to see the exercises applied in each activity guide and their names according to each platform; On the fourth, fifth and sixth column, it is possible to see, respectively, the number of students to whom the activities were proposed, the number of students that submitted the solutions for the exercises and the number of successful submissions.

The Tables 6.1 and 6.2 were taken from the short paper [44], to be published this year in the International Computer Programming Education Conference - ICPEC 2020, regarding this study.

Table 6.1: Results of activity guides application during the first semester of academic year 2019/2020.

Activity Guide	Online Platform(s)	Exercise(s)	N° of Students Attending to Classes	N° of Students Submissions	N° of Correct Submissions
Elementary Data Types					
1.0	Codeboard and URI	From URI Online Judge:			
		1002: Circle Area	45	27	24
		1005: Average 1	45	24	16
		1006: Average 2	45	23	16
Tests and Conditions					
2.0	URI Online Judge	From URI Online Judge:			
		1041: Coordinates of a Point	45	22	18
		1042: Simple Sort	45	22	19
2.1	URI Online Judge	From URI Online Judge:			
		1044: Multiples	45	22	17
		1046: Game Time	45	19	15
Iteration Instructions					
3.0	C Tutor and CPuzzles	From CPuzzles:			
		Puzzle A03	45	24	23
		Puzzle A10	45	24	18
3.1	URI Online Judge	From URI Online Judge:			
		1060: Positive Numbers	45	10	10
		1064: Positives and Average	45	10	6
3.2	CodinGame	From CodinGame: Classic: Temperatures	45	4	3
3.3	Codeboard and URI	From URI Online Judge:			
		1071: Sum of Consecutive Odd	45	9	8
		1132: Multiples of 13	45	8	7

The Table 6.2 is the continuation of Table 6.1.

Table 6.2: 2nd part of the results of activity guides application during the first semester of academic year 2019/2020.

Activity Guide	Online Platform(s)	Exercise(s)	N° of Students Attending to Classes	N° of Students Submissions	N° of Successful Submissions
Functions					
4.0	Coderbyte	From Coderbyte: Time Convert	45	9	9
		First Factorial	45	9	9
4.1	CodinGame	From CodinGame: Classic: The River I	45	1	1
Arrays					
5.0	C Tutor and CPuzzles	From CPuzzles: Puzzle C01	45	3	2
		Puzzle C03	45	3	2
5.1	URI Online Judge	From URI Online Judge: 1173: Fill the Vector I	45	2	1
		1174: Vector Selection I	45	2	0
5.2	CodinGame	From CodinGame: Classic: The Descent	45	2	2

In addition to the general results shown above, it is possible to observe below specific results on each platform used in the study.

Results Regarding the Experience with *URI Online Judge* Platform

The tests carried out in the first semester of academic year 2019/2020 with IPB students, using the *URI Online Judge* platform, showed the following results:

- 8 exercises were requested in total, in *URI Online Judge* platform, 2 per each activity guide applied (Activity Guides 2.0, 2.1, 3.1 and 5.1).
- From the 45 students attending classes at the Polytechnic Institute of Bragança, 39 subscribed at *URI Online Judge's* discipline created by the teacher on *URI Online Judge Academic*.
- From the 39 students subscribed at *URI Online Judge's* discipline, 26 submitted

exercises in the platform.

- The percentage of the number of correct exercises, according to the ones submitted, was 84% in Activity Guide 2.0, 78% in Activity Guide 2.1, 80% in Activity Guide 3.1 and 25% in Activity Guide 5.1.

Results Regarding the Experience with *Codeboard* Platform

The tests carried out in the first semester of academic year 2019/2020 with IPB students, using the *Codeboard* platform, showed the following results:

- 5 exercises were requested in total, in *Codeboard* platform, 3 in the Activity Guide 1.0 and 2 in the Activity Guide 3.3.
- From the 45 students attending classes at the Polytechnic Institute of Bragança, 27 submitted exercises in the *Codeboard* platform.

In Table 6.3, is possible to visualize the results regarding the automatic assessment tests implemented in each activity guides' exercise, inside *Codeboard* platform, in the first semester.

Table 6.3: Results regarding the automatic assessment tests implemented in *Codeboard* platform, in the first semester.

Activity Guide	Exercise	Number of Automatic Tests	Average of Tests Passed	Average of Tests Failed
1.0	1002: Circle Area	3	75%	25%
1.0	1005: Average 1	3	51.3%	48.7%
1.0	1006: Average 2	3	47.7%	52.3%
3.3	1071: Sum of Consecutive Odd	1	33%	66%
3.3	1132: Multiples of 13	1	63%	37%

Results Regarding the Experience with *C Tutor* Platform

The tests carried out in the first semester of academic year 2019/2020 with IPB students, using the *C Tutor* platform, showed the following results:

- 4 exercises were requested in total, in *C Tutor* platform, 2 in the Activity Guide 3.0 and 2 in the Activity Guide 5.0.
- From the 45 students attending classes at the Polytechnic Institute of Bragança, 24 submitted exercises in the *C Tutor* platform.
- The percentage of the number of correct exercises, according to the ones submitted, was 91% in Activity Guide 3.0 and 67% in Activity Guide 5.0.

Results Regarding the Experience with *CodinGame* Platform

The tests carried out in the first semester of academic year 2019/2020 with IPB students, using the *CodinGame* platform, showed the following results:

- 3 exercises were requested in total, in *CodinGame* platform, 1 in the Activity Guide 3.2, 1 in the Activity Guide 4.1 and 1 in Activity Guide 5.2.
- From the 45 students attending classes at the Polytechnic Institute of Bragança, 5 submitted exercises in the *CodinGame* platform.
- The percentage of the number of correct exercises, according to the ones submitted, was 75% in Activity Guide 3.2, 100% in Activity Guide 4.1 and 100% in Activity Guide 5.2.

Results Regarding the Experience with *Coderbyte* Platform

The tests carried out in the first semester of academic year 2019/2020 with IPB students, using the *Coderbyte* platform, showed the following results:

- 2 exercises were requested in total, in *Coderbyte* platform in the Activity Guide 4.0.
- From the 45 students attending classes at the Polytechnic Institute of Bragança, 9 submitted exercises in the *Coderbyte* platform.
- The percentage of the number of correct exercises, according to the ones submitted, was 100% in the Activity Guide 4.0.

6.2.2 Results of the Research Applied on Programming II Course at IPB

In the second semester of academic year 2019/2020, it was applied, at the Polytechnic Institute of Bragança, a set of activity guides that partially covered the syllabus of Programming II. During this semester, the research was carried out with the students enrolled in the Informatics Engineering course.

The *Files* Programming II content was not covered in this research due to the classified platforms lack of compatibility with this content. Moreover, the *Structs* Programming II content was not covered in this research as it was the last content given in the second semester of academic year 2019/2020, therefore there was not enough time to apply this content to students before the completion of this study.

The syllabus of Programming II used in the activity guides, in the second semester of academic year 2019/2020 were:

1. Strings;
2. Pointers;
3. Recursion;

In the second semester of academic year 2019/2020, it was proposed the activity guides to 112 students that were divided into three different classes from Informatics Engineering course at IPB. However, regarding the students mentioned, only an average of 55 students attended classes, where the teacher emphasized the importance of carrying out the proposed activities.

In Table 6.4 below, it is possible to see the results regarding the application of the activity guides during the second semester of academic year 2019/2020 with IPB students. On the first column, it is shown the activity guides applied, numbered according to each syllabus taught; On the second column, it is shown the online platforms used; On the third column, its possible to see the exercises applied in each activity guide and their names

according to each platform; On the fourth, fifth and sixth column, it is possible to see, respectively, the number of students to whom the activities were proposed, the number of students that submitted the solutions for the exercises and the number of successful submissions.

Table 6.4: Results of activity guides application during the second semester of academic year 2019/2020.

Activity Guide	Online Platform(s)	Exercise(s)	N° of Students Attending to Classes	N° of Students Submissions	N° of Correct Submissions
Strings					
6.0	URI Online Judge	From URI Online Judge: 1238: Combiner	55	28	22
		1253: Caesar Cipher	55	27	21
6.1	URI Online Judge	From URI Online Judge: 1257: Array Hash	55	22	19
6.2	Codeboard	From Luís Damas book: Ex. 3.1, Ch 7 - Page 171	55	21	14
6.3	C Tutor	From Luís Damas book: Ex. 2.1, Ch 7 - Page 171	55	15	15
6.4	URI Online Judge	From URI Online Judge: 1871: Zero means Zero	55	16	13
		1873: Rock-paper-scissors	55	17	11
Pointers					
7.0	C Tutor	From Luís Damas book: Ex. 3, Ch 8 - Page 200	55	10	10
		Ex. 4, Ch 8 - Page 200	55	10	10
7.2	Codeboard	From Luís Damas book: Ex. 8, Ch 8 - Page 201	55	4	3
		Ex. 9, Ch 8 - Page 201	55	4	3
7.3	C Tutor and CPuzzles	From CPuzzles: Puzzle F19	55	8	8
		Puzzle F20	55	8	8
Recursion					
8.0	C Tutor	From Luís Damas book: Ex. 4, Ch 9 - Page 226	55	4	4
		Ex. 5, Ch 9 - Page 226	55	4	4
8.1	C Tutor	From Luís Damas book: Ex. 6, Ch 9 - Page 227	55	3	3

In addition to the general results shown above, it is possible to observe below specific results on each platform used in the study.

Results Regarding the Experience with *URI Online Judge* Platform

The tests carried out in the second semester of academic year 2019/2020 with IPB students, using the *URI Online Judge* platform, showed the following results:

- 5 exercises were requested in total, in *URI Online Judge* platform, 2 in the Activity Guides 6.0 and 6.4 and 1 exercise in the Activity Guide 6.1.
- From the 55 students attending classes at the Polytechnic Institute of Bragança, 50 subscribed at *URI Online Judge's* discipline created by the teacher on *URI Online Judge Academic*.
- From the 50 students subscribed at *URI Online Judge's* discipline, 29 submitted exercises in the platform.
- The percentage of the number of correct exercises, according to the ones submitted, was 78.2% in Activity Guide 6.0, 86.4% in Activity Guide 6.1 and 72.7% in Activity Guide 6.4.

Results Regarding the Experience with *Codeboard* Platform

The tests carried out in the first semester of academic year 2019/2020 with IPB students, using the *Codeboard* platform, showed the following results:

- 3 exercises were requested in total, in *Codeboard* platform, 1 in the Activity Guide 6.2 and 2 in the Activity Guide 7.2.
- From the 55 students attending classes at the Polytechnic Institute of Bragança, 22 submitted exercises in the *Codeboard* platform.

In Table 6.5, is possible to visualize the results regarding the automatic assessment tests implemented in each activity guides' exercise, inside *Codeboard* platform, in the second semester.

Table 6.5: Results regarding the automatic assessment tests implemented in *Codeboard* platform, in the second semester.

Activity Guide	Exercise	Number of Automatic Tests	Average of Tests Passed	Average of Tests Failed
6.2	Ex. 3.1, Ch 7 - Page 182	1	66.7%	33.3%
7.2	Ex. 8, Ch 8 - Page 201	1	75%	25%
7.2	Ex. 9, Ch 8 - Page 201	1	75%	25%

Results Regarding the Experience with *C Tutor* Platform

The tests carried out in the second semester of academic year 2019/2020 with IPB students, using the *C Tutor* platform, showed the following results:

- 8 exercises were requested in total, in *C Tutor* platform, 1 in the Activity Guide 6.3, 2 in the Activity Guides 7.0, 7.3 and 8.0 each, and 1 in the Activity Guide 8.1.
- From the 55 students attending classes at the Polytechnic Institute of Bragança, 15 submitted exercises in the *C Tutor* platform.
- The percentage of the number of correct exercises, according to the ones submitted, was 100% in all the Activity Guides mentioned above.

Besides the study with the students from the Polytechnic Institute of Bragança, in the second semester, a research was done with some students from the Federal Technology Education Center of Minas Gerais, which can be seen in next section.

6.2.3 Results of the Research with CEFET-MG Students

In the second semester of academic year 2019/2020, a research was carried out with 44 students enrolled in Computer Engineering course, from the Federal Technology Education

Center of Minas Gerais (CEFET-MG), who were enrolled in Programming I discipline. For this research to be performed, the activity guides were directly delivered to students by the author, due to the COVID-19 pandemic, which started in the first semester of the year 2020, causing an academic break at CEFET-MG and making impossible the communication with the institution's teachers. Moreover, as the communication with the students became more difficult, it is possible that not all students received the proposed activities.

In Table 6.6 it is possible to see the results regarding the application of the activity guides during the second semester of academic year 2019/2020. On the first column, it is shown the activity guides applied, numbered according to each syllabus taught; On the second column, it is shown the online platforms used; On the third column, its possible to see the exercises applied in each activity guide and their names according to each platform; On the fourth, fifth and sixth column, it is possible to see, respectively, the number of students to whom the activities were proposed, the number of students that submitted the solutions for the exercises and the number of successful submissions.

Table 6.6: Results of the research with the students from CEFET-MG.

Activity Guide	Online Platform(s)	Exercise(s)	N° of Students	N° of Students Submissions	N° of Correct Submissions
Elementary Data Types					
1.0	URI Online Judge	From URI Platform: 1004: Simple Product	44	11	8
		1007: Difference	44	11	8
Tests and Conditions					
2.0	URI Online Judge	From URI Platform: 1041: Coordinates of a Point	44	10	6
		1042: Sort Simples	44	10	6
2.1	URI Online Judge	From URI Platform: 1044: Multiples	44	7	6
		1046: Game Time	44	7	6
Iteration Instructions					
3.0	URI Online Judge	From URI Platform: 1060: Positive Numbers	44	7	5
		1064: Positives and Average	44	7	5
3.1	C Tutor and CPuzzles	From CPuzzles Platform: Puzzle A03	44	6	6
		Puzzle A04	44	6	6
3.2	C Tutor and CPuzzles	From CPuzzles Platform: Puzzle A06	44	6	6
		Puzzle A10	44	6	6

In addition to the general results shown above, it is possible to observe below specific results on each platform used in the study.

Results Regarding the Experience with *URI Online Judge* Platform

The tests carried out using the *URI Online Judge* platform showed the following results:

- 8 exercises were requested in total, in *URI Online Judge* platform, 2 per each activity guide applied (Activity Guides 1.0, 2.0, 2.1 and 3.0).
- From the 44 students enrolled in the programming course of the Federal Technology

Education Center of Minas Gerais, 14 subscribed at *URI Online Judge's* discipline created by the author on *URI Online Judge Academic*.

- From the 14 students subscribed at *URI Online Judge's* discipline, 11 submitted exercises in the platform.
- The percentage of the number of correct exercises, according to the ones submitted, was 72.7% in Activity Guide 1.0, 60% in Activity Guide 2.0, 85.7% in Activity Guide 2.1 and 71.4% in Activity Guide 3.0.

Results Regarding the Experience with *C Tutor* Platform

The tests carried out using the *C Tutor* platform showed the following results:

- 4 exercises were requested in total, in *C Tutor* platform, 2 per each activity guide applied (Activity Guides 3.1 and 3.2).
- From the 44 students enrolled in the programming course of the Federal Technology Education Center of Minas Gerais, 6 submitted exercises in the C Tutor platform.
- The percentage of the number of correct exercises, according to the ones submitted, was 100% in the Activity Guides 3.1 and 3.2.

6.3 Analysis of Results

After the results shown above, it was possible to notice that the number of students submissions and successful submissions decreases as the syllabus get more complex. This result can be explained owing to the fact that students have more difficulty to do more complex exercises alone. Moreover, since these activities are not mandatory and they start to be very busy and worried with final exams, just a few number of students did the last activity guides. Furthermore, the small amount of submissions from CEFET-MG students can be explained by the lack of communication with them, due to the COVID-19 pandemic and the institutions academic break. However, to better evaluate these results,

a survey was conducted with students using questionnaires and interview method, which can be seen in the next section.

6.3.1 Evaluation Methods

To evaluate the students' results in the research carried out using the activity guides created, evaluation methods must be present in the scope of this study. The quantitative results obtained in Section 6.2 are important for measuring the number of students who were committed to carrying out the activities, using new methodologies mainly outside the classroom context. However, quantitative results alone are not enough to analyze the experience that students had with each online platform used when performing the activity guides. In this sense, the application of qualitative research methods becomes necessary to better evaluate these results.

According to Leitão and Prates [45], qualitative methods are inherently exploratory and are always guided by an open study question, which exclude questions that aim to confirm hypotheses, expectations and previous knowledge. If the researcher wants to explore an unknown scenario to obtain in-depth information about a certain context, he/she will find strong subsidies in this type of methodology. In this sense, the use of qualitative methods like interviews and questionnaires in this research becomes important to better understand and interpret the opinions related to each student and teacher participating in this study.

Interviews allow you to have direct communication with people who are relevant for research or system design and evaluation. They are especially useful for collecting meanings related to aspects that cannot be captured by direct observation [46].

In this study, a *free interview* was conducted at the end of the first semester of academic year 2019/2020 with the students from IPB. The *free interview*, according to Leitão and Prates [45], does not follow any pre-established script but is guided by the main research question and is appropriate at the beginning of the study, when one wishes to explore more about the topic.

The questionnaire is a written data collection instrument, which can be used in qualitative and quantitative research, to collect opinion, measure attitudes and gather data and meanings about a given phenomenon through a coherent set of questions or statements [47].

In this study, questionnaires were applied at the end of the academic year 2019/2020 with the students and teachers from IPB and with the students from CEFET-MG.

6.3.2 Interview Results

On day 09 of January, 2020, in the end of the first semester of academic year 2019/2020, a *free interview* was conducted with 4 students from IPB, enrolled in Informatics Engineering course, who were willing to represent their class in order to provide their experience about the use of the activity guides developed for the methodology created in this study, throughout the entire the Programming I course.

The *free interview* lasted around 30 minutes and it was questioned issues regarding the platforms used. The topics discussed were:

1. Topics regarding the activity guides sent to the students:

- If the activity guides proved to be sufficient for the correct execution of the exercises.

Results: For the majority of the students, the activity guides used were enough for the correct execution of the activities.

- If the exercises proposed were related to the learned content.

Results: All the students thought that the exercises proposed were related to the learned content.

2. Topics regarding the online platforms used:

- If the platforms were accessible, easy to use and had quick response.

Results: The students described the majority of the platforms tested as easy to

use and with quick response. However, they had initial difficulties while using *CodinGame* platform, on understanding what to do in each exercise proposed, inside its IDE. The students also mentioned that *Coderbyte* platform is not accessible, as its exercises are not totally free to use.

- If the platforms used were considered important systems to motivate the students in programming more.

Results: All the students agreed that the platforms used motivated them to program more outside the classroom context.

- Which platform(s), from the ones experienced, the students thought to be more important in students motivation while practicing more programming outside the classroom context.

Results: The students agreed that the platforms *URI Online Judge*, *Codeboard* and *C Tutor* made an important role in motivating more the students while practicing programming, besides them being accessible and easier to use.

The last topic discussed in the interview was questioning how could we motivate more the students that did not collaborate in this study. The students answered that it could be possible to give them more information regarding the platforms and their characteristics, showing what they would be able to do inside them. Moreover, the students claimed that the activities could be evaluated with additional scores in their courses.

The interview results could show that, according to the opinion of IPB students, the use of online platforms in teaching programming methodologies can indeed motivate them to practice more outside the classroom context and, consequently, make them more committed with the course.

As some of the online platforms used in the first semester of academic year 2019/2020 were not well evaluated by the students, it was opted not to continue with the application of *CodinGame* and *Coderbyte* platforms in the second semester.

6.3.3 Questionnaires Results

In order to better evaluate the students' results and understand the teachers' and students' opinions about the experience they had with the new methodology developed in this work, questionnaires were used as a form of opinion research. Through these questionnaires, the following results were obtained:

Results Regarding the Questionnaire Applied to IPB Students

The questionnaire to students at the Polytechnic Institute of Bragança was applied after the end of the academic year 2019/2020, as a way to understand their opinions about the teaching/learning methodology implemented. The questionnaire available in *Appendix A* was applied to the students from Informatics Engineering course, who were willing to answer it after having experience with the methodology created in this study.

The results of this questionnaire can be seen, according to each question in the *Appendix A*, in Tables 6.7 and 6.8 below. In total, 37 students that experienced this methodology answered the questionnaire below. On the first column, it is shown the question number of the questionnaire; On the second column, it is shown the questions' alternatives; On the third column, its possible to see the frequency of the alternative chosen - the amount of students who chose this answer; On the fourth column it is shown the percentage of the students who chose this alternative.

Table 6.7: Results of the questionnaire from *Appendix A* applied to IPB students, in the second semester of academic year 2019/2020.

Students from IPB in second semester of academic year 2019/2020			
Question Number	Alternatives	Frequency	Percentage
1	Completed all the activities	4	10.8%
	Completed most of the activities	24	64.9%
	Completed few activities	9	24.3%
2.1	Totally Disagree	0	0%
	Partly Disagree	2	5.4%
	Neutral	8	21.6%
	Partly Agree	18	48.6%
	Totally Agree	9	24.3%
2.2	Totally Disagree	0	0%
	Partly Disagree	1	2.7%
	Neutral	9	24.3%
	Partly Agree	16	43.2%
	Totally Agree	11	29.7%
2.3	Totally Disagree	0	0%
	Partly Disagree	2	5.4%
	Neutral	7	18.9%
	Partly Agree	19	51.4%
	Totally Agree	9	24.3%
2.4	Totally Disagree	1	2.7%
	Partly Disagree	4	10.8%
	Neutral	18	48.6%
	Partly Agree	13	35.1%
	Totally Agree	1	2.7%

Table 6.8: 2nd part of the results of the questionnaire from *Appendix A* applied to IPB students.

Students from IPB in second semester of academic year 2019/2020			
Question Number	Alternatives	Frequency	Percentage
2.5	Totally Disagree	0	0%
	Partly Disagree	1	2.7%
	Neutral	1	2.7%
	Partly Agree	9	24.3%
	Totally Agree	26	70.3%
2.6	Totally Disagree	1	2.7%
	Partly Disagree	1	2.7%
	Neutral	8	21.6%
	Partly Agree	17	45.9%
	Totally Agree	10	27%
3	Codeboard	22	59.5%
	URI Online Judge	24	64.9%
	C Tutor	19	51.4%
	Coderbyte	2	5.4%
	CodinGame	4	10.8%
4	None of them	1	2.7%
	Codeboard	15	40.5%
	URI Online Judge	16	43.2%
	C Tutor	13	35.1%
	Coderbyte	3	8.1%
	CodinGame	8	21.6%
5	None of them	5	13.5%
	Codeboard	7	18.9%
	URI Online Judge	7	18.9%
	C Tutor	22	59.5%
	Coderbyte	0	0%
	CodinGame	3	8.1%
6	None of them	5	13.5%
	Inside the classroom	12	32.4%
6	Outside the classroom	25	67.6%

For question number 7, some students described that, to motivate students who did not collaborate with this project, it could be possible to show them more advantages of using the platforms, presenting their main functionalities, in order to make activities easier to understand. In addition, an option very spoken was to offer the students an additional

value in the final grade of the course, as a way to reward students who completed the exercises.

For question number 8, as a suggestion of improving the activity guides, some students described that informing more about the peculiarities of each platform would help them better execute the activities inside them.

When analyzing the results regarding the questionnaire applied to IPB students about their experience with the methodology created, throughout the academic year 2019/2020, it can be observed that the most complete and accepted platforms for the objective of this study were *Codeboard*, *URI Online Judge* and *C Tutor*. Furthermore, the *C Tutor* platform was considered the best system to discover errors and correct them. The *URI Online Judge* platform proved to be more useful to students and should be the best ally for programming beginners. Finally, the application method of the activity guides considered most useful for learning programming was outside the classroom.

Moreover, it is possible to see that the students experienced some difficulties in carrying out the proposed activities. This result may have occurred, as they had the opportunity to work with more complex exercises throughout the academic year 2019/2020, with subjects from Programming I and II course.

In general, the activity guides proposed were considered sufficient for the execution of the exercises. The application of these exercises indeed motivated students in practicing more programming exercises beyond the classroom context.

Results Regarding the Questionnaire Applied to IPB Teachers

The questionnaire to teachers from the Polytechnic Institute of Bragança was applied in the second semester of the academic year 2019/2020, as a way to understand their opinions about the teaching/learning methodology implemented. The questionnaire available in *Appendix C* was applied to the Informatics Engineering teachers who were willing to answer it after having experience with each activity guide, from the methodology created in this study, applied to their students. In other words, whenever the teacher applied an activity guide to his/her students, the questionnaire should be answered.

The results of this questionnaire can be seen, according to each question in the *Appendix C*, in Tables 6.9 and 6.10 below. In total, the questionnaire had 12 answers, made by Informatics Engineering teachers, according to some of the activity guides they experienced. On the first column, it is shown the question number of the questionnaire; On the second column, it is shown the questions' alternatives; On the third column, it is possible to see the frequency of the alternative chosen - the amount of times the teachers have chosen this answer; On the fourth column it is shown the percentage of the chosen alternative when compared to the other ones.

Table 6.9: Results of the questionnaire from *Appendix C* applied to IPB teachers.

Teachers from IPB in the academic year 2019/2020			
Question Number	Alternatives	Frequency	Percentage
1	Inside the Classroom	0	0%
	Outside the Classroom	10	100%
2	Strings	5	41.7%
	Pointers	3	25%
	Recursion	2	16.7%
	Structs	0	0%
3	Codeboard	2	20%
	URI Online Judge	3	30%
	C Tutor	5	50%
7.1	Totally Disagree	0	0%
	Partly Disagree	1	8.3%
	Neutral	0	0%
	Partly Agree	6	50%
	Totally Agree	5	41.7%

Table 6.10: 2nd part of the results of the questionnaire from *Appendix C* applied to IPB teachers.

Teachers from IPB in the academic year 2019/2020			
Question Number	Alternatives	Frequency	Percentage
7.2	Totally Disagree	0	0%
	Partly Disagree	0	0%
	Neutral	1	8.3%
	Partly Agree	6	50%
	Totally Agree	5	41.7%
7.3	Totally Disagree	0	0%
	Partly Disagree	1	8.3%
	Neutral	2	16.7%
	Partly Agree	3	25%
	Totally Agree	6	50%
7.4	Totally Disagree	0	0%
	Partly Disagree	0	0%
	Neutral	0	0%
	Partly Agree	10	83.3%
	Totally Agree	2	16.7%
7.5	Totally Disagree	0	0%
	Partly Disagree	0	0%
	Neutral	0	0%
	Partly Agree	8	80%
	Totally Agree	2	20%
7.6	Totally Disagree	0	0%
	Partly Disagree	0	0%
	Neutral	0	0%
	Partly Agree	4	40%
	Totally Agree	6	60%

In question 2, the alternative *Structs* was not evaluated owing to the fact that there was not enough time for the teachers to propose activity guides with this content before the completion of this work.

For questions 4 to 6.2 from the teacher questionnaire, it was possible to observe that not all students that received the activity guides submitted the exercises. Also, the number of submissions and correct submissions normally decrease in the second or third exercise, from each activity guide applied.

For question 8, the teachers proposed, as improvements on the activity guides, the addition of different levels of difficulty when displaying the exercises, so that teachers and students would better understand the type of exercise they would be dealing with. Also, it was proposed the addition of short videos explaining better each step the teachers and students should follow to complete the activity.

For question 9, the teachers proposed, as improvements on *EasyCoding* system, classifying each exercise available in the platform with different levels of difficulty. By doing this, the teachers would know more easily the exercises complexity before choosing them. They also suggested a search for different mechanisms to help motivate more the students in doing the activities proposed.

When analyzing the results regarding the questionnaire applied to IPB teachers about their experience with the methodology created, throughout the academic year 2019/2020, we could observe that, in general, the activity guides for teachers and the ones applied to students were effective and sufficient for the execution of the activities. Also, the students performance were considered satisfactory and the online platforms used showed a good response.

Moreover, the activity guides were applied, in its entirety, outside the classroom context. Which can be explained by students greater ease to use the online platforms after the classes, with more available time. Finally, the *EasyCoding* system was evaluated as satisfactory when analyzing the number of exercises available and its utilization and response, giving teachers the autonomy to create their own activity guides. The *EasyCoding* system will certainly facilitate the continuation of this methodology in the next academic

years, when an intensification of activities outside the classroom context is expected.

Results Regarding the Questionnaire Applied to CEFET-MG Students

The questionnaire for students at the Federal Technology Education Center of Minas Gerais (CEFET-MG) was applied after their experience with each of the web platforms applied, which were *URI Online Judge* and *C Tutor*. The questionnaire aimed to understand their opinions about the teaching/learning methodology implemented.

From the 11 students who submitted exercises in *URI Online Judge* platform, 9 answered the questionnaire and from the 6 students who submitted exercises in *C Tutor* platform, 6 answered the questionnaire about this platform. The number of students is reduced because the experiment was made only by students who, despite being without classes, volunteer to test the methodology created.

It is important to emphasize that, due to the COVID-19 pandemic, at the beginning of the year 2020, the questionnaire for the students was changed in order to better analyze this different context, as the methodology was not implemented by a teacher, but by the author of this study. Moreover, some questions regarding the online platforms had changed as the students didn't experience all the platforms mentioned in this study.

The questionnaire available in *Appendix B* was applied to students from Computer Engineering course, from CEFET-MG, in the second semester of academic year 2019/2020. The results of this questionnaire can be seen in Tables 6.11 and 6.12 below. On the first column, it is shown the question number of the questionnaire; On the second column, it is shown the questions' alternatives; On the third column, its possible to see the frequency of the alternative chosen - the amount of students who chose this answer - by each platform experienced; On the fourth column it is shown the percentage of the students who chose this alternative, by each platform experienced.

Table 6.11: Results of the questionnaire from *Appendix B* applied to CEFET-MG students.

Students from CEFET-MG in second semester of academic year 2019/2020					
Question Number	Alternatives	Frequency		Percentage	
		URI	C Tutor	URI	C Tutor
1	Completed all the activities	6	6	66.7%	100%
	Completed most of the activities	1	0	11.1%	0%
	Completed few activities	2	0	22.2%	0%
2.1	Totally Disagree	1	0	11.1%	0%
	Partly Disagree	1	0	11.1%	0%
	Neutral	1	0	11.1%	0%
	Partly Agree	2	1	22.2%	16.7%
	Totally Agree	4	5	44.4%	83.3%
2.2	Totally Disagree	0	0	0%	0%
	Partly Disagree	0	0	0%	0%
	Neutral	0	0	0%	0%
	Partly Agree	1	0	11.1%	0%
	Totally Agree	8	6	88.9%	100%
2.3	Totally Disagree	0	0	0%	0%
	Partly Disagree	1	0	11.1%	0%
	Neutral	1	0	11.1%	0%
	Partly Agree	1	1	11.1%	16.7%
	Totally Agree	6	5	66.7%	83.3%
2.4	Totally Disagree	2	5	22.2%	83.3%
	Partly Disagree	4	1	44.4%	16.7%
	Neutral	1	0	11.1%	0%
	Partly Agree	1	0	11.1%	0%
	Totally Agree	1	0	11.1%	0%

Table 6.12: 2nd part of the results of the questionnaire from *Appendix B* applied to CEFET-MG students.

Students from CEFET-MG in second semester of academic year 2019/2020					
Question Number	Alternatives	Frequency		Percentage	
		URI	C Tutor	URI	C Tutor
2.5	Totally Disagree	0	0	0%	0%
	Partly Disagree	0	0	0%	0%
	Neutral	0	0	0%	0%
	Partly Agree	0	0	0%	0%
	Totally Agree	9	6	100%	100%
2.6	Totally Disagree	0	0	0%	0%
	Partly Disagree	0	0	0%	0%
	Neutral	0	0	0%	0%
	Partly Agree	2	1	22.2%	16.7%
	Totally Agree	7	5	77.8%	83.3%

For question 3, all students stated that the platforms used should be indeed considered as allies of programming beginners, explaining that they are very important extensions to put the study and learning of programming into practice in an effective way. It was also emphasized that this is a way of reviewing the content studied and implement what was learned. When answering about their experience with *C Tutor* platform, it was described that it is a good tool for beginners, as the student can see step by step the execution of his/her code, and better understand the concepts studied in theory. When answering about their experience with *URI Online Judge* platform, the students analyzed that this tool ranges from simple exercises to more complex programming exercises, which motivates students to practice more.

For question 4, the students proposed as improvements in the activity guides to increase the number of proposed exercises and further explanations about the acceptance

requirements of the submitted codes, in the case of exercises on the *URI Online Judge* platform. It was also proposed as improvement in the *C Tutor* activity guide, to better explain how the mechanics of executing line by line of code in the system works.

For question 5, the students described as an idea to motivate the ones who did not collaborate with this project, to encourage competition between the students, in a healthy way and fostering competitiveness and thus instigate students to collaborate, by giving extra grades in the class.

When analyzing the results regarding the questionnaire applied to CEFET-MG students about their experience with the methodology created, in the second semester of academic year 2019/2020, it can be observed that the online platforms used indeed motivated students to practice more programming exercises outside the classroom context.

Moreover, the CEFET-MG students did not experience many difficulties in carrying out the proposed activities. This result can be explained as the students did not have the possibility of doing many complex exercises, due to the COVID-19 pandemic that caused an academic break at the beginning of the year 2020. In general, the activity guides proposed were considered sufficient for the execution of the activities and the online platforms showed a good response. Moreover, the exercises proposed to CEFET-MG students served as an additional distance learning method.

Chapter 7

Conclusions

7.1 Final Considerations

From the analysis of the results obtained above, it was possible to identify some characteristics regarding the application of the activity guides with programming exercises to be solved, inside the online platforms classified, as a teaching methodology. It was observed that students were more willing to adhere to this methodology at the beginning of each academic semester, when the exercises were simpler to solve. The causes of this may have been the decrease in available time as the academic semester ended, in addition to a possible less interest in solving more complex exercises.

The possibility that may explain the not-so-high number of students engaged with the proposal of this study is the fact that the activity guides proposed were completely voluntary, with no additional score for students in the enrolled subject. Furthermore, the activities were proposed by sending them to students' academic e-mail, which may have contributed to a smaller number of students participating in the study, since not every student reads the academic e-mail.

In general, the students' opinion about the methodology created was very positive and they agreed that the use of the online platforms classified can indeed motivate them to practice more programming outside the classroom context. The teachers' opinion

showed that student performance was satisfactory when applying this methodology and the *EasyCoding* system offered a good response when generating the activity guides.

Therefore, when applying this methodology in programming classes, it can contribute to a greater engagement by students and, consequently, help decrease failure rates in the first semesters of technology courses. For this to happen, this methodology should be intensified in the next semesters and the activities performed should be considered for student evaluation. The new pedagogical methodologies previewed for the very next future include a decrease of the number of presential classes and an increase of asynchronous activities. This will require more work from teachers when it comes to preparing new teaching methodologies. In this sense, *EasyCoding* system will be a great help in providing innovative activities for programming students.

7.2 Future Perspectives

As perspectives for future work, some proposals stand out:

- Encourage the systematic adoption of the methodology created in this work to assist in programming teaching.
- Encourage the use of the *EasyCoding* system, by the teachers, for the generation of activity guides to be used inside and outside the classroom, as a complement to study planning.
- Make this methodology part of the evaluation of students enrolled in programming subjects, to be done using student submissions on the classified platforms.
- Improve the *EasyCoding* system, making it more useful to teachers by enhancing the functionalities and developing new features.

The *EasyCoding* system currently provides important features to assist teachers in generating new activities according to their needs. The platform was developed in a

modular and generic way. Therefore, the system is prepared to receive upgrades and new features in the future.

The activity guides generated have proven to be important in assisting the teacher and students in the use of the online platforms. Moreover, the application of the activity guides showed satisfactory results regarding students' motivation to practice more programming beyond the classroom context.

Bibliography

- [1] R. L. L. SILVA FILHO, “A evasão no ensino superior brasileiro: Novos dados”, Available in: http://institutolobo.org.br/imagens/pdf/artigos/art_088.pdf, Acesso em: 04 de março de 2020, vol. 16, p. 2017, 2018.
- [2] L. S. Almeida and R. Vasconcelos, “Ensino superior em portugal: Décadas de profundas exigências e transformações”, *Innovación Educativa*, vol. 18, pp. 23–34, 2008.
- [3] W. Nagai, C. Izeki, and R. Dias, “Experiência no uso de ferramentas online gamificadas na introdução à programação de computadores”, in *Anais do Workshop de Informática na Escola*, vol. 22, 2016, p. 301.
- [4] M. A. Borges, “Avaliação de uma metodologia alternativa para a aprendizagem de programação”, in *VIII Workshop de Educação em Computação–WEI*, 2000, p. 15.
- [5] C. E. Rapkiewicz, G. Falkembach, L. Seixas, N. dos Santos Rosa, V. V. da Cunha, and M. Klemann, “Estratégias pedagógicas no ensino de algoritmos e programação associadas ao uso de jogos educacionais”, *RENOTE-Revista Novas Tecnologias na Educação*, vol. 4, no. 2, 2006.
- [6] K. M. Kapp, *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.
- [7] G. Zichermann and C. Cunningham, *Gamification by design: Implementing game mechanics in web and mobile apps*. " O'Reilly Media, Inc.", 2011.
- [8] Codecademy, Available in: <https://www.codecademy.com/>, Accessed in: 23 de março de 2020.

- [9] CodeCombat, *Coding games to learn python and javascript*, <https://codecombat.com/>, Accessed: 2020-02-12.
- [10] J. H. Sharp, “Using codecademy interactive lessons as an instructional supplement in a python programming course”, *Information Systems Education Journal*, vol. 17, no. 3, p. 20, 2019.
- [11] J. Swacha and P. Baszuro, “Gamification-based e-learning platform for computer programming education”, in *X world conference on computers in education*, 2013, pp. 122–130.
- [12] Scratch, *Imagine, program, share*, <https://scratch.mit.edu/>, Accessed: 2020-03-24.
- [13] Code.org, *Learn computer science. change the world*, <https://code.org/>, Accessed: 2020-03-24.
- [14] L. Severgnini and E. Soares, “O serious game codecombat e o professor como mediadores da aprendizagem do pensamento computacional”, in *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Infomática na Educação-SBIE)*, vol. 30, 2019, p. 684.
- [15] F. Desmoulins and N. Antoniazzi, Available in: <https://www.codingame.com/about/team>, Accessed in: 2020-01-20.
- [16] Coderbyte, Available in: <https://coderbyte.com/>, Accessed in: 2020-01-20.
- [17] J. O. Chaves, A. Castro, R. Lima, M. V. Lima, and K. Ferreira, “Mojo: Uma ferramenta para auxiliar o professor em disciplinas de programação”, in *Congresso Brasileiro de Ensino Superior a Distância, Belém, PA*, 2013.
- [18] A. Kurnia, A. Lim, and B. Cheang, “Online judge”, *Computers & Education*, vol. 36, no. 4, pp. 299–315, 2001.
- [19] S. R. Labs, *Spoj brasil (spoj)*, <https://br.spoj.com/>, Accessed: 2020-03-29.
- [20] U. O. Judge, *Online judge*, <https://onlinejudge.org/>, Accessed: 2020-03-29.

- [21] J. O. M. Chaves, A. F. Castro, R. W. Lima, M. V. A. Lima, and K. H. Ferreira, “Integrando moodle e juizes online no apoio a atividades de programação”, in *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)*, vol. 24, 2013, p. 244.
- [22] S. Wasik, M. Antczak, J. Badura, A. Laskowski, and T. Sternal, “A survey on online judge systems and their applications”, *ACM Computing Surveys (CSUR)*, vol. 51, no. 1, pp. 1–34, 2018.
- [23] I. Foundation, *The icpc international collegiate programming contest*, <https://icpc.baylor.edu/>, Accessed: 2020-04-12.
- [24] M. Selivon, J. L. Bez, N. A. Tonin, and R. Erechim, “Uri online judge academic: Integração e consolidação da ferramenta no processo de ensino/aprendizagem”, *Anais do 23º Workshop sobre Educação em Computação (WEI 2015)*, n. CSBC, 2015.
- [25] S. Combéfis, G. Beresnevičius, and V. Dagienė, “Learning programming through games and contests: Overview, characterisation and discussion”, *Olympiads in Informatics*, vol. 10, no. 1, pp. 39–60, 2016.
- [26] F. Prados, I. Boada, J. Soler, and J. Poch, “Automatic generation and correction of technical exercises”, in *International conference on engineering and computer education: Icece*, vol. 5, 2005.
- [27] C. Estler and M. Nordio, Disponível em: <https://codeboard.io/about>, Accessed in: 2020-01-17.
- [28] N. A. Tonin and J. L. Bez, “Uri online judge: A new classroom tool for interactive learning”, in *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS)*, 2012, p. 1.
- [29] S. Schleimer, D. S. Wilkerson, and A. Aiken, “Winnowing: Local algorithms for document fingerprinting”, in *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, 2003, pp. 76–85.

- [30] G. Philip J., *Online Python Tutor: Embeddable web-based program visualization for CS education*, Available in: <http://doi.acm.org/10.1145/2445196.2445368>, Accessed: 2020-01-17, 2013.
- [31] B. Kjell, Available in: <https://chortle.ccsu.edu/CPuzzles/CPuzzlesMain.html>, Accessed in: 2020-01-20.
- [32] M. Almeida. (2020). Metodologia de suporte à aprendizagem de programação, [Online]. Available: <https://mestrado-marcelaviana.webnode.com/> (visited on 05/19/2020).
- [33] J. Moreno, “Digital competition game to improve programming skills”, *Journal of Educational Technology & Society*, vol. 15, no. 3, pp. 288–297, 2012.
- [34] M. Almeida. (2020). Easycoding - metodologia de suporte à aprendizagem de programação, [Online]. Available: https://marciviana.github.io/projeto_mestrado.html (visited on 05/05/2020).
- [35] I. GitHub. (2020). About github pages, [Online]. Available: <https://help.github.com/en/github/working-with-github-pages/about-github-pages#guidelines-for-using-github-pages> (visited on 03/02/2020).
- [36] I. GitHub. (2020). Github.com help documentation, [Online]. Available: <https://help.github.com/en/github> (visited on 03/28/2020).
- [37] L. Damas, *Linguagem C. 10th Edition*. LTC Editora, 2007.
- [38] M. S. Silva, *Criando sites com HTML: sites de alta qualidade com HTML e CSS*. Novatec Editora, 2008.
- [39] M. S. Silva, *JavaScript-Guia do Programador: Guia completo das funcionalidades de linguagem JavaScript*. Novatec Editora, 2010.
- [40] MrRio, *Jspdf: A library to generate pdfs in javascript*. <https://github.com/MrRio/jsPDF>, Accessed: 2020-03-29.
- [41] T. J. Foundation, *Jquery: Write less, do more*, <https://jquery.com/>, Accessed: 2020-03-29.

- [42] G. H. Libraries, *Hosted libraries / google developers*, <https://developers.google.com/speed/libraries>, Accessed: 2020-03-30.
- [43] Microsoft. (2020). Getting started with vba in office, [Online]. Available: <https://docs.microsoft.com/en-us/office/vba/library-reference/concepts/getting-started-with-vba-in-office> (visited on 05/19/2020).
- [44] M. V. P. Almeida, M. J. V. Pereira, L. M. Alves, and G. A. R. Barbosa, “Easy-coding: Methodology to support programming learning”, *International Computer Programming Education Conference - ICPEC 2020 (Approved and pending publication)*, 2020.
- [45] C. F. Leitão and R. O. Prates, “A aplicação de métodos qualitativos em computação”, *Jornadas de Atualização em Informática*, vol. 2017, pp. 43–90, 2017.
- [46] I. Seidman, *Interviewing as qualitative research: A guide for researchers in education and the social sciences*. Teachers college press, 2006.
- [47] J. Lazar, J. H. Feng, and H. Hochheiser, *Research methods in human-computer interaction*. Morgan Kaufmann, 2017.

Appendix A

Questionnaire Applied to Students from IPB

Name: _____

Educational institution enrolled: _____

Course you are enrolled in: _____

1. About the proposed exercises in the activity guides:

- You have managed to complete all activities
- You completed most of the activities
- You completed few activities

2. Knowing that the numbers mean:

1 = Totally Disagree 2 = Partly Disagree 3 = Nor Disagree Nor Agree (Neutral)
4 = Partly Agree 5 = Totally Agree

1. The platforms used in the activity guides proved to be easy to use and quick to respond.	1	2	3	4	5
2. You consider that you understood what was proposed in each activity guide.	1	2	3	4	5
3. The activity guide proposed to the student proved to be sufficient for the correct execution of the activity.	1	2	3	4	5
4. You experienced difficulties in carrying out the activities in terms of programming.	1	2	3	4	5
5. The exercise proposals of the activity guides are related to the content studied.	1	2	3	4	5
6. In general, the proposed activity guides motivated you to practice more programming exercises, in addition to the classroom.	1	2	3	4	5

3. Which of the platforms used proved to be the most useful and should be considered as the best ally for programming beginners?

- Codeboard
- URI Online Judge
- C Tutor
- Coderbyte
- CodinGame
- None of them

4. Which platforms used can lead students who don't know how to program to try new things?
 Codeboard
 URI Online Judge
 C Tutor
 Coderbyte
 CodinGame
 None of them
5. Which platforms used should be considered the best to discover errors and fix them?
 Codeboard
 URI Online Judge
 C Tutor
 Coderbyte
 CodinGame
 None of them
6. What form of application of activity guidelines do you find most useful for learning programming?
 Inside the classroom context
 Outside the classroom context
7. How could we motivate students who have not collaborated with this project?

8. Any suggestions for improving the proposed activity guides?

Appendix B

Questionnaire Applied to Students from CEFET-MG



Master's in Information Systems
Questionnaire applied to students from CEFET-MG

Name: _____

Educational institution enrolled: _____

Course you are enrolled in: _____

Platform used:

- URI Online Judge
- C Tutor
- Codeboard

1. About the proposed exercises in the activity guides:

- You have managed to complete all activities
- You completed most of the activities
- You completed few activities

2. Knowing that the numbers mean:

1 = Totally Disagree 2 = Partly Disagree 3 = Nor Disagree Nor Agree (Neutral)
4 = Partly Agree 5 = Totally Agree

1. The platform used in the activity guides proved to be easy to use and quick to respond.	1	2	3	4	5
2. You consider that you understood what was proposed in each activity guide.	1	2	3	4	5
3. The activity guide proposed to the student proved to be sufficient for the correct execution of the activity.	1	2	3	4	5
4. You experienced difficulties in carrying out the activities in terms of programming.	1	2	3	4	5
5. The exercise proposals of the activity guides are related to the content studied.	1	2	3	4	5
6. In general, the proposed activity guides motivated you to practice more programming exercises, in addition to the classroom.	1	2	3	4	5

3. Do you consider that the platform used should be considered as an ally of programming beginners? Justify.

4. How could we motivate students who have not collaborated with this project?

5. Any suggestions for improving the proposed activity guides?

Appendix C

Questionnaire Applied to Teachers from IPB

Name: _____

1. Context of application of the activity:
 - Inside the classroom
 - Outside the classroom
2. Subject chosen in the activity guide:
 - Strings
 - Pointers
 - Recursion
 - Structs
3. Selected platform(s):
 - URI Online Judge
 - Codeboard
 - C Tutor
4. Number of students that received the activity: _____
5. Number of students that submitted the exercise 1: _____
- 5.1 Number of students that submitted the exercise 2 (if it exists): _____
- 5.2 Number of students that submitted the exercise 3 (if it exists): _____
6. Number of students that submitted correctly the exercise 1: _____
- 6.1 Number of students that submitted correctly the exercise 2 (if it exists): _____
- 6.2 Number of students that submitted correctly the exercise 3 (if it exists): _____
7. Knowing that the numbers mean:

1 = Totally Disagree 2 = Partly Disagree 3 = Nor Disagree Nor Agree (Neutral)
4 = Partly Agree 5 = Totally Agree

1. The activity guide proposed to the teacher proved to be sufficient for the correct execution of the activity with the students.	1	2	3	4	5
2. The activity guide proposed to the student proved to be sufficient for the understanding and execution of the activity.	1	2	3	4	5
3. The students' performance was satisfactory.	1	2	3	4	5

4. The platform(s) used in the activity are easy to use and provide quick response.	1	2	3	4	5
5. The EasyCoding system has a satisfactory number of exercises available.	1	2	3	4	5
6. The EasyCoding system is easy to use and provides quick response.	1	2	3	4	5

8. What does the teacher suggest to improve the activity guide proposed to the teacher? And in the activity guide proposed to the student?

9. What does the teacher suggest for improvement in the EasyCoding system?
