Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Reducing Failure Rates in Introduction to Computer Systems: A Problem-Based Learning Approach with RISC-V Assembly Language

Thales Lima Menezes

Artigo apresentado como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Marcus Vinicius Lamar

Brasília
2025

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Marcelo Grandi Mandelli

Banca examinadora composta por:

Prof. Dr. Marcus Vinicius Lamar (Orientador) — CIC/UnB
Prof.ª Dr.ª Germana Menezes da Nóbrega — CIC/UnB
Prof.ª Dr.ª Maristela Terto Holanda — CIC/UnB

CIP — Catalogação Internacional na Publicação

Menezes, Thales Lima.

Reducing Failure Rates in Introduction to Computer Systems: A Problem-Based Learning Approach with RISC-V Assembly Language / Thales Lima Menezes. Brasília : UnB, 2025.
15 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2025.

1. Reprovação Estudantil, 2. Linguagem Assembly RISC-V,
3. Videoaulas, 4. Educação Remota, 5. Aprendizagem Baseada em Problemas (ABP)

CDU 37

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Reducing Failure Rates in Introduction to Computer Systems: A Problem-Based Learning Approach with RISC-V Assembly Language

Thales Lima Menezes

Artigo apresentado como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Marcus Vinicius Lamar (Orientador)
CIC/UnB

Prof.ª Dr.ª Germana Menezes da Nóbrega
CIC/UnB

Prof.ª Dr.ª Maristela Terto Holanda
CIC/UnB

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 28 de julho de 2025

Dedicatória

Dedico esse trabalho a todos aqueles que ajudaram a suportar as dores da construção e aplicação do mesmo; e para cada aluno que evoluiu pelos frutos desse trabalho.

Agradecimentos

Agradeço a Deus por ter me acompanhado durante essa pesquisa e ouvido as orações de amigos e familiares que ansiavam pela conclusão deste trabalho. Agradeço ao Professor Dr. Marcus Vinicius Lamar, por acreditar no *pitch de elevador* de um aluno com muitos sonhos e pela orientação paciente durante anos turbulentos. Agradeço o apoio de toda minha família, proporcionando um ambiente tranquilo e estável para compilar os resultados dessa pesquisa. Agradeço aos amigos, que dedicaram tempo revisando, indicando leituras, compartilhando conhecimentos de suas áreas de estudo e acompanharam os prantos e surtos que estiveram presentes durante a execução dessa pesquisa. Agradeço aos monitores da disciplina de Introdução aos Sistemas Computacionais (ISC) que me receberam na disciplina durante o período de intervenção e expandiram as iniciativas ao longo dos anos seguintes. Agradeço aos alunos que se dispuseram a participar dos aulões e interagiram com os vídeos do canal do YouTube, incentivando a criação de conteúdo com feedback genuíno e apresentando trabalhos mais impressionantes a cada semestre. Agradeço à Universidade de Brasília (UnB) e ao Departamento de Ciência da Computação (CiC) por terem me proporcionado o ambiente e os recursos para a realização deste trabalho. E, por fim, à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) por patrocinar o acesso aos artigos.

Resumo

Esta pesquisa apresenta uma metodologia complementar para aprimorar a disciplina de Introdução aos Sistemas Computacionais (CIC0003) (ISC), criada em 2015-2, na Universidade de Brasília (UnB), no Brasil. O objetivo é reduzir as taxas de reprovação dos alunos na disciplina, auxiliando-os na elaboração do projeto final da disciplina, por meio de melhorias na experiência de aprendizagem da linguagem Assembly RISC-V, utilizando técnicas de Aprendizagem Baseada em Problemas (ABP). A metodologia vem sendo validada, monitorada e aprimorada iterativamente desde 2019-1, evoluindo de sessões presenciais para videoaulas remotas, focadas na resolução de problemas e no desenvolvimento de jogos. Os resultados indicam uma redução de 69,14% nas taxas de reprovação na disciplina de ISC, comparando os dados de 2018-2, antes da primeira intervenção, até 2025-1, o semestre mais recente. Trabalhos futuros sugerem refinar a experiência de aprendizagem, criar ferramentas de suporte inteligentes e estender a intervenção a disciplinas relacionadas em nosso programa, como Organização e Arquitetura de Computadores (CIC0099) (OAC).

Palavras-chave: Reprovação Estudantil, Linguagem Assembly RISC-V, Videoaulas, Educação Remota, Aprendizagem Baseada em Problemas (ABP)

Abstract

This research presents a complementary methodology to enhance the Introduction to Computer Systems (CIC0003) (ISC) course, created in 2015-2, at the University of Brasília (UnB), in Brazil. The goal is to reduce student failure rates in ISC by helping students make the final project of the course, through improvements to the learning experience of RISC-V Assembly language using Problem-Based Learning (PBL) techniques. The methodology has been iteratively validated, monitored and improved since 2019-1, evolving from in-person sessions to remote video lessons focused on problem-solving and game development. Results indicate a 69.14% reduction in ISC failure rates comparing the data from 2018-2, before the first intervention, until 2025-1, the most recent semester. Future work suggests refining the learning experience, creating intelligent support tools, and extending the intervention to related courses in our program, such as Computer Organization and Architecture (CIC0099) (OAC).

Keywords: Student Failure, RISC-V Assembly Language, Video Lessons, Remote Education, Problem-Based Learning (PBL)

Reducing Failure Rates in Introduction to Computer Systems: A Problem-Based Learning Approach with RISC-V Assembly Language

Thales Lima Menezes
Departament of Computer Science
University of Brasília
Brasília, Brazil
thlmenezes@pm.me

Prof. Dr. Marcus Vinicius Lamar
Departament of Computer Science
University of Brasília
Brasília, Brazil
lamar@unb.br

***Abstract*—**
This research presents a complementary methodology to enhance the Introduction to Computer Systems (CIC0003) (ISC) course, created in 2015-2, at the University of Brasília (UnB), in Brazil. The goal is to reduce student failure rates in ISC by helping students make the final project of the course, through improvements to the learning experience of RISC-V Assembly language using Problem-Based Learning (PBL) techniques. The methodology has been iteratively validated, monitored and improved since 2019-1, evolving from in-person sessions to remote video lessons focused on problem-solving and game development. Results indicate a 69.14% reduction in ISC failure rates comparing the data from 2018-2, before the first intervention, until 2025-1, the most recent semester. Future work suggests refining the learning experience, creating intelligent support tools, and extending the intervention to related courses in our program, such as Computer Organization and Architecture (CIC0099) (OAC).

***Index Terms*—Student Failure, RISC-V Assembly Language, Video Lessons, Remote Education, Problem-Based Learning (PBL)**

I. Introduction

The Department of Computer Science (CIC), at University of Brasília (UnB) in Brazil, has been taking numerous actions over the years to reduce dropout and failure rates: reformulating the curriculum, modifying the first semester courses; reformulating the first programming course, Algorithms and Computer Programming (CIC0004) (APC) and second programming course Data Structures (CIC0090) (ED); characterizing the profile of students who enter the course; among other actions, as mentioned in [1].

But the curriculum also has another challenge that students, and faculty, need to face: the Computer Organization and Architecture (CIC0099) (OAC) course.

The OAC course serves as a foundational prerequisite for several advanced courses, including Basic Software (CIC0104), Concurrent Programming (CIC0202), Fundamentals of Operating Systems (CIC0205), Compilers (CIC0204). The course aims to bridge the gap between hardware and software, focusing on core concepts that underpin modern computers and exploring organizational paradigms that determine the capability and performance of computing systems.

Upon completion of the course, students should understand how a computer system works and why it exhibits specific performance characteristics. They should also acquire proficiency in assembly programming and basic hardware design involving microprocessors, using hardware description languages (HDL) and FPGA-based prototyping systems.

However, the course has historically faced high failure rates (sum of failed and withdrawn students), and required a lot of effort from the professors to increase the number of students that can apply for each class and open more options at different times. In 2025, OAC has 4 classes with different professors to handle all the new students and the one that have failed the course in the past, and the professor would like to give the students more options if possible, to handle the demand.

This scenario led, in 2015, to the creation of a new course, Introduction to Computer Systems (CIC0003) (ISC), for the first semester. The goal of ISC was to introduce students to the historical context of computing and hardware development, reinforce mathematical concepts relevant to circuit studies, and present machine language on processor architectures, anticipating topics from Logic Circuits (CIC0229) (CL), Logic Circuits Laboratory (CIC0231) (LCL), and OAC.

Since then, as an unintended consequence, students have been exposed to two different programming languages during their first semester, affecting the experience of the "first programming course" and, in effect, possibly affecting the students' confidence in their ability to code and, with the possibility of failing a course, increasing the risk of dropping out of the Computer Science program completely, as seen in [2].

Furthermore, despite this initiative, the failure rate in OAC remained high, and the newly created ISC course also experienced failure rates above 40%. And while a lot of changes have been implemented around the programming courses focusing on the methodology and student experience, the ISC course hasn't seen the same improvements.

The high failure rates in ISC and OAC courses contribute to student attrition and represent a significant challenge to the Computer Science program and the Department as a whole, requiring more professors and classes to be available to handle all the repeated and new students that are required to take the OAC course, and in the worst case scenario, blocking the student from going forward with the program, from lack of prerequisites to take more advanced classes. These challenges stem from a combination of factors, including the abstract nature of Assembly language, the steep learning curve for introductory programming concepts, and a lack of learning materials and activities directly related to game development, which directly impact the success of the students in the final project.

This research aims to address the high failure rates in the ISC course by implementing a novel pedagogical approach that integrates Problem Based Learning (PBL) with game development. This approach seeks to directly address the aforementioned challenges by helping students have references, in Assembly language, of the main game development concepts they will need during the final project development, like game loops, handling graphical interfaces and Keyboard Input/Output; similar to the changes made to the APC and ED courses.

This research aims to evaluate the effectiveness of a novel teaching methodology in improving student performance and increasing engagement in the ISC course. Specifically, we hypothesize that students in the semesters where the PBL-based game development activities were applied will demonstrate improved problem-solving skills and achieve higher grades on the final project compared to students on the other semesters.

To achieve these objectives, we created classes focused on game development that could help students, no matter the final project; also translated some material that was only available in English; and recorded video classes based on solving common problems that may arise during development.

II. Introduction to Computer Systems (ISC)

Analyzing further the ISC syllabus, we have the following equation for the course grade in 2025. The mean grade on weekly online tests, denoted as M_T, is calculated as:

$$M_T = \frac{1}{16}\left(\sum_{i=0}^{15} T_i\right),$$

where T_i represents the grade on the i-th weekly test.

The minigames grade, M_G, is calculated from optional online exercises available for the students, that can add up to one extra point in the final grade, by the following:

$$M_G = \frac{1}{6}\left(\sum_{i=1}^{6} G_i\right),$$

where G_i represents the grade on the i-th minigame.

The final grade, M_F, is then calculated as:

$$M_F = \frac{E_1 + E_2 + E_3 + \text{Game} + M_T}{5} + M_G,$$

where E_1, E_2, and E_3 correspond to the course's exams, and "Game" corresponds to the course's final project to develop a game using Assembly Language.

The course's final grade is evaluated using the score ranges seen in Table I.

TABLE I
Grade definitions and corresponding score ranges.

Grade	Score Range
SS (Superior)	9.0 to 10.0
MS (Medium Superior)	7.0 to 8.9
MM (Medium)	5.0 to 6.9
MI (Medium Inferior)	3.0 to 4.9
II (Inferior)	0.1 to 2.9
SR (No Performance)	Zero or attendance < 75%

Although there have been changes since 2015, the Game project is still a relevant part of the course's final grade that could steer the grade in either direction, with a similar impact as one of the exams.

Every semester a new game, usually from the arcade era, is selected for the students to create their version of it using the assembly language and the RISC-V Assembler and Runtime Simulator https://github.com/TheThirdOne/rars (RARS).

The game project is presented during the last class of the semester, students must document the project through a 4-page scientific article, according to the provided model, with the sections: Summary, Introduction, Methodology, Results Obtained, Conclusion and Bibliographic References; and prepare a presentation of no more than 10 minutes about the development, techniques used, difficulties faced, and a demonstration of the game, with the allowed and recommended use of cheats to advance to the next stages to allow progressing as much as possible in the 10-minute time limit.

During the presentation, a group of peers will be grading the game using the following criterion presented, as exemplified in Table II.

TABLE II
Evaluation criteria and weights for the game presentation.

Criterion	Weight
Graphical interface (Bitmap Display, 320×240, 8 bits/pixel)	1.0
Keyboard interface (Keyboard and Display MMIO simulator)	1.0
MIDI audio interface (examples of ecalls: 30, 31, 32, 33)	1.0
At least 2 levels with different layouts (map can be static)	1.0
Player animation and movement	1.0
Player ability to destroy parts of the map	0.5
Player collision with obstacles and boundaries	0.5
Power-up mechanics	0.5
Victory condition	0.5
At least 2 different enemy types	1.0
Enemy animation and movement	0.25
HUD (heads-up display) showing updated game information	0.25
Music and sound effects	0.5
Documentation	1.0

The specific evaluation criteria may vary each semester according to the selected game, but notice how if your group (1) uses the graphical, keyboard and MIDI audio interfaces (2) write a 4-page article documenting the development process and (3) be present on the presentation day, you already scored

4 out of 10 points, without even presenting anything related to the specific game chosen for the semester. Furthermore, most of the points are associated with generic enough criteria that allows students to get points for their creativity and focusing on a minimum viable product, rather than a perfectly playable game.

III. Related Work

[3] used data mining techniques to predict student dropout in an introductory programming course. The researchers identified key factors contributing to dropout based on exam and homework data.

[4] correlates student dropout to some key factors like poor project management skills, poorly designed labs and poor problem-solving abilities.

[5] discusses the main aspects related to segmenting educational videos approaches, the importance of the ability to control the playback (forward, rewind, pause) in providing learners with autonomy and control over their learning pace; and how dividing video lectures into logical and semantically meaningful units facilitates reuse and enhances the learning experience.

[6] reports and evaluates teaching based on Project-Based Learning towards an undergraduate subject of Applied Health Informatics. The study found that students reported a positive impact of the discipline on their training and improvement in computational thinking abilities, combining Problem-Based Learning (PBL) with video-based learning has the potential to create a more engaging and effective learning environment.

[7] and [8] describe the new peer-mentoring program for the CS1 course (APC), additional office hours on Saturdays, one-on-one scheduled office hours with peer mentors, active tracking of student engagement to identify students who may need help, and review sessions on Saturdays.

IV. Methodology

This research aims to address the high dropout and failure rates in ISC by applying techniques inspired by the APC program [7], [8] to create game development lectures and materials in Portuguese.

A. In-Person Sessions

The ISC course already has graduate students as tutors to answers questions brought by the students, similar to office hours; but, in the new program, there will be tutoring hours presenting the assembly RISC-V language to the students from a game development point-of-view, instead of the hardware perspective shown in the traditional course. The in-person sessions should emphasize the practical application of the content, aiming to increase students' comfort level with assembly and reduce uncertainty regarding their ability to deliver the final project.

The lectures were created by selecting the most frequent concepts of game development needed to complete the final project, content extracted from the book "The RISC-V Reader: An Open Architecture Atlas" by David Patterson and Andrew Waterman [9] and the RARS wiki.

In the first lecture the goals are to explain the difference between Reduced Instruction Set Computer (RISC) and Complex Instruction Set Computer (CISC), how to use labels, what are assembler directives, how to store data in assembly (.data memory, registers and immediate values), what is the register bank and how its organized, how commands (.text) and data (.data) are separated, pseudo-instructions, environment calls (ecall); ending with some practice exercises.

The second lecture is focused on students understanding how to apply equivalent syntax from higher level languages concepts they were familiarized from APC classes: conditionals, loops and functions, the content of this lecture should be presented by comparing both the higher level language code and the assembly code, presenting a collection of tiny programs that each demonstrate a particular programming concept; ending with practice exercises.

The third and final lecture is focused on the advanced concepts about the assembly language and the RARS tooling available, in this lecture the concepts of stack pointer, macros, constants and splitting code in different files are introduced to help students document and organize the code better to work efficiently; also, in this lecture is explained how to use the graphical, keyboard and MIDI audio interfaces.

The sessions were optional and structured to be 2-hours long, at most, held in classrooms at the Pavilhão João Calmon (PJC), before the ISC classes.

Teaching content in Portuguese, translated from the original sources available in English and extracting content from OAC book that, although available in Portuguese, could be overwhelming for first semester students; and presenting topics that may be used regardless of the specific game for the final project, allows students to engage with the process of solving problems in a similar context to the final project presenting topics from a peer graduate student perspective [4].

This builds upon findings from [2], since empowering students with the knowledge and the confidence to deliver the final project can significantly increase the final course grade, as seen in the M_F equation for the ISC final grade, and a higher course grade means the students will have a higher Academic Performance Index (IRA) by the end of their first semester; this research will analyze the data from the ISC course grades between 2017-1, the first semester where students had a final project, and the last semester of 2024, the most recent data available; this will highlight the scenario before, during and after the intervention on grades from the final project and the whole course grade, similar to the analysis made in [3].

The materials to be used during the sessions include slides, available at https://github.com/thlmenezes/aulas/tree/master/RISC-V, and the RARS development environment. The slides were redesigned and revised by the course professor, incorporating more code examples, animations, and parallels with other programming languages compared to the original material used in the mandatory course lectures.

It is important to note that in 2019-1, the in-person sessions were applied concurrently with the 3rd module of the course. However, in the following semester, the Assembly content in ISC was moved from the 3rd to the 2nd module, while maintaining the same final project deadline. This change provided students from the following semesters with one month extra time to study the language, develop the final project, and consume the content of the supplementary sessions, considering that the slides used in semester 2019-1 were made available to students from the beginning of semester 2019-2 and forward.

B. *Video-Lessons*

In 2020, with the transition from in-person activities to the remote environment, the *In-Person Sessions* also had to adapt, creating the video-lessons available at https://bit.ly/thlmenezesYT.

Following similar guidelines to video segmenting from [5], the video lectures were segmented in multiple videos organized in playlists, allowing students to quickly select videos with challenges more relevant to the topics they want to review. And applying a similar approach to [6], each video has a clear problem, feature or goal to be achieved by the end of the video, like building a Menu or interacting with the Keyboard; and was part of a larger project, Hangman or *Mario Bros*.

The content covered in the video-lessons was how to download RARS and where to access the slides of the in-person sessions, introduce the Hangman project by analyzing the Minimum Viable Product (MVP), how to build a game menu, how to show the hangman hidden word, how to reveal the correct guesses and how to implement the winning/losing conditions.

The traditional Problem Based Learning (PBL) method consists of seven steps [10]: Clarify unfamiliar terms and expressions in the problem text; List the problems; Analyze and discuss the problems, with explanations based on prior knowledge; Summarize the explanations, systematizing the analysis and hypotheses of the problem; Formulate learning objectives for further study and complementation of the explanations; Seek information and conduct individual study respecting the learning objectives; Integrate the information and re-discuss the case based on the advances in knowledge obtained by the group.

The main goal of the hangman series was to show students the concepts of MVP, how to break down and prioritize the game features, how to implement a game loop, how to manipulate strings and memory, how to listen to user input by pooling. The series began by analyzing the main features that made a software application recognizable as a hangman game (MVP), then proceeding to prioritize each feature and solving each problem individually, adding features incrementally until the feature list was entirely done.

Besides the hangman series, there are 2 videos about a Super Mario bros project using the same framework of starting with an MVP, listing the key features and prioritizing them. This unfinished series extends the content from the hangman series by showing the users how to replace the internal console input by the Keyboard MMIO tool.

It is important to note that since the videos were published in 2020-1 until the end of 2022-2, the videos were listed in the "Aprender3" environment, where all the course online resources are listed. However, in 2023, the links were hidden from the students, in order to compare the viewership from when it was listed and when it was hidden and see if any relevant impact could be seen.

V. Results

A. *In-Person Sessions*

In the first intervention, there was an increase in the number of students that delivered and presented the final project from 8 to 20 students, Fig. 1 and Fig. 7. The same positive tendency can be seen in the overall percentage of students being approved, grades MM or greater, Fig. 3 and Fig. 9, growing from 50% to 69.23%; also the number of failed students, grades MI or lower, dropped from 16.67% to 2.56% and the dropout rates were reduced, from 26.19% to 20.51%.

It is also notable that although the number of projects delivered increased substantially, the project grade did not fluctuate noticeably, maintaining close to 6.5 mean grade among the submissions, as per Fig. 2 and Fig. 8.

Also, there was a shift between the grade distribution overall, if we compare the data before and after the intervention, we can see an increase specially in the SS and MS grades, see Table IV and Fig. 10.

During the second intervention, during semester 2019-2, the same pattern could be seen in the data: reduced percentage of student dropout from 20.51% to 12.20%, sustained percentage of approved students with less than 1% variation, slightly increase percentage of failed students from 2.56% to 4.88% as seen in Fig. 4 and Table IV.

In OAC, as per Fig. 11, there was an increase in number of approved students comparing numbers from 2019, 2020, 2021 and 2022; accompanied by a decrease of failed students overall and an increase of dropouts, with a peak in 2021-2.

B. *Video-Lessons*

The channel proved itself a very effective tool to not only in maintaining the research in times of social distancing, but creating a strong engagement of views and a greater sense of participation.

The RISC-V videos accumulated, since their debut, 2.830 views, the videos have spikes in views according to the UnB student calendar, notable by the view spikes, coinciding with the dates of the ISC assembly exam and final project due dates, see Fig. 12.

It is possible to extract, from the views, the themes that most students have interacted, being the 3 most relevant subjects: memory manipulation, menu logic, interaction with keyboard mapped in memory, see Table III. The introduction videos have the most views from both series, hangman and mario bros, and the viewership progressively reduces as the subjects grow more complex and advanced.

TABLE III
YOUTUBE VIEWERSHIP BY VIDEO

Video	Number of Views
RISC-V - Aula 0	585
Jogo da Forca - Introdução	470
Jogo da Forca - Menu	349
Mario Bros (side story) - Como Usar o Teclado?	273
Jogo da Forca - Palavra Oculta	258
Mario Bros - Introdução	212
Jogo da Forca - Menu pt.2	190
Jogo da Forca - Revelando Acertos	174
Jogo da Forca - Tratando Inputs	118
Jogo da Forca - Menu pt.3 + Desafio	80
Jogo da Forca - Vitória ou Morte	68
Jogo da Forca - Separando Arquivos	53

The students also have access to the source code created during the videos so they can choose if they want to watch the video or read the code to see how each challenge was solved.

Analyzing the course grades in the 2020-1, we have no relevant change in the percentage of approved students but the percentage of students that never showed to any classes and the failure were reduced to 0%, see Fig. 5, and the number of withdraws went from 0% to 21.05%, see Fig. 6.

During the second remote semester, the percentage of approved students went from 68.42% to 53.66% and withdraws increased from 21.05% to 39.02%, see Fig. 6. The students who reached the end of the course during those remote semesters had a similar performance, as seen in Table IV.

TABLE IV
DISTRIBUTION OF GRADES BY SEMESTER.

Semester	SS	MS	MM	MI	II	SR
2015-2	3	9	15	6	5	6
2016-1	2	5	19	13	9	4
2016-2	2	6	13	5	7	12
2017-1	1	5	12	8	13	4
2017-2	2	6	11	2	10	7
2018-1	1	6	9	4	9	7
2018-2	1	9	11	4	8	9
2019-1	7	12	8	3	5	4
2019-2	7	11	10	1	3	9
2020-1	6	9	11	0	4	0
2020-2	5	12	6	1	1	0
2021-1	8	13	10	1	0	11
2021-2	4	11	5	4	9	7
2022-1	6	15	9	2	2	5
2022-2	4	15	7	5	1	8
2023-1	12	15	7	0	1	8
2023-2	12	12	10	3	4	2
2024-1	20	9	3	2	0	6
2024-2	16	10	6	0	3	7

C. OAC 2025-1

In 2025-1, there was an OAC class with 40 enrolled students being 38 students from the Computer Science Bachelor program, meaning students that went through ISC unlike the other programs that need the OAC course; from those 38 students, 1 student dropped out from the course and the rest were approved, meaning 97.37% approved students from the total that were enrolled from the Computer Science Bachelor program.

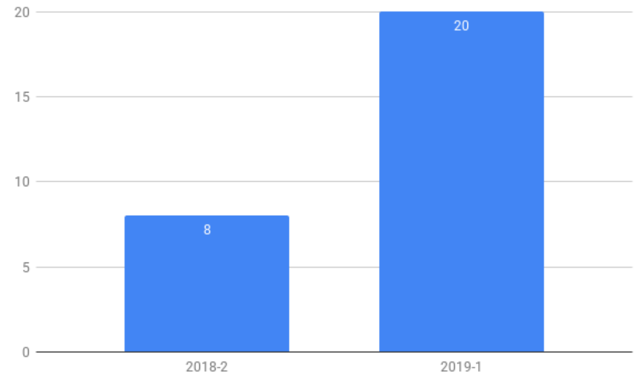

Fig. 1. Graph showing number of students that delivered the final project between semesters 2018-2 and 2019-1

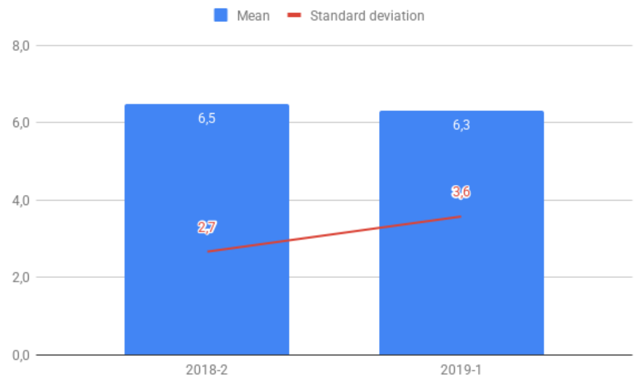

Fig. 2. Graph showing final project grades between semesters 2018-2 and 2019-1

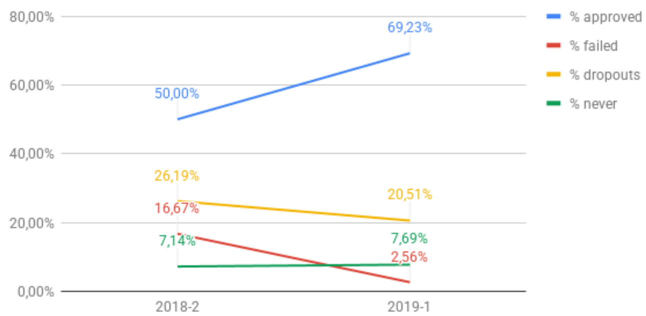

Fig. 3. Graph showing percentages of students approved, failed, dropouts and never for the ISC course between semesters 2018-2 and 2019-1

VI. CONCLUSION

This research investigated the impact of supplementary learning interventions on student performance in the Introduction to Computer Systems (CIC0003) (ISC) course at CIC UnB

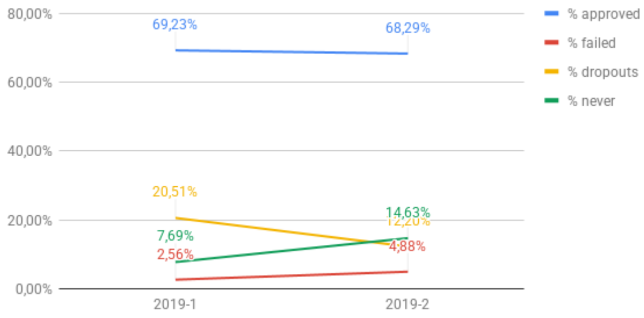

Fig. 4. Graph showing percentages of students approved, failed, dropouts and never for the ISC course between semesters 2019-1 and 2019-2

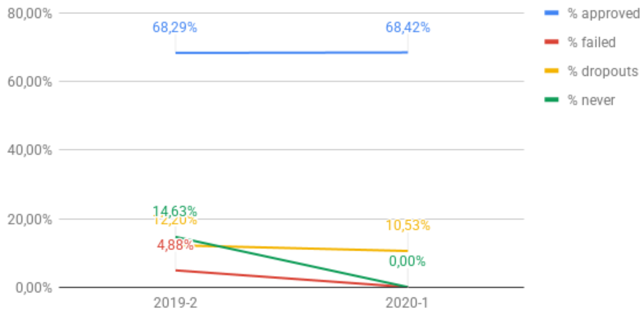

Fig. 5. Graph showing percentages of students approved, failed, dropouts and never for the ISC course between semesters 2019-2 and 2020-1

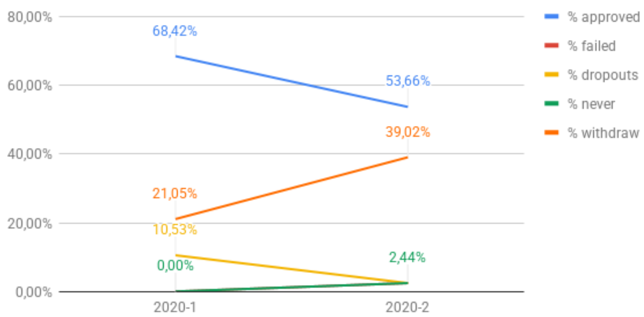

Fig. 6. Graph showing percentages of students approved, failed, dropouts and never for the ISC course between semesters 2020-1 and 2020-2

Brazil, focusing on Assembly language programming in the context of the final project. The interventions included optional in-person sessions and video-lessons designed to reinforce key concepts, provide practical problem-solving experience, and foster a sense of community among students.

The results indicate a noticeable and consistent improvement in course grades and an increase in the average number of projects submitted, suggesting that students felt more confident in their ability to complete the final assignment. The in-person sessions provided a valuable opportunity for students to engage with the material in a collaborative and supportive environment, while the video-lessons offered a flexible and

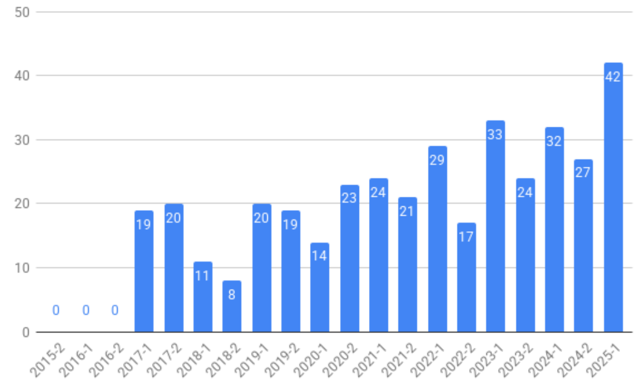

Fig. 7. Graph showing number of students that delivered the final project between semesters 2015-2 and 2025-1

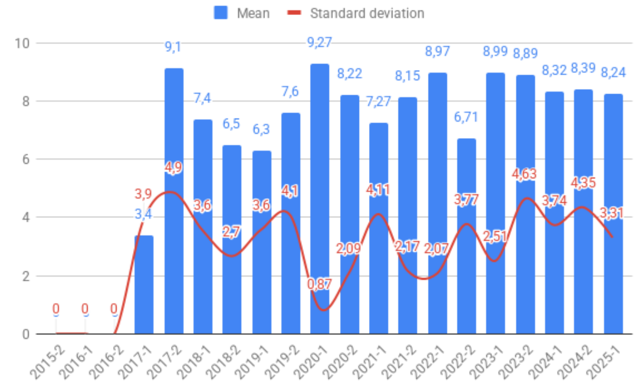

Fig. 8. Graph showing final project grades between semesters 2015-2 and 2025-1

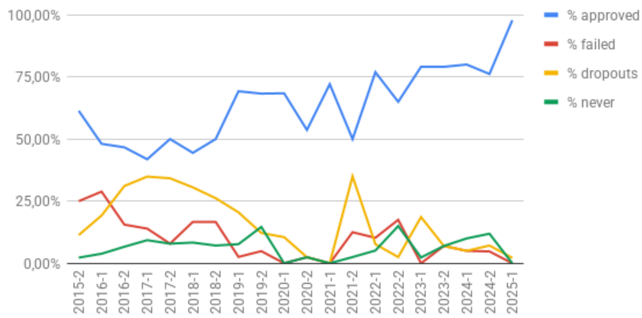

Fig. 9. Graph showing percentages of students approved, failed, dropouts and never for the ISC course between semesters 2015-2 and 2025-1

accessible learning resource that students could use to review concepts and work through problems at their own pace.

The YouTube channel was well-received by the students, demonstrating its effectiveness as a tool for engaging students with the course material. A particularly interesting finding was the increased engagement observed in the semester when the content was being recorded (2020-1) and by sharing previews

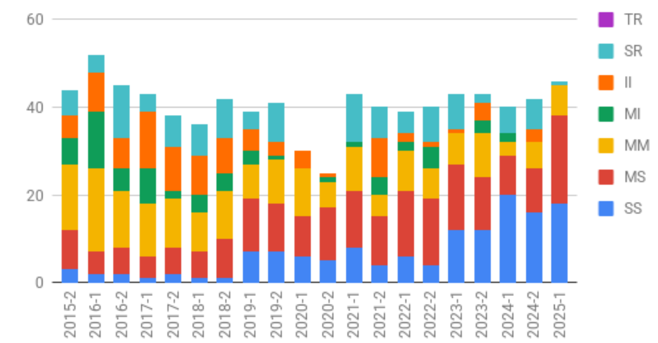

Fig. 10. Graph showing final course grade between semesters 2015-2 and 2025-1

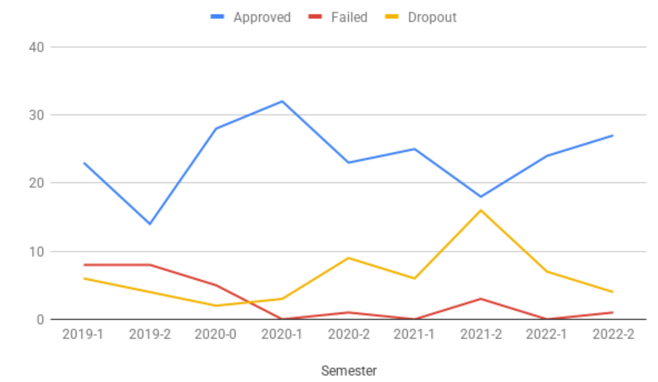

Fig. 11. Graph showing OAC number of approved, failed and dropout students between semesters 2019-1 and 2022-2 from professor Lamar classes

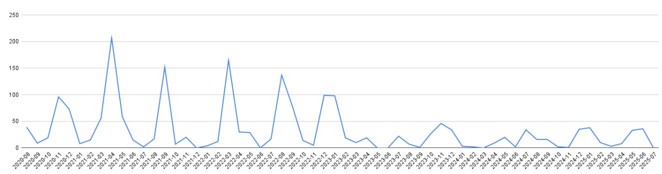

Fig. 12. Graph showing RISC-V playlist monthly viewership between dec/2020 and jul/2025

of editing progress, updates on projects, and discussing relevant topics with viewers/students using Microsoft Teams. The students were interacting with threads and leaving comments and, as expected, the channel experienced a higher volume of traffic in the weeks leading up to the end-of-semester deadlines (final project due date), indicating that students were using the video-lessons to prepare for the final project.

These findings suggest that a combination of in-person and online learning interventions can be an effective strategy for improving student performance in introductory computer science courses, particularly those that involve challenging topics such as, but not limited to, the Assembly programming language. The key to success was to create a supportive learning environment and provide students with flexible and accessible learning resources.

While this research provides valuable insights into the impact of supplementary learning interventions, it is important to acknowledge its limitations. While the interventions demonstrably reduced failure rates in ISC, there was a slight increase in failure rates compared to the periods immediately following the intervention. However, these rates did not return to pre-intervention levels, indicating a sustained positive impact.

Also, another factor to consider is that during the 2020 pandemic with remote classes it had an external interference in the class, it's hard to maintain the same numbers even tho having a lot of video and content available for the students and this class has its own specific behavior due to the changes we made to adapt into this scenario, those were one of the key reason to not continue this in person due to the pandemic impact.

The impact on the OAC course could not be measured precisely by analyzing the historic data in this research, due to the challenge to analyze the impact of the intervention, specially the in-person sessions, one year later and expecting the students from the Computer Science Bachelor program that participated in the research to be studying OAC in that window and the need to analyze multiple class data from multiple professors using data from classes that had to be tailored to handle students from other programs as well that can share the same class.

But, we can have some insights looking to the data we acquired from the 2025-1 OAC classes that had 95% of the enrolled students being form the Computer Science Bachelor program and, analyzing those students, the class reached more than 97% approved rate; although that doesn't mean there is any direct correlation with the interventions or its long term effects, it's the best this research could find to isolate the impacts.

[2] demonstrated a correlation between a higher Academic Performance Index (IRA) and a lower probability of student dropout in CIC. This research strengthens the belief that success in early courses, such as ISC, significantly contributes to student confidence and academic persistence, that affects not only the current course but could have a lasting impact on the whole program itself, as reinforced by the data from the OAC in 2025-1, that shows students with astonishing performance in what is still considered a high failure rate course.

Another hard to measure impact of this research is the learning material created by other students that participated in this project, like the content creation from Victor Lisboa in this youtube playlist and the curation of learning materials and tools in the LAMAR repository. This demonstrates that what started as a help to the first semester grades has become a mindset and culture that help more students get invested, making all the content created more effective every year from the project start.

Since this research has started, there were new tools available that were made by other students and is available to the community help to generate new tools to simplify to process of developing content for ISC class or to help students create even more complex games in a shorter time frame, for instance the Midi2RiscV tool that "converts the data from a mid/midi file and creates a .data file containing the list of notes each track and voice has, in a way that a RISC-V assembly code can interpret and play" and Fast Pretty Good RISC-V Assembly Rendering System (FPGRARS), available at https://leoriether.github.io/FPGRARS, that is "a RISC-V assembly simulator with a graphics display window and keyboard input, similar to RARS, but 200 times faster"; it also important to notice and point that new and recurrent students, that have succeeded in the ISC class tend to continue to help as tutors, creating content and helping new students, demonstrating a cycle of success to keep with the study goals in the long term.

Future research should explore the long-term impact of these interventions and build upon the successful strategies employed in other programming courses within the department.

One promising avenue is to enhance the development experience by creating a more streamlined and accessible development environment. This could include a web-based or mobile environment for code execution and debugging, similar to a web/mobile version of the FPGRARS project, reducing the need for local installations and improving accessibility across different devices.

Another key area for future work is expanding and diversifying content by developing a wider range of video lessons and supplementary materials, addressing common challenges faced by students in Assembly language programming and incorporating different pedagogical approaches. This includes new ways to use video lectures, and the development with the team itself for more and more classes. Applying validated techniques from the APC and ED courses, such as peer mentoring, collaborative projects, and personalized feedback, to the ISC context could also be highly beneficial.

Extending support to OAC is another potential direction, recognizing that this would serve a broader audience beyond just Computer Science Bachelor students, where creation assembly exercises that are specific to the OAC class can be tested, where in person validation is required and it has the high potential to create more advanced students. The results with the 2025-1 OAC class can bring new insights, there are always new aspects that can come to help with this, that we are more than willing to do more and improve student grades, and we would like to take some notes about in which ways they need the most to continue.

Finally, future research should investigate the potential of training an AI model to support students during development, providing personalized guidance, answering questions, and acting as a virtual tutor or pair programming partner, an "AI RISC-V Tutor", where the first steps have been made and as the machine are working there could be new students helping other, creating a stronger community and a more scalable study for every student.

REFERENCES

[1] M. Holanda, M. Mandelli, E. Ishikawa, and D. Da Silva, “Introdução ao ambiente universitário na computação: Uma disciplina para o acolhimento dos estudantes no departamento de ciência da computação da universidade de brasília,” *Revista Brasileira de Informática na Educação*, vol. 30, p. 494–518, out. 2022. [Online]. Available: https://journals-sol.sbc.org.br/index.php/rbie/article/view/2451

[2] G. P. Garcia and M. Holanda, “Análise preditiva em relação a evasão do curso de ciência da computação na universidade de brasília,” University of Brasilia, Tech. Rep., 2024. [Online]. Available: https://bdm.unb.br/handle/10483/40796

[3] D. dos Santos Silva, S. C. Tamayo, M. Pessoa, F. Pires, D. B. F. Oliveira, E. H. T. de Oliveira, and L. S. G. de Carvalho, “Minerando dados de um juiz on-line para prever a evasão de estudantes em disciplinas introdutórias de programação,” *Anais do Simpósio Brasileiro de Informática na Educação*, pp. 1343–1352, 11 2020. [Online]. Available: https://sol.sbc.org.br/index.php/sbie/article/view/12890

[4] T. Beaubouef and J. Mason, “Why the high attrition rate for computer science students,” *ACM SIGCSE Bulletin*, vol. 37, pp. 103–106, 6 2005.

[5] E. Barrére, J. Souza, and E. R. Soares, “Framework para segmentação temporal de vídeos educacionais,” *Anais do Simpósio Brasileiro de Informática na Educação*, pp. 972–981, 11 2020. [Online]. Available: https://sol.sbc.org.br/index.php/sbie/article/view/12853

[6] L. C. Santos, D. F. Neves, and F. dos Santos Menezes, “An assessment of project-based learning towards applied health informatics teaching in an undergraduate class,” *Anais do Simpósio Brasileiro de Informática na Educação*, pp. 202–211, 11 2020. [Online]. Available: https://sol.sbc.org.br/index.php/sbie/article/view/12776

[7] M. Holanda, C. Castanho, I. Bandeira, and D. Silva, “Relato de experiência da monitoria da disciplina primeira linguagem de programação do departamento de ciência da computação da universidade de brasília,” in *Anais do XXX Workshop sobre Educação em Computação*. Porto Alegre, RS, Brasil: SBC, 2022, pp. 13–25. [Online]. Available: https://sol.sbc.org.br/index.php/wei/article/view/20815

[8] M. Holanda, I. N. Bandeira, C. D. Castanho, A. B. de Sousa, and D. da Silva, “Programa de monitoria da disciplina de programação introdutória na universidade de brasília,” *Revista Brasileira de Informática na Educação*, vol. 32, pp. 157–180, 3 2024.

[9] D. Patterson and A. Waterman, *The RISC-V Reader: an open architecture Atlas*. Strawberry Canyon, 2017.

[10] M. Kubrusly, R. A. Coelho, K. L. Augusto, A. Aires, P. Junior, D. Costa, O. Santos, and C. M. C. D. Oliveira, “Percepção docente sobre a aprendizagem baseada em problemas no ensino remoto durante a pandemia covid-19,” *Research, Society and Development*, vol. 10, p. e53510515280, 5 2021, definição de PBL. [Online]. Available: https://rsdjournal.org/index.php/rsd/article/view/15280