2022 31st Annual Conference of the European Association for Education in Electrical and Information Engineering (EAEEIE) | 978-1-6654-8442-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/EAEEIE54893.2022.9820594

# A Web-based Platform and a Methodology to Teach Programming Languages in Electrical Engineering Courses

Inácio Fonseca[1]
[1]*Polytechnic Institute of Coimbra, Coimbra Institute of Engineering*
3030-199 Coimbra, Portugal
inacio@isec.pt

Nuno Cid Martins[1]
[1]*Polytechnic Institute of Coimbra, Coimbra Institute of Engineering*
3030-199 Coimbra, Portugal
ncmartin@isec.pt

Fernando Lopes[1,2]
[2]*Instituto de Telecomunicações - Coimbra*
Pinhal de Marrocos, Pólo II DEEC
3030-290 Coimbra, Portugal
flopes@isec.pt

***Abstract*—Teaching programming languages is a fundamental component of electrical engineering education. However, it poses very serious challenges, mostly associated with very different backgrounds and motivation among students, as well as with its strong dependence on the graphical presentation of topics and on the student-teacher interaction.**

**In this paper we present the experience in the development and operation of a web-based platform and a methodology to support student interaction and assistance in classroom teaching of programming languages in electrical engineering courses. The developed platform allows the installation of highlighting, linting and programming languages' recognition tools. It uses software modules adapted from Codiad and other tools.**

**Two example teaching modules are detailed. The first module refers to teaching programming Web Interfaces in an "Applied Information Systems" course, integrating an Electrical Engineering Master degree. The second module refers to the use of Matlab for statistical analysis in a "Maintenance and Quality Control" course, integrating an Electrical Engineering Bachelor degree. An experimental module for coding in Ladder via a graphical web interface for teaching PLC programming, is also part of the current development status.**

**In the near future, it is intended to expand the platform for teaching programming for IoT, with support for the Arduino, ESP32 and STM32 microcontrollers. Support for other specific electrical engineering topics such as signal processing and DSP hardware can also be integrated in the platform.**

***Index Terms*—Programming languages, microcontrollers, teaching platform, electrical engineering**

## I. Introduction

In electrical engineering education, programming is used in a wide range of areas and applications. It is fundamental for simulation, device firmware and system software. Applications can be found in embedded systems, automation and industrial communications, sensors and instrumentation, signal processing and telecommunications, IoT devices and networks, and power and energy components and systems. However, teaching programming languages is a complex and challenging task, often posing several difficulties. These difficulties exist at both the methodology and technical levels.

On the one hand, there are often strong discrepancies between students' initial knowledge and skills. In a classroom environment, this usually results in difficulties that arise from simultaneously having students who are highly motivated and quickly absorb knowledge and carry out practical experiences, and therefore want more contents, and students who have difficulty understanding syntax and quickly become discouraged if they are not supported at this early stage of learning. This happens with all languages, as for example with C, Matlab, Python and Web Programming, whether for embedded systems, desktop, smartphone or web applications. On the other hand, the technological conditions in the classroom, such as graphical presentation and IT equipment, may be limited and not flexible nor adequate to specific programming environments.

Currently, there are several tools that allow online editing, compilation and debugging of software programs, such as Octave Online [1] and Matlab Online [2], Codelabs, Visual Studio Code for Web [3], among others. This illustrates the interest in these tools and the versatility they offer. However, these single-user or collaborative developing tools are generally not focused on in-person or online interactive classroom teaching. Additionally, using non-owned tools, leaves institutions vulnerable to policy changes on free and paid software tools.

In this paper we present the experience in the development and operation of a web-based platform and a methodology to support student interaction and assistance in classroom teaching of programming languages in electrical engineering courses [4]. The developed platform allows the installation of highlighting, linting and programming languages' recognition tools. It uses modules adapted from Codiad [5] and other tools. The platform allows, for example, for the teacher to introduce exercises and associate code to each exercise. The teacher can further introduce an initial skeleton for the solution and make the solution available in real time during the class or deferred to another convenient time. This share is automatically visible to all students enrolled in the class. The students can, at any time, decide to use one of the available options and start solving the proposed exercise from that point on, with optional help that is available at each stage.

978-1-6654-8442-8/22/$31.00 ©2022 IEEE

## II. Technology Context

Educational tools to support teaching and learning of technology topics can take several forms, including educational web-based platforms, educational devices and game-based educational applications. In the following sub-sections, these three types are shortly presented and exemplified.

### A. Educational web-based platforms

Web-based learning platforms deliver a helpful environment for students, to independently solve science and technology problems. An example is LoopAcademic [6], a web environment proposed for learning and teaching basic programming. This environment, developed in CSS, HTML5, JavaScript and other technologies, was designed based on the needs of students who attend introductory programming subjects. It uses a methodology that starts with selecting the web development tools and finishes with the coding of the front-end [6].

Another example, in the perspective of independent learning through an online platform, is proposed in [7]. This platform was constructed for rapid prototyping and experimenting with software agents and to help programmers creating web-based applications in a simple and controlled way. The implemented data-flow and visual programming environment allows to discover, gather and handle information available on the web. It also permits an easy incorporation of software modules, including modules that can be characterized by exhibiting an autonomous intelligent behaviour.

Despite the success of the example tools just presented, educational web-based platforms need further exploration, leading to a more structured and supportive strategy, in order to significantly improve learning results [8]. Though educators and platforms can contribute to support the learning process, the literature shows few cases in which educators can successfully interact with the students in such environments [8]. The work presented in [8], provides empirical proof on the success of applying a structured and supportive educational strategy in autonomous web-based platforms, and settles that learners have better performance in the class with this kind of approach. Their results also emphasize the critical role of educators in a web-based learning environment. An application case of the mentioned strategy can be seen in the following example. The authors in [9] describe the implementation of an IoT remote laboratory in teaching microprocessor technology. A monitoring device allows the instructor to supervise the progress of the students' microcontroller tasks, by sending collected data to the teacher's server. The IoT remote laboratory implementation shows that students' tasks were improved through the interactive classes.

The learning process can be facilitated when the basic programming concepts are in the student's native language. PanScript [10], an open-source web-based platform, follows this objective by allowing to collect and simplify (by translating tokens, error messages, and code samples) many text-based programming languages, in a beginner and friendly web-based programming platform.

### B. Educational devices

Educational technology aims to improve specific components of the teaching and learning process. It includes conventional devices, such as, for instance, computers and overhead projectors, as well as newer devices, such as, for example, smart interactive multi-touch surfaces, Augmented Reality (AR) devices, and task specific devices. Two demonstrative examples of educational devices are presented next.

An original didactic device, called the Centrifugal Ring Positioner (CRP), is presented in [11]. This tool helps to illustrate various concepts on closed-loop control within multidisciplinary contexts. Allowing the visualization of control principles, missing in some educational setups, and offering the possibility of solving control problems with different levels of complexity, the use of this device increases the students' interest and curiosity.

AR for education is an emergent area of research and application, typically used to aid in lectures, exercise classes and laboratories [12]. The systematic review regarding AR and engineering education in [12], shows that the academic performance is increased in most of the revised studies. However, it is also concluded that "there is plenty of room for the future use of AR in engineering studies, but each engineering area must identify adequate educational purposes". An example of AR for education is presented in [13]. The application allows an interactive study of the space-state equations of a series-connected Resistance–Inductor–Capacitor circuit and its assembly in a virtual proto-board. It also permits the study of the behaviour of Bidirectional Direct Current converters when energy is exchanged between two systems. This tool explores the valuable feedback from students and, compared to traditional teaching, allows for better cognitive performance.

### C. Game-based educational applications

Learning by playing educational games or solving amusing problems has a big impact on students' performance. That is why gamification in education has become a focus of attention in recent years. According to [14], "gamification is the practice of using game design elements, game mechanics and game thinking in non-game activities to motivate participants". In a systematic literature review in [15], a set of 53 examples of educational game-based applications are identified, with different approaches to evaluate their success. Three representative cases are presented next.

In [16], the authors propose a game-based learning platform to aid students learning cybersecurity, at their own pace, and give them the chance to improve hacking skills with ethics, in a safe environment. The platform was designed based on an adaptation of the John Keller instructional and motivation model (the ARCS) [17]. It includes a virtual laboratory with the necessary tools for practice and a web portal where all challenges and learning materials are hosted.

Code Factory, in [18], aims to be a friendly learning environment, where students, without previous programming knowledge, can explore basic programming concepts in a clear, motivating way and according to their cognitive level.

It consists of a platform-style game, where students can control a virtual robot, overcoming several obstacles in a test of reflexes and logical reasoning, having their first contact with the logic of programming through a motivating and constructivist approach.

With the Covid-19 pandemic, E-learning tools became more relevant. A mixture of web-based platforms and game-based applications, a project under the name of "Framework for Gamified Programming Education (FGPE)", is presented in [19]. This open-source and general-purpose tool (i.e., not integrated with a particular course on a specific programming language), with a significant assortment of gamification components, can effectively support the organization of online gamified programming courses.

## III. Tendencies and Similar Tools

Today, teaching programming is moving to online either through dedicated web platforms, or through courses on YouTube, Udemy, etc. To learn to program is an intensive task of training and re-training, where the intensity depends heavily on the specific language and on the involved paradigm. Some global tendencies and a set of tools similar to our proposal are presented in this section.

### A. Tendencies

Data Flow Programming (DFP) allows to conceive a program by looking graphically the data flowing through a series of transformations and operations. Successful examples are LabVIEW from National Instruments, NodeRed, *n8n.io*, Crosser, among many others [20].

To introduce programming to young students (kids), strategies like Block Programming or Lego Programming have been introduced. Examples are Scratch (from MIT), PictoBlox, Snap, Hopscotch, Tyner, Beetle Blocks, among others [21]. PictoBlox is based on Scratch, connects to any hardware and introduces AI programming for kids (see STEM Project Hub for specific examples). Snap allows to use block and text programming together. Hopscotch is designed for kids from 4 to 7. Tyner is a game-design platform created with the aim of teaching kids how to code. Beetle Blocks is a visual code for 3D design and fabrication - it prints to a 3D printer.

In text-based programming, some languages allow web-based tools [22], while with others, it can be complex if using hardware without TCP/IP connectivity. Multiple tools specific for learning, like W3Schools [23], have linting and correction support as standard in the development environment. Language specific tools are common. *online-python.com* and Anaconda tools and Jupyter Notebook *jupyter.org* are good examples for Python. Textual languages usually have an online rapid development tool to accelerate learning. Suggested AI-supported software patterns start to be seen in some text-based development environments [24]. In Education and E-Learning, AI is expected to grow, not only on evaluating student's responses, but also on deciding a personal and effective learning path for each student [25]. In the AI age, students need to know these five technologies: Artificial Intelligence, Machine Learning, Deep Learning, Natural Language Processing and Computer Vision - normally, but not only, Python-based [26]. CodeSandbox is an online code editor and prototyping tool that makes learning, creating and sharing Web Apps faster [22]. CodePen [22] is also another good example among many others.

### B. Similar tools

Big companies are now focused on web-based educational/professional tools, with excellent educational products. An example is Tinkercad, with an online Arduino IDE for programming, running and debugging software, hardware and electrical circuits in a virtual simulator.

In this digital era, many companies and institutions are developing platforms and tools for education. However, some of these platforms and tools may be free in a first step approach, but with a strategy of turning them paid in the future. Thus, it can be expected that in the future educational institutions will have to pay for the access to these products, in a per-student fee basis.

Similar ideas to what we want conceptually accomplish, can be found in *learnprogramming.online*, a paid tool, where students have access to organized programming lessons. The Web Interface (see *example 01*) can be seen in Fig. 1.

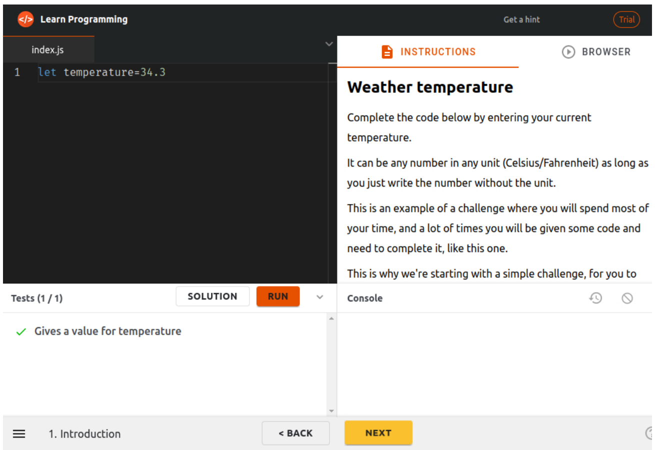


Fig. 1: *learnprogramming*'s Web Interface.

In Section IV we will present our experimental platform, configured to teach PHP and Matlab using a web-based approach. It is fully integrated in the teaching of two electrical engineering courses. The current development, with the successful addition of debugging capabilities and support for other languages is also presented in Section V.

## IV. Web Platform for Teaching and Learning

The Web Platform is split between a Web Code Editor and the supporting content organized as classes, numbered from 1 to 15. Two courses are now using this method of teaching support, *Applied Information Systems* - AIS and *Maintenance and Quality Control* - MQC. In AIS students learn to design a database for electrical engineering information.

In terms of language, they learn PHP, because it is very similar to C programming and given that C is part of the students' curriculum. In PHP programming, students learn to work with forms and to create an API for transferring data between embedded systems and a server. In MQC, students learn maintenance theory, control quality tools, control charts, quality control based on industrial computer vision software and they use maintenance software, and reliability calculus. Matlab is used as a supporting tool for calculations.

### *A. AIS - Applied Information Systems*

Two main problems in teaching programming languages come from: a) thinking about the organization of the code itself, for students with technical know-how of the language, and b), for students still in the previous step, where the technical details are unknown or not yet learned, typically they need basic exercises. In this second situation, b), it is important for the teacher to be able to provide, as initial help, a skeleton of the program for the exercise and, at a later stage, the full code solution. Fig. 2 to Fig. 4 present some of the example exercises for students in b), where basic syntax exercises help to consolidate knowledge. In the Teacher View mode, the teacher can write the exercise, in Portuguese or in English (for Erasmus students), test a solution and record it in private or public mode, as well as create the initial skeleton and insert links to pages with more help information.

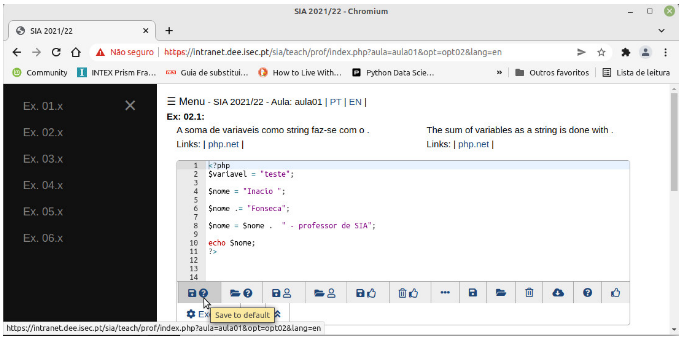


Fig. 2: Example exercise for students with language syntax difficulties - Teacher View. Syntax to add strings.

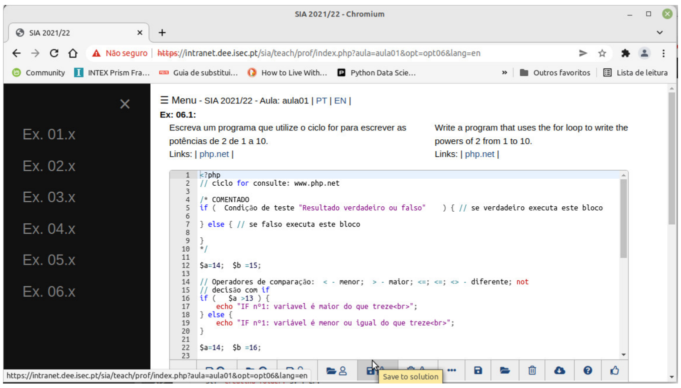


Fig. 3: Example exercise for students with language syntax difficulties - Teacher View. Syntax for loops.

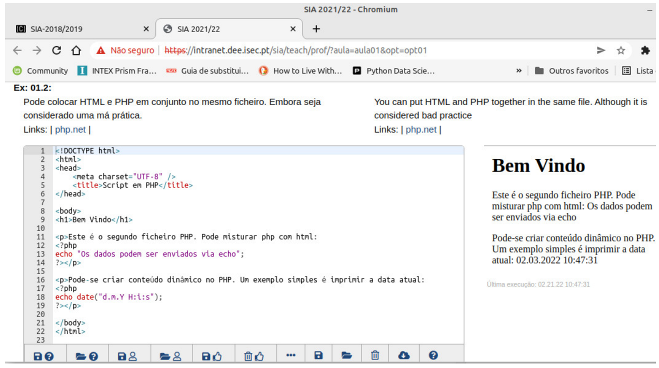


Fig. 4: Example exercise for students with language syntax difficulties - Teacher View. Syntax for HTML with PHP inside.

All students (case a) and case b)) can use the basis interface (Web Editor - see Fig. 6). In this interface, in Student View, they can edit a *.conf* file and configure PHP software packages to be installed by the package manager, specify the folder structure, setup ad-hoc download locations for other libraries, for example *javascript*, and identify the application authors. This file can be provided by the teacher in an exercise, thus ensuring that all students share the same setup. With several configuration files, the teacher can exemplify different code organization structures (see Fig. 5) and thus allow students to understand the positive aspects of each teacher's proposal. The code in Listing 1 shows an example of such a configuration file. The teacher also provides two complementary PHP programs, "setup.php" and "setup_clean.php". The first reads the configuration file and performs the indicated actions: creates the folder structure and downloads software. The second deletes downloaded software and empty folders.

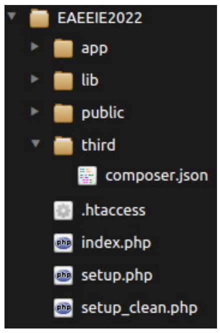


Fig. 5: Program structure in an example exercise.

### *B. MQC - Maintenance and Quality Control*

In the case of MQC, 9 lessons were created to teach the 7 quality control tools, control charts, and reliability calculations using Matlab via a webpage. The interface is the same as described in Section IV-A. However, it allows students to write and run programs in Matlab. Fig. 6 and Fig. 7 present the Student View for the set of lessons and an example exercise, respectively. The benefit of this process leads the teaching to be focused on the practical experimentation of the student, without the need to install any computer tool.

Listing 1: Configuration file *composer.json*

```
{"name": "inacio/trabalho",
"description": "Sistema via web",
"version": "0.0.1",
"time": "2021-11-05 21:00:00",
"type": "project",
"license": "MIT",
"authors": [{
   "name": "Inácio Fonseca",
   "email": "inacio@isec.pt",
   "homepage": "http://www.isec.pt",
   "role": "Developer"
}],
"support": {"email": "inacio@isec.pt"},
"app_download": [
 {
   "target": "public/fotos",
   "files": {
        "logo-isec.png":
"www.isec.pt/assets_isec/logo-isec.png"
   },
   "allow.htaccess": "BaseDir2All"
 },{
   "target": "third/jscss",
   "files": {
        "bulma.min.css":
"cdn.jsdelivr.net/npm/css/bulma.css",
        "fontawesome.js":
"use.fontawesome.com/v5.15.4/js/all.js",
        "fontawesome.css":
"use.fontawesome.com/v5.15.4/css/all.css"
   },
   "allow.htaccess": "Dir2All"
 },{
   "target": "third/webfonts",
   "files": {
        "fa-solid-900.woff2":
 "use.fontawesome.com/fa-solid-900.woff2"
   },
   "allow.htaccess": "Dir2All"
 }],
"app_folder_structure": [
        "public/fotos",
        "lib/templates/css",
        "lib/templates/js",
        "third/jscss",
        "third/webfonts"
],
"require": {
        "php": "^8.0",
        "symfony/process": "^4.0",
        "twig/twig": "v3.3.3",
        "smarty/smarty": "v3.1.40",
        "doctrine/orm": "2.10.2",
        "nette/forms": "v3.1.5",
        "phpmailer/phpmailer": "v6.5.3"
}}
```

As in case b) of section IV-A, the Student View allows reading and recording the program, as well as accessing the solution and the initial skeleton provided by the teacher.

## V. Conclusions and Future Work

The present system makes it possible to focus teaching on the student, freeing the need for external use of software and allowing the teacher to better organize lessons. The student has no need to install any tool on his own computer and has centralized access to all the contents.

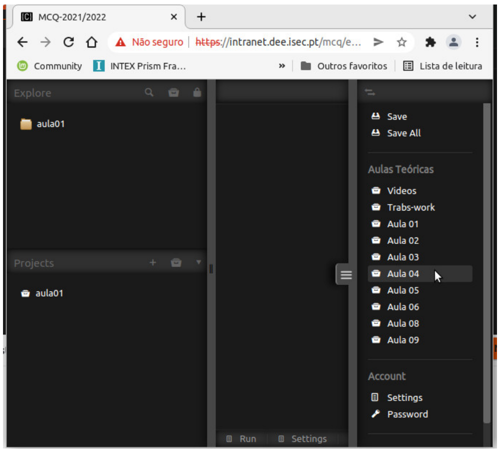


Fig. 6: MQC: Student View with 9 lessons available.

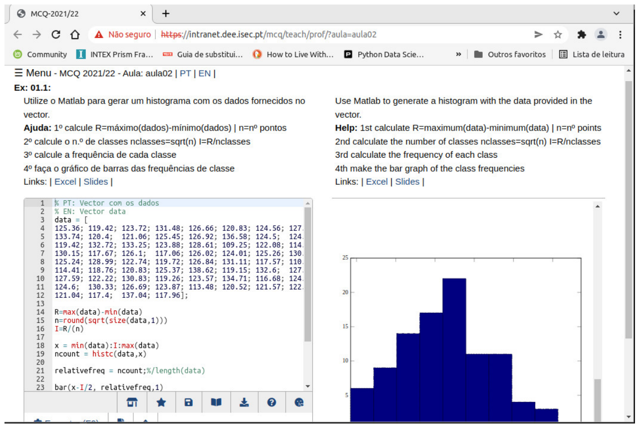


Fig. 7: MQC: Student View with histogram chart example.

In both courses, AIS and MQC, it is possible to start the courses immediately, without initial software installation and tweaking. Students who may want to install the software on their personal computers can do it outside the classroom, and thus becoming autonomous in terms of Internet connection. In the case of AIS they can further work collaboratively on practical exercises outside the classroom (see *example 2* or *example 3*). The overall receptivity from the students has been very good and we observed less dropouts and better average classifications (to be further analysed).

The benefits of an engineering school developing its own teaching and learning tools for specific applications are: a) apply its own technical knowledge; b) maintain its pedagogical autonomy; c) maintain its financial and operational independence from platforms that may change policies in the future.

Tests are being carried out to introduce debug capabilities in the Web Editor. This new functionality will use Code-OSS (Visual Studio Code). It is now possible to debug programs in PHP online (Fig. 8, Fig. 9). Adding debug in C and the loading of the binaries on embedded systems is also being implemented in this new version of the editor. Figure 10 shows another experimental module to be added, consisting on a specific editor for Ladder programming for PLC devices.

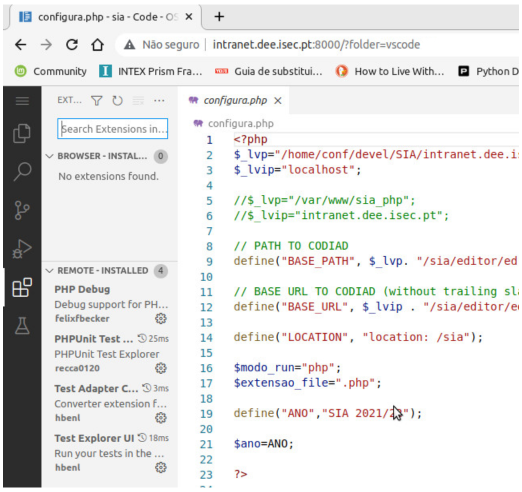


Fig. 8: Code OSS via web, with debug extensions for PHP.

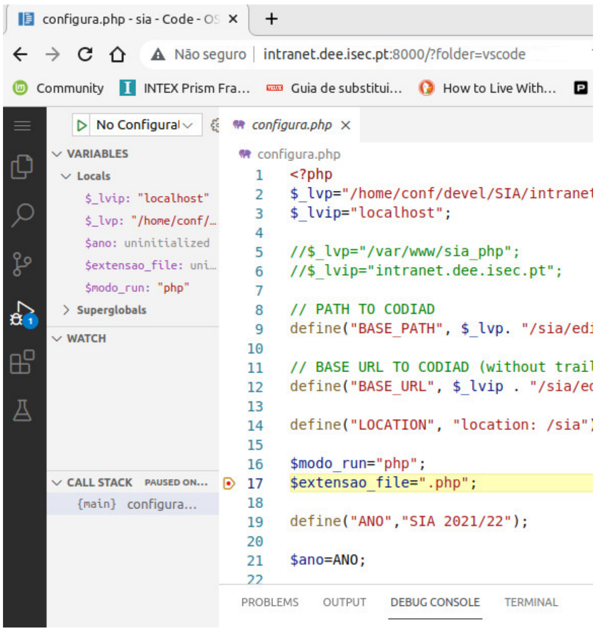


Fig. 9: Online PHP with step by step debug session.

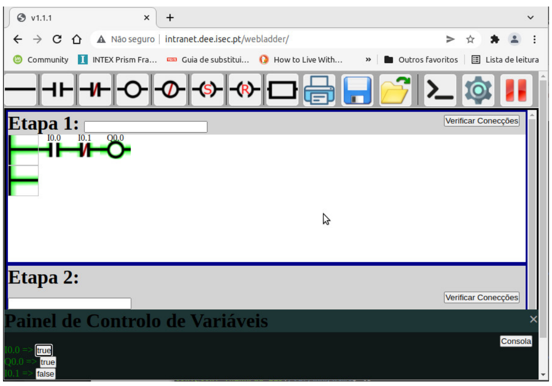


Fig. 10: Experimental Web Ladder Editor - Proof of concept.

## REFERENCES


[1] GNU, "Octave Online". [Online]. Available: octave-online.net

[2] Mathworks, "Matlab Online". Accessed: March 2022. [Online]. Available: matlab.mathworks.com

[3] Microsoft, "Vscode for Web". Accessed: March 2022. [Online]. Available: vscode.dev

[4] "Developed Web-platform". [Online]. Available: intranet.dee.isec.pt/sia and intranet.dee.isec.pt/mcq

[5] Tools, Codiad IDE. [Online]. Available: codiad.com & php-einfach.de

[6] Souza, D., Coutinho, J., & Sousa, R. D. (2022). LoopAcademic: Helping the Learning and Teaching of Introductory Programming. In Perspectives and Trends in Education and Technology (pp. 561-571). Springer, Singapore.

[7] Mosconi, M. and Porta, M. (1998). A Modular Software Platform for Programming Web Applications, Proc. of the First Int. Workshop on Interaction Agents, May 24, 1998, L'Aquila, Italy, pp. 61-63.

[8] Cui, Y., Zhao, G., & Zhang, D. (2022). Improving students' inquiry learning in web-based environments by providing structure: Does the teacher matter or platform matter? British Journal of Educational Technology

[9] Jacko, P., Bereš, M., Kováčová, I., Molnár, J., Vince, T., Dziak, J., Fecko, B., Gans, Š. and Kováč, D. (2022). Remote IoT Education Laboratory for Microcontrollers Based on the STM32 Chips. Sensors, 22(4), 1440.

[10] Brum, D. K. (2021). PanScript: a free platform for teaching programming in many human languages.

[11] Jordens, X., Wilmart, R., Garone, E., Kinnaert, M., & Catoire, L. (2022). A Project-Based Learning Approach for Building an Affordable Control Teaching Lab: the Centrifugal Ring Positioner. IEEE Access.

[12] Alvarez-Marin, A., & Velazquez-Iturbide, J. A. (2022). Augmented Reality and Engineering Education: A Systematic Review. IEEE Transactions on Learning Technologies.

[13] Sandoval Pérez, S.; Gonzalez Lopez, J.M.; Villa Barba, M.A.; Jimenez Betancourt, R.O.; Molinar Solís, J.E.; Rosas Ornelas, J.L.; Riberth García, G.I.; Rodriguez Haro, F. (2022). On the Use of Augmented Reality to Reinforce the Learning of Power Electronics for Beginners. Electronics 2022, 11, 302.

[14] Al-Azawi, R., Al-Fatma, F., and Al-Blushi, M. (2016). Educational gamification vs. game based learning: Comparative study. Int. J. Innovat. Manag. Technol. 7, 132–136.

[15] Calderón, A., and Ruiz, M. (2015). A systematic literature review on serious games evaluation: An application to software project management. Comput. Educ. 87, 396–422.

[16] Khan, M.A., Merabet, A., Alkaabi, S. et al. Game-based learning platform to enhance cybersecurity education. Educ Inf Technol (2022).

[17] J. Keller, Development and use of the ARCS model of instructional design, J. Instr. Dev. 10 (1987), no. 2, 45–52.

[18] Gomes, A., Santos, A., Martins, N. e Páris, C., "Code Factory - Um Jogo Sério para ensinar a programar", Coleção – Estratégias de Ensino e Sucesso Académico, Editores CINEP/IPC, ISBN: 978-989-98679- 6-3 (impresso) e ISBN: 978-989-98679-7-0 (ebook), 2016.

[19] Paiva, J. C., Queirós, R., Leal, J. P., Swacha, J., & Miernik, F. (2022). Managing Gamified Programming Courses with the FGPE Platform. Information, 13(2), 45.

[20] "DFP - Data Flow Programming, LabView, NodeRed, n8n, Crosser". [Online]. Available: www.ni.com/pt-pt/shop/labview.html, nodered.org, n8n.io, www.crosser.io

[21] "BP - Block Programming, Scratch, Pictoblox, Snap, Hopscotch, Tyner, Beetle Blocks". [Online]. Available: scratch.mit.edu/, thestempedia.com/product/pictoblox, snap.berkeley.edu/index, gethopscotch.com, tynker.com, beetleblocks.com

[22] "Text Programming in Web-based Tools". Ex: codesandbox, codepen.io. [Online]. Available: codesandbox.io/, codepen.io

[23] W3Schools, "Text Programming in Web". [Online]. Available: w3schools.com

[24] Gourav Sharma, "How Artificial Intelligence and Machine Learning are Revolutionizing Software Development". [Online]. Available: www.computer.org/publications/tech-news/trends/ai-is-changing-software-development

[25] Patty Brown, "AI-powered tools are revolutionizing software development industry". [Online]. Available: linkedin.com/pulse/how-ai-powered-tools-revolutionizing-software-industry-patty-brown/

[26] SAS, "Five AI technologies that you need to know". [Online]. Available: www.sas.com