CBLTool: A Web-Based Tool for Documenting Challenge Based Learning Journeys

Júlio Necyk Maia
Department of Informatics, PUC-Rio
Rio de Janeiro, Brazil
julio.maia@les.inf.puc-rio.br

Andrew Diniz da Costa
Department of Informatics, PUC-Rio
Rio de Janeiro, Brazil
andrew@les.inf.puc-rio.br

Juliana Alves Pereira
Department of Informatics, PUC-Rio
Rio de Janeiro, Brazil
juliana@inf.puc-rio.br

Alessandro Garcia
Department of Informatics, PUC-Rio
Rio de Janeiro, Brazil
afgarcia@inf.puc-rio.br

Carlos José Pereira de Lucena
Department of Informatics, PUC-Rio
Rio de Janeiro, Brazil
lucena@inf.puc-rio.br

Abstract

Several active learning methodologies have been adopted to support student learning processes. A methodology that stands out is Challenge Based Learning (CBL), which offers a framework divided into three phases, namely Engage, Investigate and Act. CBL seeks to guide learning through the resolution of real-world challenges. To back up the application of this methodology, educators and students are induced to deal with a plethora of tools, such as, those designed to create text documents, presentations, worksheets and many other artifacts, that make the documentation of learning journeys possible. However, these general-purpose tools do not explicitly incorporate the concepts and structures defined by the methodology adopted, forcing users involved to manually adapt the available resources, define the organization of the journey, and incorporate pedagogical terms and elements. These adaptations can be complex and time-consuming, specially for students who lack deep expertise on relevant methodological aspects. Thus, this paper presents CBLTool, a web-based tool that aims to document learning journeys that apply CBL. This tool provides the core elements of CBL, such as its phases, terms related to each phase, and also allows the development of challenges in individual or collaborative projects. In order to evaluate its use, an empirical evaluation was performed. The results indicate how this tool can allow users to efficiently perform routine tasks along a CBL journey, contributing with the learning process for both experienced and non-experienced users of that methodology.

Demo video: https://doi.org/10.6084/m9.figshare.32393373

Keywords

Challenge Based Learning, Active Learning, Angular, Django, Software Framework

1 Introduction

Challenge Based Learning (CBL) has emerged as a powerful active learning methodology in modern education [1, 23, 29]. This methodology drives learners to tackle real-world problems in their local communities through creative collaboration, deep investigation, and action [23, 29]. However, implementing CBL on a scale introduces substantial operational difficulties. To implement this methodology, practitioners often struggle with the systematic, iterative documentation required across the open-ended stages of the learning journey [32]. Traditional learning management systems offer rigid, linear structures that do not accommodate the fluid, adaptive, and artifact-centered nature of challenge based environments [1, 29]. This problem highlights the urgent need for dedicated, flexible software tools designed around pedagogical abstractions [23, 29].

The CBL methodology has been used in institutions scattered around the world [4, 11, 21, 22, 26, 33, 34], including Brazil [6–8]. It has been applied in high school, graduate and postgraduate courses in software engineering (e.g., [8]) , as well as in specific technology training settings (e.g., [6, 7]). However, users exploring CBL lack support of unified tool support to document their learning journeys. They are left with the option of using various general-purpose tools, such as separate editors for texts, presentations, worksheets, and many other types of artifacts. These generic tools are not interlinked and are not explicitly structured in terms of CBL phases and concepts. This scenario pushes users to manually adapt the available resources, define the organization of the journey, and manually incorporate pedagogical terms and elements. These adaptations are complex and time-consuming, specially for students who lack deep expertise on relevant methodological aspects.

To address this gap, the *CBLTool* was developed as an architectural and conceptual foundation to build CBL resources. CBLTool is a comprehensive web-based application that acts as the primary operational resource for educators and students. CBLTool is engineered as a robust, production-ready system that systematically maps the pedagogical stages of CBL into interactive digital workspaces. By instantiating the CBL's predefined software abstractions, the tool supports students as they dynamically document their progress through projects, phases, individual pages, and customizable content blocks. A major contribution of CBLTool lies in its ability to solve the complex technical challenges associated with distributed, multi-user educational software. In collaborative learning environments, students must work simultaneously on shared documents without data loss, race conditions, or lag to the interface. This article provides a comprehensive overview of how CBLTool achieves these requirements. We discuss its layered architecture, detail the user flows and screens, explain the underlying page-locking concurrency control and autosave mechanisms, and describe the Docker-based containerized infrastructure established for production-like deployments.

The theoretical foundation for CBLTool is reinforced by recent conceptual frameworks that emphasize the complexity of capturing learning data in active environments. Helker et al. [20] propose a multi-level structure to monitor the conditions and outcomes of CBL, arguing that the open-ended nature of such challenges requires robust documentation systems to effectively map long-term student engagement. Complementarily, Wambacq and DePryck [35] validate the CBL Competency Framework against OECD 2030 guidelines, highlighting that the systematization of reflection cycles and the construction of collaborative artifacts depend on specialized digital environments. Together, these works ratify the architectural relevance of CBLTool as an essential tool for operationalizing the collection and organization of pedagogical evidence that traditional documentation methods fail to support.

Our motivation to create the CBLTool is the lack of CBL-based dedicated tools in the web environment to document CBL learning journeys. We conducted a Systematic Literature Review (SLR) aimed at eliciting tools with this purpose and no relevant CBL dedicated tool was found [25]. According to our SRL results, the CBL tool, proposed in this present work, is a novel way of creating and documenting CBL's learning journey, except for iOS-only CBL Nudge [5] tool that merely provides tips for the CBL usage. Papers found in the the literature review brought tools that can be used with CBL, but there was no tool with the CBL concepts intrinsic to its design and structure. Notion [30] was a tool that presents reasonable adaptability to the CBL methodology, being used with CBL by learners, but its usage was considered too limited by users in our empirical evaluation (Section 5.2). Other relevant tools, although general-purpose ones, found in the SLR were Miro [27], Google Docs [14], Google Slides [13] and Google Sheets [15]. The tool reported in [24] is used with CBL and can be adapted to document learning journeys, but it does not bring the CBL methodology's concepts as explicit resources to being used inside the tool's scope.

Different from existing tools, some distinguishing CBLTool features include: edition of personal data specific for each project, creation and management of CBL projects, definition of elements associated with each challenge, documentation of CBL primary concepts, sharing and cooperation in a project with multiple users, well-controlled concurrent use of the same project and pedagogical elements by multiple users, addition of visual components, including text, image and file blocks. The CBLTool architecture is described in detail in Section 3. In order to evaluate its use, an empirical evaluation was performed. The results indicate how this tool can allow users to efficiently perform routine tasks along a CBL journey, contributing with the learning process for both experienced and non-experienced users of that methodology. The CBLTool workflow shows that educators and students can document, share and edit the CBL concepts and phases, helping and guiding a CBL challenge from beginning to the end, with a solid structured design that can support and enhance a CBL project documentation.

2 Challenge Based Learning

Challenge Based Learning (CBL) is an active methodology emerged from "Apple Classrooms of Tomorrow — Today" (ACOT2) [1] initiative. This initiative sought to define the fundamental design principles for 21st-century education. Building upon these principles, Apple, Inc. collaborated with leading educators to refine and validate the CBL methodology [29]. The CBL methodology's framework has its focus on three phases: *Engage*, *Investigate* and *Act*, in which the learner can document their learning journey [3].

The *Engage* phase is responsible for taking learners on relation to some challenge. Initially, a *Big idea* should be defined, which is a theme, area or domain to which the challenge is related. Next, an *Essential Question* should be defined. It is a question that engages learners to reflect about personal interests and the needs of the community. The third concept offered in that phase is *Challenge*, which calls to action the essential question for deeper understanding of the subject considered.

The *Investigate* phase requires research for creating the necessary knowledge foundation to propose some solution that deals with the challenge identified in the previous phase. To conduct this investigation, learners should identify *Guiding Questions* (GQs), which are questions that must be answered during the research process. In order to answer them, *Guiding Activities* (GAs) should be made, i.e., necessary actions that will make it possible to answer GQs. Examples of GAs could be reading a paper or a book and interviewing someone. In addition, resources used per GA are called *Guiding Resources* (GR). Examples of GRs could be books, academic papers, web sites visited, people interviewed, and the like. At the end of this phase, learners analyze the data and consolidate the knowledge acquired from the research.

In the last phase, called *Act*, learners are expected to transform the knowledge acquired from the previous phase into a concrete *Solution* to the proposed challenge. After developing a solution, learners proceed with its *Implementation* in a real or simulated context. Next, some *Evaluation* is conducted to assess the solution's impact, effectiveness and limitations. Reflection is another activity encouraged by CBL and can take place at any stage of the challenge, allowing learners to analyze what worked well, what could be improved, and how their understanding evolved throughout the learning journey. Through this process, CBL fosters critical thinking, creativity, teamwork, and problem-solving skills. Consequently, the CBL methodology has been widely and extensively used in industry-academia collaborations and digital learning environments [32].

3 Architecture, Interface and Collaboration

In this section, CBLTool is presented in detail in terms of its requirements, architecture, user interface, and technologies used through the development and conceptual processes.

CBLTool. The main features and requirements of the tool are as follows. First, it supports registration, login with password encryption in hash format. Each user is authenticated to access each account on the system through an access token. Users can create, delete, and order projects (e.g., by its creation date). It allows editing project title and image as well as personal profile data. Users can document their learning journey by creating pages for each of the three CBL phases. Pages can contain visual components, text, image, and file blocks. New users can be invited to a single project, in which they can collaborate constrained by the lock and unlock strategy to concurrent edition. Also, the tool aims at offering easy usability to find and use the options provided. It supports the compatibility to various browsers, such as Google Chrome [16], Mozilla Firefox [28] and Opera[31]. All users' and projects' data are persisted in a database.

Architecture of CBLTool. CBLTool's architecture is designed to separate user interface concerns from the complex domain logic of the domain classes. Client and server communication is handled entirely through RESTful APIs utilizing JSON as the data exchange format. This modular design guarantees that the frontend can remain responsive, while the backend focuses on the domain's principal concerns, such as data integrity, pedagogical features, access control, and concurrency management across different team projects. To illustrate the architectural structure of CBLTool, Figure 1 shows the main components of the architecture.

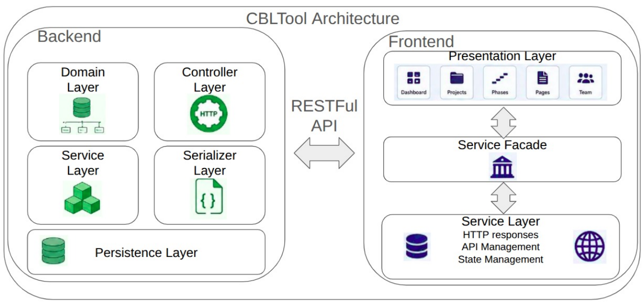

Figure 1: CBLTool Architecture Overview.

The backend of CBLTool does not strictly follow the Model-View-Controller (MVC) pattern in favor of a service-oriented layered architecture. The system is split into four highly cohesive layers: the Domain, Controller, Serializer and Service Layers. Such separation guarantees that changes to business rules or data model do not affect the whole system, making future maintenance easier and contributing to the efficient scalability when implementing new functionality [9, 10]. The Domain layer defines the main object-relational models mapped directly to the frozen spots such as User, Project, Phase, and Page, preserving the essence of CBL methodology. The Controller layer deals with HTTP requests processing and initial validation of requests. It is implemented using generic views from DRF framework. The Controller layer handles HTTP requests and coordinates initial validation. Built on DRF's generic classes, views and translate incoming requests into domain-specific actions. The Serializer layer enforces strict data validation, schema enforcement, and (de)serialisation of JSON objects. It ensures secure input sanitisation before persisting changes.

Finally, the Service layer contains the main business logic of the application. The purpose of this layer is to separate domain logic from its persistence implementation, communicating only with the access objects of the database providing basic get, set, and create operations. By enforcing this strict separation of concerns, the backend can be easily implemented and adapted by any developer. For instance, the persistence layer can be updated or optimized without affecting the core controller structures or public API endpoints. This modularity directly reflects the reusable spirit of the underlying instantiation tool, proving that framework-based application development can yield clean, maintainable backend architectures.

The frontend architecture is built using Angular [18]. It is a single-page application that is interactive. The Facade pattern is used to manage the state of the application and prevent the UI components from being too dependent on the backend services. The Facade pattern implies that the backend endpoints do not interact with components directly. Instead they communicate with service facades that manage the underlying API logic and local state. The service facades are the centralized structures that communicate frontend with backend and serve as an intermediary for the UI components. They manage asynchronous HTTP operations, keeping the components' code simple. The State and Persistence services manage the client-side memory buffers. This allows for navigation between different pages. It also manages data transmissions to prevent server strain. By managing the state at the service level the application ensures that the view synchronization, navigation caching and background saving occur seamlessly.

Figure 2 shows the main classes of the Domain layer that represent the core CBL concepts. Each user (CBLUser) can manage multiple CBL projects (*Project* class), whose each project contains the three phases (*Phase* class) offered by the methodology. Each phase is associated with important concepts (e.g., *Big Idea* class, *Essential Question* class etc.) and the data related per Phase are offered through web pages (*Page* class). Additionally, each project can have reflections associated. Last, but not least, the *ProjectStrategy* class allow define how projects are listed to the users. Currently, the projects are displayed according to their creation date order.

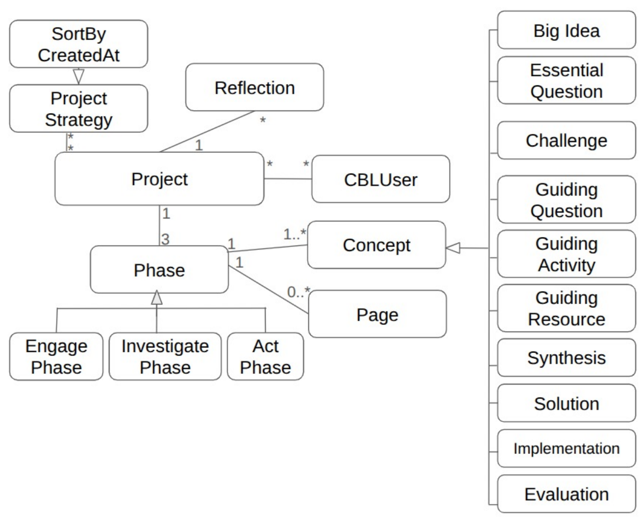

Figure 2: CBLTool Domain Layer Class Diagram.

Use Case: UI Flows and User Experience. The user interface of CBLTool is designed to make it easy for students to use. It abstracts the complexity of CBL and guides students, through a digital representation of their learning journeys. The UI is designed to reduce load allowing student groups to focus on solving their challenges within each CBLTool's project created, giving edition and customization liberty with the present features, like blocks and CBL phases' structure.

The application starts at the Login Screen, where users can log in or navigate to the Register Screen to create an account. Once logged in, the student is taken to their dashboard, the projects workspace. If the user has no teams, the dashboard displays a message encouraging them to start a project. When creating a project, users can define a title and upload a cover image. Active projects are displayed as cards, which users can search, sort or delete. This makes it easy for users to work on challenges at the same time keeping team members, resources and requirements organized. This entry flow ensures that users are immediately oriented within their specific collaborative scopes. By organizing workspaces around individual projects, the tool makes it easy for students to participate in multiple challenges simultaneously, keeping team members, resources, and requirements strictly isolated and organized, matching operational capabilities found in commercial alternatives.

When one starts a project, the CBL workspace becomes visible. This workspace is set up around three phases: Engage, Investigate and Act. The user can move around these phases using the menu at the top of the page. The first page of each phase is organized around what the student need to learn. For example, when the student is in the Engage part, the first page helps to write down the Big Idea, Essential Question and Challenge. The next pages in each phase are more open so one can use them in a way that works for his/her project. This way of setting things up helps connect what the student learns in school to how the software is designed. The CBL workspace is set up to help the learner. It also gives the student the freedom to work on his/her project in a way that makes sense for he/she and his/her team.

In each phase, users have the freedom to create their documentation using a side panel that provides three types of drag and drop blocks: (i) **Text Block** is a container where users can write notes, ideas and summaries of what they have learned. (ii) **Image Block** allows users to upload images and figures that can be resized by dragging their borders, and (iii) **File Block** enables users to add files from their computers in different formats.

To support the drag and drop feature in the CBLTool architecture (see Figure 1), we implemented a DragDropService in the Service Façade and a DropzoneComponent in the Service Layer. These components communicate through the Observer pattern [12], ensuring synchronization between user interactions and the document layout. Users can freely move, delete and add blocks, making the documentation process more flexible and intuitive than the previous approach, which relied on a single text editor. Figure 3 shows an example of a page having the multiple blocks.

Collaboration and Concurrency Features. CBL is a team effort. CBLTool is made to help many users work together at the same time. When students work in a group and have to finish something, they all need to look at and work on the same project files. If the system is not well structured, they might overwrite each others work, which can be really frustrating and cause people to lose their work. To stop this from happening and keep everyone work safe, CBLTool uses page-level locking. Figure 4 exemplifies two users at the same page. When a student starts editing a page, this first users obtains the lock, locking the page so nobody else can edit it at the same time. If someone else tries to edit the page, the server prohibits its use and shows a lock icon on the page. This lock is always on and gets updated in real time. If they close their browser or stop working on the page for a long time, or change between pages, the lock goes away and someone else can start editing the page. CBLTool enables people to work together, but keeping everyone work safe while collaborating in real time.

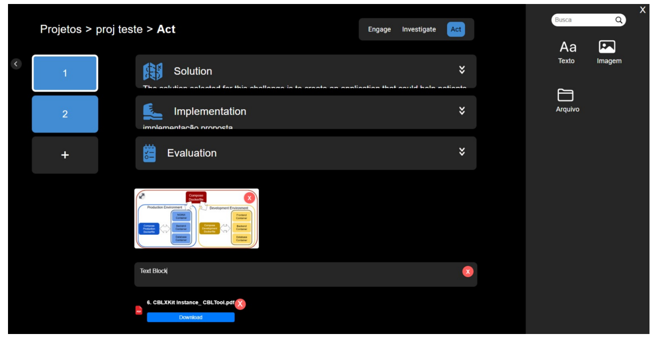

Figure 3: Drag-and-Drop Document Editor showcasing active content blocks being arranged in real-time.

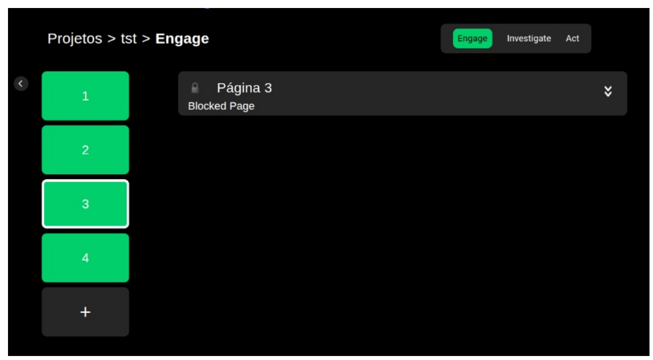

Figure 4: Active Concurrency Lock UI displaying the gray lock icon and read-only mode for concurrent users.

The CBLTool has a system that helps keep everything up to date. This system is called the PhaseAutoSaveService. It checks for changes to the content blocks and saves them automatically. The user does not have to click any buttons to save his/her own work. The CBLTool does not save every keystroke. This is because it would send many requests to the server. Instead it waits for a time after one stops typing; then, it saves the work produced. When the CBLTool saves the work, it does two things. First, it updates the information in the database. This information is like a list of details about each block of content. Then, it uses the HtmlGenerationService to make a HTML file. This file functions as a snapshot of a page inside each phase. It can guarantee the data edited in each page, phase and project are persisted. The CBLTool does this to make sure that the users' work is safe.

The tool can always get the user to his/her work even if something happens to the database or the system. The CBLTool follows the rule of making it sure that the network and server are not too busy, while enabling the user to have a good experience. The persistence of the users' work occurs in the background, saving all data created and configured in every element presented on the screen.

Students can feel confident that their work is being saved and they can keep working without worrying about losing anything, so they can focus on their learning journey.

4 Empirical Evaluation of CBLTool

To evaluate the easy of use and performance of CBLTool, we conducted an experiment with people who actually use CBL. We gave them a procedure so they could try and test the tool and give feedback about it. We would want to know how people compare the CBLTool with the tools they used before with CBL. In particular, we wanted to know if CBLTool assists people in keeping track of what they learn along their educational journey. To proceed with the experiment, we designed a survey. It relied on Google Forms [17] to centralize the collected data from each participant; it offered a permanent link to be accessed by the participants during the experiment period, which lasted for approximately two weeks.

Evaluation Methodology and Participants. The evaluation used a way to choose participants who worked with the CBL framework before. We got 8 people who were already using it in different scenarios. We first asked them some questions about themselves like what they studied, what they do for work, how long they used CBL, and if they used tools to keep track of things. The recruited group had 6 students who were still studying for their bachelors degree, and 2 who were studying for their masters degree. Most of them were studying Software Engineering or Computer Science. One person was studying Business Administration. Seven out of eight people said they used Notion [30] to organize and track their CBL work. The evaluation required each person to perform 9 tasks with the CBLTool. These tasks showed the process of setting up the documentation and management of a team project. The tasks included configuring CBLTool, documenting CBL artifacts, and managing CBL challenges. The 9 tasks are listed in Table 1.

Table 1: Usability Evaluation Task Sequence

Performance Evaluation Tasks (T1–T9)
T1 (User Registration): Setting up an account through the signup form.
T2 (Authentication): Authenticating and logging into the dashboard.
T3 (Project Creation): Initializing a new collaborative CBL project.
T4 (Phase Milestone Editing): Documenting the *Big Idea*, *Essential Question*, and *Challenge* in the *Engage* phase.
T5 (Dynamic Block Manipulation): Adding, dragging, and configuring content blocks (text, image, and files).
T6 (Inter-Phase Navigation): Managing sub-pages within the workspace.
T7 (Guided Investigation): Structuring Guiding Questions, Activities, and Resources in the *Investigate* phase.
T8 (Action Documentation): Formulating the solution and evaluation criteria in the *Act* phase.
T9 (Multi-Project Management): Initializing a secondary project and using sorting features.

To measure how well participants performed, they were required to write down the start and end times per each task. Because of the survey model limitation, we were unable track seconds, only the difference between the start and end of each task with a timestamp with the hour and minute (e.g., 12:55 and 12:57). We changed task times under a minute to 1.0 minute so the data would be consistent. After finishing the tasks, participants filled out a survey with ended questions to give feedback, share advantages, disadvantages as well as suggesting improvements.

Table 2: Task Completion Times per Participant (in Minutes)

Participant	T1	T2	T3	T4	T5	T6	T7	T8	T9	Total Time
P1	2	1	2	2	2	1	1	1	1	13
P2	2	1	1	1	1	1	1	1	1	10
P3	2	1	3	3	1	2	5	3	1	21
P4	4	1	3	2	3	2	2	2	1	20
P5	2	1	1	2	1	1	1	2	1	12
P6	5	1	2	2	2	1	1	3	1	18
P7	11	1	2	5	2	3	2	2	1	29
P8	2	1	1	2	3	2	1	1	1	14
Average	**3.75**	**1.00**	**2.00**	**2.38**	**1.88**	**1.63**	**1.75**	**1.88**	**1.00**	**17.25**

Research Questions. Next, our research questions (RQs) are defined.

(RQ1) How fast can CBLTool help CBL practitioners in their challenge documentation? By analyzing different web tools used with CBL, we noticed that there are currently no tools designed around CBL principles and concepts for web environment. With that being said, a lot of different tools can be used for CBL documentation, but they all need to be adapted accordingly to the CBL concepts and framework, and it can cause confusion and be too *time-consuming* for this adaptation. From a dedicated tool that already brings all the methodology's concepts, phases, and steps, the documentation process can significantly be improved by eliminating the need for manual adaptation. The creation of a web tool, such as CBLTool, designed around CBL concepts can enhance the execution time and experience of documenting CBL learning journeys. Then, we measured how much time was spent by participants performing the 9 typical tasks with CBL.

(RQ2) Which features should a web tool have for supporting CBL-based documentation? First, the CBL methodology has three phases and each phase got its own concepts. The first feature should support these phases and concepts, so that the user can navigate through the CBL phases without needing to adapt a generic tool. Other features to be concerned are: (i) freely editing text documentation, (ii) adding images, and (iii) as CBL has specific artifacts' creation in its methodology, it is important to add files for these artifacts in any phase of the methodology. With these features being highlighted, the CBLTool was created to address these features, and later on bring more features that could support the CBL methodology's documentation process. To address this research question, we asked qualitative feedback about the usefulness of certain CBLTool features as compared to existing tools.

5 Empirical Results

5.1 (RQ1) Quantitative Performance Analysis

The experiment was conducted in Portuguese so participants could understand the questions easily. We then translated their answers to English. We looked at how it took to complete each task to see how well users did and how easy the tool was to use. The Table 2 shows task completion times for each participant across nine tasks. The results show that CBLTool helps users complete tasks quickly and efficiently. The average time to complete all nine tasks was 17.25 minutes. Simple tasks like login and creating a project took one minute. The main documentation tasks and block-dragging interactions took between 1.63 and 2.38 minutes.

One task that stood out was user registration, which took an average of 3.75 minutes. One participant took 11 minutes. We then checked why this was the case. Participants said they were sometimes hesitant to agree that certain types of information were *mandatory*; the form asked for information like birth date and phone number and they do not feel comfortable providing them as these information are usually confidential and already stored by the university system. Also, they would not like to have the educators contacting the via their phone numbers.

> *"Why is it relevant to know my birthday or cell phone number? ... Personally, in situations like this, I do not give out my personal info."*

Participant 7 had a hard time with the registration task. They took 11 minutes to complete it. This was much longer than the average and counted as an outlier. However, overall, the CBLTool helped users complete pedagogical tasks, which we considered more challenging, very quickly. For instance, tasks like project creation, were fast because of auto login from the browser and project creation had a quick task, only naming the project and uploading a picture if the user wants. The registration task was harder then predicted, being one of the tasks that took more time to be completed in the overall time. In response to this finding, these fields should be configured as optional to reduce cognitive friction, improve user trust, and speed up the on boarding process, and also made some of the fields optional, like birth date and telephone.

5.2 (RQ2) Comparative and Qualitative Analysis

The qualitative feedback we got from participants shows that using a tool dedicated for CBL is quite positive. According to them, this is much better than using generic tools that need to be adapted to be used with the methodology, included Notion, Miro, Google Docs, and others. They considered that, as CBLTool is designed based on the methodology's concepts, all the features were useful and pleasant to use. People liked that the tool was explicitly structured around the three phases of CBL, which were already settled up: Engage, Investigate and Act. This structure helped teams stay organized and follow the steps in a disciplined manner. Participants also liked the way the tool looked and worked; below is a representative comment of one participant:

> *"It's really cool to drag the blocks! I imagine that the search bar is for the N blocks that are yet to come. The images are resized, both vertically and horizontally. However, I wonder if image resizing doesn't need to be treated differently. For example, I tested a vertical photo and a horizontal photo, and both ended up square. Also, I uploaded a zip file and it worked perfectly."*

The only problem people found was that the image blocks in the tool were always the same size. This meant that large pictures could not be uploaded without distorting the image. Some other changes we could do to make the tool more efficient are: (i) make it clearer when the user is moving blocks, like a area to indicate where to drop the blocks, highlighting the dropzone, and (ii) letting the title of pages being directly edited and adding a Google Sign-in feature. Overall, this first study shows that the tool is easy to use and its features are really helpful for educators and students who are involved in CBL projects.

5.3 Limitations and Threats to Validity

Next, we describe the limitations and threats to validity exist within the empirical evaluation. To resolve the limitations that arose, a new revision on the project's code was produced, with a more in-depth look into UX and UI elements to upgrade user experience on CBLTool usage. Some feedbacks collected were considered as points of improvements, with one of the suggestions being to support Google Signin [19] and Apple Id [2] to accelerate login and registration in our application.

Sample Size and Selection Bias: The study was limited to eight participants, which restricts the generalizability of the quantitative performance metrics. The cohort was selected from students and practitioners within our academic ecosystem. Their high baseline familiarity with web systems and software technologies may have resulted in faster task completion times than would be observed in non-technical user groups.

Evaluation Environment Constraints: The evaluation did not include active screen or audio recording during testing sessions, meaning subtle usability hesitations or navigation issues had to be inferred solely from written comments and logged task durations. Due to space constraints, other threats are discussed in [25].

6 Conclusion and Future Work

The CBLTool is a collaborative platform that supports the CBL methodology. The empirical study showed that participants found the platform easy to use and its features useful for supporting collaborative learning. In particular, features such as automatic saving and document locking help prevent conflicts during teamwork. As future work, we plan to expand the empirical study to include a larger and more diverse group of participants, including educators and students from different universities, regardless of their prior experience with CBL, while also improving the evaluation metrics. We also intend to provide a dedicated access mode for educators, enabling them to monitor students' progress, assess their work, and support them throughout their CBL journey through real-time dashboards. Finally, we plan to integrate AI-based features, such as auto-completion, chatbots, and personal assistants, to accelerate documentation, improve the user experience, and provide guidance on both the platform and the application of the CBL methodology.

Artifact Availability

The source code for CBLTool is available at https://figshare.com/articles/software/CBLXToolPFP/32393391 or https://doi.org/10.6084/m9.figshare.32393391.v8 under the MIT license.

Acknowledgements

Project supported by the Ministry of Science, Technology and Innovation (MCTI), with funding provided under Law No. 8,248/1991 (Brazilian Information Technology Law), Agreement No. 01245.0208 20/2023-1 TPA/195/SOFTEX/PUC-Rio, dated November 27, 2023. The LLMs usage was initially for reference and to start some sections and later on fully adapted or changed by the authors.

References

[1] Apple Inc. 2008. *Apple Classrooms of Tomorrow-Today: Learning in the 21st Century*. Technical Report. Apple, Inc., Cupertino, CA, USA.

[2] Apple Inc. 2019. Sign in with Apple. https://developer.apple.com/sign-in-with-apple/. Accessed: 2026-05-24.

[3] Apple Inc. 2024. *Challenge Based Learning 1 - Framework Overview*. Technical Report. Apple, Inc., Cupertino, CA, USA.

[4] Neusi Aparecida Navas Berbel. 2021. Marie Leijon, Petri Gudmundsson, Patricia Staaf and Cecilia Christersson. *Innovations in Education and Teaching International* 59, 9 (2021), 609–618. doi:10.1080/14703297.2021.1892503

[5] Challenge Institute. 2019. CBL Nudge. https://www.challengebasedlearning.org/project/cbl-nudge/. Accessed: 2024-03-05.

[6] Andrew Diniz da Costa, Hendi Lemos Coelho, Ricardo Almeida Venieris, Carlos José Pereira de Lucena, Gustavo Robichez Carvalho, and Marcelo Fernandes Pereira. 2020. Assessing a Multidisciplinary Group of Undergraduate Students Applying the Challenge Based Learning Methodology to Learn Mobile Development. In *Proceedings of the XXXIV Brazilian Symposium on Software Engineering (SBES '20)*. Association for Computing Machinery, New York, NY, USA, 700–709. doi:10.1145/3422392.3422484

[7] Andrew Diniz da Costa, Carlos José Pereira de Lucena, Hendi Lemos Coelho, Gustavo Robichez Carvalho, Hugo Fuks, and Ricardo Andrade Venieris. 2018. Multidisciplinary Groups Learning to Develop Mobile Applications from the Challenge Based Learning Methodology. In *Proceedings of the XXXII Brazilian Symposium on Software Engineering (SBES '18)*. Association for Computing Machinery, New York, NY, USA, 318–327. doi:10.1145/3266237.3266256

[8] Andrew Diniz da Costa, Carlos José Pereira de Lucena, and Gustavo Robichez de Carvalho. 2023. Applying Remotely the Challenge Based Learning Methodology in Undergraduate and Postgraduate Disciplines. In *Anais do III Simpósio Brasileiro de Educação em Computação (EDUCOMP 2023)*. Sociedade Brasileira de Computação (SBC), Porto Alegre, RS, Brazil, 163–172. doi:10.5753/educomp.2023.228163

[9] Django Software Foundation. 2005. Django: The Web framework for perfectionists with deadlines. https://www.djangoproject.com. Accessed: 2024-03-10.

[10] Encode OSS Ltd. 2011. Django REST Framework. https://www.django-rest-framework.org. Accessed: 2024-03-10.

[11] Siobhán E. Gallagher and Timothy Savage. 2020. Challenge-based learning in higher education: an exploratory literature review. *Teaching in Higher Education* 28, 6 (2020), 1135–1157. doi:10.1080/13562517.2020.1863354

[12] Erich Gamma, Richard Helm, Ralph Johnson, and John Vlissides. 1994. *Design patterns: Elements of reusable object-oriented software*. Addison-Wesley Professional, Boston, MA, USA.

[13] Google LLC. 2006. Conte Histórias Impactantes com o Google Slides. https://www.google.com/slides/about/. Accessed: 2017-03-12.

[14] Google LLC. 2006. Google Docs: Documents. https://docs.google.com/document/u/0/. Accessed: 2018-03-03.

[15] Google LLC. 2006. Planilhas On-line e Colaborativas. https://workspace.google.com/products/sheets/. Accessed: 2017-09-23.

[16] Google LLC. 2008. Google Chrome Browser. https://www.google.com/chrome/. Accessed: 2018-03-22.

[17] Google LLC. 2008. Google Forms. https://docs.google.com/forms/. Accessed: 2025-05-05.

[18] Google LLC. 2010. Angular: The modern web developer's platform. https://angular.io. Accessed: 2024-03-10.

[19] Google LLC. 2015. Google Sign-In. https://developers.google.com/identity/sign-in/web/. Accessed: 2026-05-24.

[20] Kerstin Helker, Miguel Bruns, Isabelle M. M. J. Reymen, and Jan D. Vermunt. 2025. A framework for capturing student learning in challenge-based learning. *Active Learning in Higher Education* 26, 1 (2025), 213–229. doi:10.1177/14697874241230459

[21] Kerstin Helker-Schlotmann, Isabelle M. M. J. Reymen, Miguel Bruns, and Jan D. Vermunt. 2026. Re-designing to challenge-based learning – what does it mean for student learning processes and outcomes? *European Journal of Engineering Education* (2026). doi:10.1080/03043797.2026.2645648

[22] Vianney Lara-Prieto, Jorge Membrillo-Hernández, and Patricia Caratozzolo. 2022. *Implementation of the Challenge-Based Learning Approach at the Tecnologico de Monterrey, Mexico*. Emerald Publishing Limited, 69–88. doi:10.1108/978-1-80117-490-920221004

[23] Challenge Based Learning. 2024. The Challenge Based Learning Framework. https://www.challengebasedlearning.org/framework/

[24] Chacharin Lertyosbordin, Sorakrich Maneewan, and Surapon Boonlue. 2021. Conceptual Framework of Google's Online Learning Tools for Python Programming Activity on Challenge-Based Learning. *Turkish Online Journal of Qualitative Inquiry (TOJQI)* 12, 6 (July 2021), 9659–9666.

[25] Júlio Necyk Cavalcanti Maia. 2026. *CBLXKit: Framework para Desenvolver Ferramentas Web que Ampliem a Experiência de Uso do Challenge Based Learning*. Dissertação (Mestrado em Informática). Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro, Brasil.

[26] Jorge Membrillo-Hernández, Miguel de Jesús Ramírez-Cadena, Carlos Caballero-Valdés, Ricardo Ganem-Corvera, Rogelio Bustamante-Bello, José Antonio Benjamín Ordoñez-Díaz, and Hugo Elizalde-Siller. 2018. Challenge-based Learning: The Case of Sustainable Development Engineering at the Tecnológico de Monterrey, Mexico City Campus. *International Journal of Engineering Pedagogy (iJEP)* 8, 3 (2018), 137–144. doi:10.3991/ijep.v8i3.8007

[27] Miro. 2011. Miro - Free Digital Whiteboard for Online Education. https://miro.com/education-whiteboard/. Accessed: 2024-03-27.

[28] Mozilla Foundation. 2002. Mozilla Firefox Browser. https://www.mozilla.org/firefox/. Accessed: 2009-07-04.

[29] Mark Nichols and Karen Cator. 2008. *Challenge Based Learning: Take action and make a difference*. Technical Report. Apple, Inc., Cupertino, CA, USA.

[30] Notion Labs, Inc. 2013. Notion Workspace Tool. https://www.notion.so/pt-br. Accessed: 2022-05-16.

[31] Opera Software AS. 1995. Opera Browser. https://www.opera.com/. Accessed: 2026-05-24.

[32] Stefano Perna, Moritz Philip Recke, and Mark H. Nichols. 2023. *Challenge Based Learning: A Comprehensive Survey of the Literature*. Research Report. The Challenge Institute, Austin, TX, USA. https://www.challengeinstitute.org/CBL_Literature_Survey.pdf

[33] May Portuguez Castro and Marcela Georgina Gómez Zermeño. 2020. Challenge Based Learning: Innovative Pedagogy for Sustainability through e-Learning in Higher Education. *Sustainability* 12, 10 (2020), 4063. doi:10.3390/su12104063

[34] Antoine Van den Beemt, Gerard van de Watering, and Myriam Bots. 2023. A framework for capturing student learning in challenge-based learning. *Higher Education* (2023). doi:10.1007/s10734-023-01032-x

[35] Ilse Wambacq and Koen DePryck. 2024. The CBL Competency Framework. In *Proceedings of EdMedia + Innovate Learning 2024*. Association for the Advancement of Computing in Education (AACE), Brussels, Belgium, 366–372. https://www.learntechlib.org/primary/p/224570/