

Análise da aplicação de metodologias ativas em disciplina de Lógica de Programação

Havana Diogo Alves ¹

¹Instituto Federal de Pernambuco – Campus Jaboatão dos Guararapes
Estr. de Bulhões - Bulhões, Jaboatão dos Guararapes - PE, 54080-000

havana.alves@jaboatao.ifpe.edu.br

Abstract. *This article analyzes the adoption of active methodologies as a teaching-learning strategy in the Logic of Programming discipline, aiming to improve indicators of failure and evasion in this subject. We present the scenario before and after the application of Problem-Based Learning, Team-Based Learning and Flipped Classroom methodologies. As a result, after six semesters of using these active methodologies, the result was satisfactory, reducing the rate of failures.*

Resumo. *Este artigo traz a análise da adoção de metodologias ativas como estratégia de ensino-aprendizagem em disciplina de Lógica de Programação, visando melhorar indicadores de reprovação e evasão desta matéria. Apresentamos o cenário anterior e o posterior a aplicação das metodologias Aprendizado Baseado em Problemas, Aprendizado Baseado em Times e Sala de Aula Invertida. Como resultado, após seis semestres de utilização destas metodologias ativas, o resultado foi satisfatório, diminuindo o índice de reprovações por falta e por nota.*

1. Introdução

O desafio de ensinar e aprender lógica de programação faz parte do dia a dia de professores e alunos dos cursos de Computação dos níveis técnico e superior, dificuldade que vem mantendo os índices de retenção nos períodos iniciais desses cursos em níveis alarmantes. [Rabelo et al. 2018] constataram que 43% dos alunos de computação, brasileiros, falharam em disciplinas introdutórias de programação, já incluindo um número alarmante de 25% de abandono da disciplina. O estudo de [Giraffa and Móra 2013] mostra que os alunos desistem dos cursos logo no primeiro ano do ensino superior de computação, e as disciplinas causadoras desta desistência são aquelas associadas ao ensino de Cálculo e de Programação. Hábitos de estudo pouco disciplinados e centrados em memorização, conhecimentos prévios desestruturados, principalmente nos domínios matemáticos e lógicos, abordagens pouco motivadoras, conteúdos pouco relacionados ao cotidiano dos sujeitos, dificuldades na compreensão do enunciado dos problemas e forte carga de conceitos abstratos estão entre os motivos citados por [Borges 2000, Giraffa and Móra 2013] para o insucesso nestas disciplinas.

Neste artigo estão descritas as mudanças realizadas na metodologia de ensino da disciplina de Lógica de Programação e Estrutura de Dados (LPED) do curso Técnico em Informática para Internet do Instituto Federal de Pernambuco, Campus Jaboatão dos Guararapes, onde o percentual médio de falhas (reprovações e evasões) desde as primeiras

turmas do curso, no 2º semestre de 2014 (entrada 2014.2) até as turmas do 2º semestre de 2017 (entrada 2017.2), passava dos 66% dos matriculados. Por ser disciplina do 1º período do curso e pré-requisito para várias outras dos períodos subsequentes, o alto índice de reprovação neste componente curricular resultava em uma alta taxa de retenção e evasão. Desta forma, várias foram as tentativas de diminuir o alto índice de reprovação em LPED, iniciando de simples mudanças nas linguagens de programação utilizadas nas aulas até a culminância da aplicação de metodologias ativas como *Problem Based Learning* (PBL) aliada à linguagem *Scratch* e uso de *Team Based Learning* (TBL).

2. Ensino de Linguagem de Programação: cenários iniciais

Desde a implantação do curso, em 2014.2 até o semestre letivo de 2017.2, a metodologia de ensino da disciplina de LPED seguia o modelo tradicional expositivo, geralmente com o seguinte roteiro: (1) explanação do professor, (2) execução de exercícios em sala, (3) listas de exercícios para a prática em casa; (4) atividade avaliativa escrita ou utilizando computador ao final de cada bimestre. Era possível perceber alguns problemas resultantes deste tipo de metodologia:

- dispersão de alguns alunos durante as aulas;
- desmotivação de boa parte dos alunos em resolver os exercícios em sala de aula;
- as listas de exercícios, que têm como objetivo estimular os estudantes a praticarem a resolução de problemas fora da sala de aula eram tratadas de duas formas: (1) se a lista não tivesse alguma pontuação atribuída, poucos eram os alunos a tentarem resolvê-la ou (2) se houvesse uma pontuação atrelada, era identificado um alto índice de plágio nas resoluções.

Durante este tempo, foram realizadas várias tentativas para mitigar as falhas no aprendizado, indo desde a mudança de professores até a escolha de diferentes linguagens de programação. A Tabela 1 exibe a configuração das aulas em cada um dos semestres letivos

Tabela 1. Mudanças realizadas no decorrer dos semestres letivos

Semestre	2014.2	2015.1	2015.2	2016.1	2016.2	2017.1	2017.2
Professor	P1	P2	P3	P4	P4	P4	P4
Ling. Unid. 1	Java	Java	Java	JS	Portugol	Portugol	Portugol
Ling. Unid. 2	Java	Java	Java	JS	JS	JS	JS
Aula	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.	Exp.

Exp. - Aula Expositiva. JS - JavaScript

vos de 2014.2 a 2017.2. Importante salientar que em todos estes semestres tivemos turmas nos turnos vespertino e noturno. Nas linhas indicadas como **Ling. Unid. 1** e **Ling. Unid. 2** estão as linguagens de programação utilizadas, respectivamente, na 1ª e na 2ª unidades de cada semestre letivo. Observa-se que todas as variáveis mudaram exceto a metodologia de ensino-aprendizagem, que continuou sendo expositiva, com atuação passiva dos estudantes. Tendo em mãos esta análise, iniciaram-se pesquisas sobre formas de ensinar que motivassem o aprendizado dos alunos.

As pesquisas mostraram que abordagens baseadas em metodologias ativas têm sido pesquisadas e indicadas como formas de proporcionar maior interesse do estudante sobre o assunto a ser estudado, tendo, em geral, como premissas:

- motivar o porquê de se estar estudando determinado conteúdo;
- tornar o estudante protagonista de seu aprendizado;
- estimular a colaboração entre colegas de grupo.

Assim, a partir de 2018.1 foram utilizados de forma isolada ou combinada três das várias metodologias ativas existentes: o Aprendizado Baseado em Problemas (ABP)[da Conceição and Guedes 2021], a Sala de Aula Invertida (*flipped classroom*) [Berrett 2012] e o Aprendizado Baseado em Equipes (ABE) [Michaelsen and Sweet 2012].

3. Aplicação das Metodologias Ativas

Em 2018.1 e 2018.2 a metodologia escolhida foi a ABP, a qual vem sendo utilizada por professores na área de informática [Yongmin 2020, Mourão 2017, Coelho 2016, de A. Oliveira et al. 2012], devido à capacidade de desenvolver as competências necessárias a um profissional da área, como resolução de problemas, trabalho colaborativo e busca auto-gerida do conhecimento necessário para se alcançar os objetivos [Kotovsky 2003]. [Saad and Zainudin 2022] encontraram diversas publicações, de 2017 a 2021, descrevendo implementações de ABP como forma eficaz de ensinar e aprender pensamento computacional. Assim, o processo de aprendizado é iniciado quando um problema é proposto pelo professor, que também disponibiliza os materiais instrucionais que irão servir como base teórica inicial para o desenvolvimento da solução. Neste processo, o professor atua como facilitador, dirimindo dúvidas, guiando e estimulando as equipes. É fundamental que os problemas criados para se trabalhar com ABP sejam bem estruturados, estimulando *brainstorming* e formulação do problema [Savery 2006].

3.1. Aplicando Aprendizado Baseado em Problemas aliado ao *Scratch*

Aliada à metodologia ABP, a linguagem *Scratch* foi escolhida para o desenvolvimento das soluções propostas na primeira unidade de cada semestre letivo por permitir focar na semântica da solução que deveria ser desenvolvida, e também por fornecer a possibilidade de criar animações completas mesmo utilizando poucos blocos de código ou seja *baby steps*, proporcionando ao estudante um caminho de aprendizado gradual, trabalhando com dois ou três conceitos em cada projeto. A aplicação de ABP-Scratch foi dividida em 6 etapas, conforme descritas a seguir:

1. **Elaboração dos problemas:** Os problemas precisam ser elaborados visando uma sequência pedagógica de complexidade, e de acordo com os itens do conteúdo programático.
2. **Organização dos grupos:** Os alunos são divididos em grupos de no máximo quatro, e cada membro recebe um dos papéis: líder, relator, desenvolvedor ou pesquisador.
3. **Disponibilização do problema e dos materiais instrucionais:** O professor deve explicar o problema a ser trabalhado durante a aula, apresentando os objetivos de aprendizagem e os materiais instrucionais disponibilizados, os quais podem ser vídeos de como utilizar os blocos do Scratch, apostilas e demais fontes de conhecimento. Neste momento é importante deixar claro que o aluno deve buscar materiais complementares. Não obstante, é essencial deixar claro os entregáveis, prazos e meios de submissão.

4. **Apresentação das animações:** Cada grupo apresenta as suas animações a toda a turma e explica o desenvolvimento do código.
5. **Feedback para turma:** Ao término do prazo, o professor deve retomar o problema apresentado, fazendo uma retrospectiva sobre os conceitos trabalhados e o desenvolvimento destes.
6. **Registro da prática:** Para efeito de futuras análises, o professor deverá registrar um relato da prática realizada, a fim de retomar estas anotações e possibilitar uma ação de melhoria na metodologia.

Em 2018.1, os grupos foram formados no primeiro dia de aula, onde os membros foram escolhidos pelos próprios alunos. Já em 2018.2, os grupos foram formados aleatoriamente para cada projeto, o que não agradava muito aos alunos, mas que era necessário para simular ambientes em que os profissionais teriam de trabalhar com pessoas diferentes.

Os problemas propostos foram elaborados de forma a seguirem grau de dificuldade gradual, iniciando com a criação de simples diálogos - onde eram necessários conceitos de algoritmos, instruções e interação com o usuário -, a criação de Jogos de Perguntas e Respostas, onde os objetivos de aprendizagem foram a compreensão dos conceitos de variáveis, operadores aritméticos, lógicos e relacionais e cláusulas condicionais simples, até a criação de jogo de colisão, onde utilizaram-se recursos de laços de repetição. A Figura 2 exibe a imagem de um dos projetos entregues por uma das equipes do turno Vespertino. A avaliação dos alunos foi realizada de forma contínua desde o dia da proposição

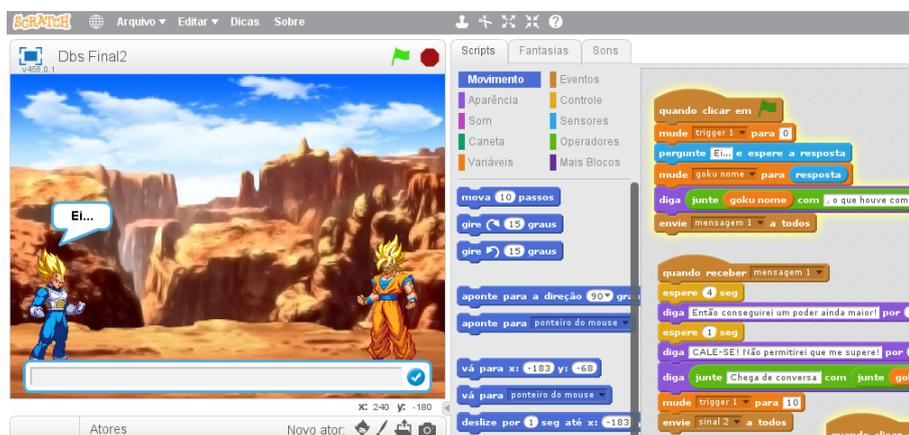


Figura 1. Animação criada por uma das equipes onde acontece um diálogo entre personagens

de cada problema. Atitudes como a não participação nas discussões em grupo, a falta de cumprimento das tarefas atreladas ao papel incumbido e ausências durante as aulas, eram anotadas e um retorno, de forma o mais imediata possível, era dado ao aluno. Para permitir o melhor acompanhamento das equipes em sala, o layout do laboratório teve de ser alterado do formato tradicional, em que as bancadas ficam todas voltadas para o quadro branco, para o formato em “U” [MEC - Ministério Da Educação et al. 2005], pois este layout facilita a movimentação de professores e monitores no ambiente, e permite observar o que os alunos estão vendo ou fazendo nos computadores, permitindo a intervenção quando necessário.

Foi possível perceber, em ambos os semestres, que as turmas do turno vespertino se mostravam mais empolgados em utilizar o Scratch, diferentemente do percebido nas

turmas noturnas. Isto pode ter acontecido pela característica etária de cada turno: à tarde a média de idade dos alunos é de 23 anos e em sua grande maioria não possuem outra atividade além do referido curso, enquanto que à noite temos alunos com média de idade acima dos 26 e que boa parte possuem outras atividades durante o dia e conseqüentemente, requerem um nível maior de objetividade e seriedade.

3.1.1. Aplicando Aprendizado Baseado em Times combinado à Sala de Aula Invertida

Nos semestre letivos de 2019.1, 2019.2, 2022.1 e 2022.2, outra abordagem passou a ser utilizada em ambos os turnos: a combinação de Aprendizado Baseado em Times e Sala de Aula Invertida, que se deu da seguinte forma:

1. Cada turma foi dividida em grupos de no máximo 4 membros;
2. O conteúdo abordado em sala era disponibilizado, no mínimo, uma semana antes em plataforma online, contendo vídeos, textos e listas de problemas referentes ao assunto ministrado;
3. As listas de problemas eram compostas por uma lista para exercícios, que era utilizada para as aulas e para a prática dos estudantes, e por uma lista avaliativa, a qual deveria ser estudada em grupo, para ser executada em data marcada da seguinte forma:
 - (a) Sorteava-se um grupo por vez. O grupo sorteado tinha um dos seus componentes também sorteado e logo após, uma das questões era escolhida aleatoriamente para ser solucionada em computador reservado para tal;
 - (b) Nos casos de o aluno sorteado não resolver o problema, ou no caso de se recusar a resolver, o grupo todo não pontuava para aquela lista. O aluno que faltava também não pontuava.
 - (c) Este procedimento era realizado até que todas os problemas da lista fossem resolvidos.

Ao final de cada unidade letiva, os estudantes que, porventura, não pontuaram em alguma das listas, tinham a possibilidade de recuperar a pontuação referente àquele assunto abordado nos problemas. Esta recuperação consistia em prova prática individual.

O que se quis estimular e o que foi percebido durante a execução desta abordagem foi a cooperação inter e intragrupo. Visto que o plágio era uma constante nas execuções de listas de exercícios no modelo tradicional, o compartilhamento de respostas passou a ser, inclusive, estimulado. O objetivo era que, devido à aleatoriedade dos sorteios de membros dos respectivos grupos e também dos problemas, os estudantes tinham muita necessidade de entenderem a resolução dos problemas e também de ajudar seus colegas neste entendimento. Dessa forma, os alunos com mais habilidade em um grupo ajudavam aqueles que não estavam compreendendo bem a resolução de cada problema. Como também não havia disputa entre os grupos, a cooperação intergrupo era bastante frequente.

Um ou outro aluno mais aplicado reclamou que não achava justa a possibilidade de serem prejudicados por colegas menos interessados. Porém, esses conseguiam recuperar suas notas na prova prática individual.

4. Comparando os resultados de metodologia tradicional *versus* metodologias ativas

Foram analisados os dados de reprovações por nota e de evasões para comparar os resultados das turmas onde se utilizaram aulas expositivas e as turmas onde metodologias ativas foram aplicadas.

Para realizar as análises, retiramos de cada turma os alunos que nunca compareceram às aulas ou que compareceram apenas à primeira semana de aula, na qual costuma-se apenas apresentar a disciplina e realizar cadastro dos estudantes nas plataformas online. Dificilmente, um aluno que apenas frequentou este período de tempo evadiu-se por não se identificar com a metodologia aplicada na disciplina, logo a situação de cada um destes está fora da influência do tipo de aula. Desta forma, foram levados em conta nas análises, apenas os estudantes que frequentaram pelo menos as duas primeiras semanas de aula, pois na segunda semana a metodologia aplicada já é posta em ação.

4.1. Reprovações por nota

Nesta seção comparamos as reprovações por nota entre os semestre letivos em que foram utilizadas Metodologias Ativas e àqueles em que foi executado o tipo de aula tradicional. É possível notar o impacto na proporção de reprovados por nota. Nas turmas da tarde onde as aulas seguiam o modelo de aulas expositivas, a mediana de reprovações por nota foi de pouco mais 27% da turma, com turmas onde mais da metade (até 57%) dos estudantes não obteve nota suficiente para ser aprovado na disciplina. Nas turmas noturnas a mediana ficou em 20% e algumas chegando a 46% de reprovação. Já nas turmas onde foram aplicadas metodologias ativas, em ambos os turnos, a mediana ficou em cerca de 20%, tendo turmas com no máximo cerca de 25%, com um *outlier* de 38% no turno vespertino.

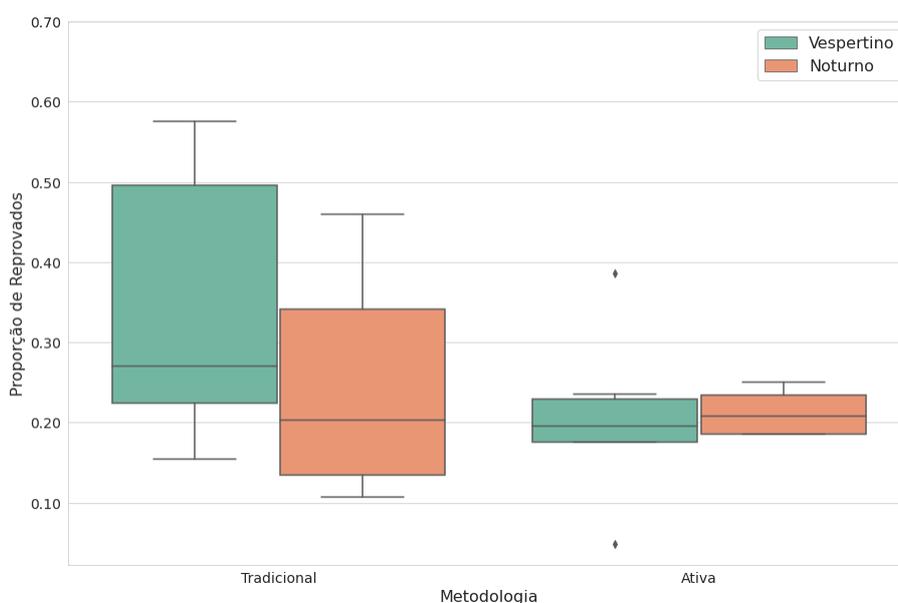


Figura 2. Boxplots das proporções de reprovados por nota nas turmas que utilizam metodologia tradicional *versus* metodologias ativas

4.2. Reprovações por falta

O impacto nas evasões também foi significativo. A Figura 3 mostra os boxplots referentes à quantidade de alunos reprovados por falta nas turmas onde houve aplicação da metodologia tradicional em contraste às que houve aplicação de metodologias ativas. A

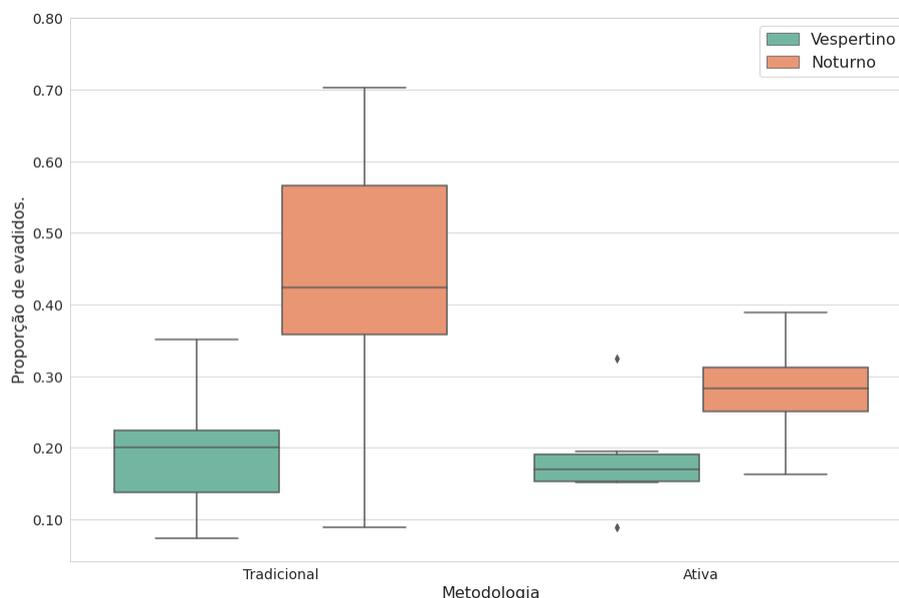


Figura 3. Boxplots das proporções de reprovados por falta nas turmas que utilizaram metodologia tradicional versus metodologias ativas

proporção de estudantes que abandonou a disciplina no turno vespertino apresentou leve diminuição na mediana, passando de 20% para cerca de 17%. Porém, nas turmas noturnas a diferença é bastante significativa, onde houve queda da mediana que passou de cerca de 42% para cerca de 28%, também passando de máximo de 70% para um máximo de menos de 40% de reprovações por falta.

Do ponto de vista comportamental, observou-se forte integração dos alunos que trabalharam com ABP e com o ABE, característica não observada em turmas anteriores. Enquanto até 2017.2 constatava-se uma falta de motivação por parte dos estudantes, inclusive para participar de eventos e festividades promovidos pela instituição, de 2018.1 em diante (excetuando-se no período de aulas totalmente remotas), os alunos passaram a participar mais ativamente das palestras, eventos científicos, *workshops* e festividades que eram promovidos pelo campus.

5. Conclusões

As metodologias ativas vêm se mostrando alternativas eficazes para que uma aprendizagem significativa se torne real, o que se mostrou promissora também no ensino técnico profissionalizante. Fato é que a carga de trabalho tanto dos alunos quanto do(a) professor(a) tende a aumentar com este tipo de metodologia, porém, no final o resultado tem se mostrado satisfatório.

Atualmente, metodologias ativas vêm sendo aplicadas na mesma instituição com alunos do Ensino Médio Integrado, e as observações e resultados de tais aplicações serão relatados em trabalhos futuros.

Referências

- Berrett, D. (2012). How ‘flipping’ the classroom can improve the traditional lecture. *The chronicle of higher education*, 12(19):1–3.
- Borges, M. A. F. (2000). Avaliação de uma metodologia alternativa para a aprendizagem de programação. *Workshop de Educação em Computação - Congresso anual da SBC*.
- Coelho, F. E. S. (2016). Primeiros passos na aprendizagem baseada em problemas. *Congresso Regional sobre Tecnologias na Educação (Ctrl+e)*.
- da Conceição, C. M. and Guedes, A. M. A. (2021). Aprendizagem baseada em problemas aplicada à programação de computadores: Um mapeamento sistemático. *RENOTE*, 18(2):570–580.
- de A. Oliveira, A. M. C., Rodrigues, R. L., and Garcia, V. C. (2012). Um mapeamento sistemático para problem based learning aplicado à ciência da computação. *Anais do Workshop de Informática na Escola - WIE 2012- XXXII Congresso da Sociedade Brasileira de Computação*.
- Giraffa, L. M. M. and Móra, M. C. (2013). Evasão na disciplina de algoritmo e programação: Um estudo a partir dos fatores intervenientes na perspectiva do aluno. *3ª Conferência LatinoAmericana Sobre el Abandono en La Educación - III CLABES*.
- Kotovsky, K. (2003). *Problem Solving-large/small, hard/easy, conscious/nonconscious, problem-space/problem-solver: The issue of dichotomization*, page 374 – 384. Cambridge University Press, UK.
- MEC - Ministério Da Educação, SEED - Secretaria de Educação A Distância, and PROINFO - Programa Nacional de Informática Na Educação (2005). *CARTILHA: Recomendações para a Montagem de Laboratório de Informática nas Escolas. 12 p.* Brasília/DF. Acesso em: 23 de fevereiro de 2019.
- Michaelsen, L. K. and Sweet, M. (2012). Fundamental principles and practices of team-based learning. *Team-based learning for health professions education: A guide to using small groups for improving learning*, pages 9–34.
- Mourão, A. (2017). Uma proposta da eficiência do uso da metodologia ativa baseada em problemas, utilizando dojo de programação, aplicada na disciplina de lógica de programação. *Anais do Workshop de Informática na Escola - WIE 2017- XXXVII Congresso da Sociedade Brasileira de Computação*.
- Rabelo, A., Maia, L. C. G., and Parreiras, F. S. (2018). Performance analysis of computer science students in programming learning. *26º WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI 2018)*, 26(1/2018).
- Saad, A. and Zainudin, S. (2022). A review of project-based learning (pbl) and computational thinking (ct) in teaching and learning. *Learning and Motivation*, 78:101802.
- Savery, J. R. (2006). Overview of problem-based learning: Definitions and distinctions. In *Interdisciplinary Journal of Problem-Based Learning*, volume 1, pages 9–20.
- Yongmin, K. (2020). The effects of pbl-based data science education classes using app inventor on elementary student computational thinking and creativity improvement. *Journal of The Korean Association of Information Education*, 24:551–562.