

Ensino de conceitos básicos de programação usando MIT App Inventor para alunos de escolas públicas: um relato de experiência

Hiero Henrique Barcelos Costa¹, João Stephan Silva Maurício¹,
Augusto de Filippo Cavalini¹, Gabriel Henrique de Souza¹, Gabriel Mattos¹,
Davi Luís de Faria Rocha¹, Júlia Zoffoli Caçador¹,
Robert Gonçalves Vieira de Souza¹, Luiz Gustavo Ferreira Nazareth¹,
Marcelo Ian Rezende Menezes¹, Pedro Henrique Dias Valle¹,
Joventino de Oliveira Campos¹

¹Departamento de Ciência da Computação
Universidade Federal de Juiz de Fora (UFJF)
Caixa Postal 20.010 – 36.036-900 – Juiz de Fora – MG – Brasil

{gabriel.mattos, davi.rocha, julia.zoffoli}@estudante.ufjf.br,
{robert.souza, luiz.nazareth}@estudante.ufjf.br,
{hiero.costa,marcelo.ian}@ice.ufjf.br
{joao.stephan, augustocavalini.filippo}@estudante.ufjf.br
gabriel.souza@engenharia.ufjf.br
{pedrohenrique.valle, joventino.campos}@ufjf.br

Abstract. *An issue that has been increasingly discussed is what skills should be developed to create professionals capable of adapting to this more technological and digital society. In this context, the main skill to be stimulated and developed is certainly computational thinking, as it is applicable and required in any professional field within this new reality. Given this, members of the GET Engenharia Computacional group together with members of the Escola de Games UFJF proposed to teach programming using MIT App Inventor. Therefore, the objective of this article is to report on the results achieved, as well as the problems faced during the experience.*

Resumo. *Uma pauta cada vez mais discutida está relacionada às capacidades que devem ser exercitadas para desenvolver profissionais capazes de se adaptar a essa sociedade mais tecnológica e digital. Nesse contexto, é certo que a principal habilidade a ser estimulada e desenvolvida é o pensamento computacional, por ser aplicável e requisitada em qualquer ramo profissional dentro dessa nova realidade. Visto isso, os integrantes do grupo GET Engenharia Computacional juntamente dos integrantes da Escola de Games UFJF propuseram o ensino de programação utilizando o MIT App Inventor. Dentro desse contexto, o objetivo desse artigo é realizar um relato sobre os resultados alcançados, assim como os problemas enfrentados durante a experiência.*

1. Introdução

O avanço das tecnologias computacionais demandam cada vez mais a presença por profissionais qualificados e capacitados [Manovich 2013]. O mundo está se encaminhando para

uma sociedade cada vez mais tecnológica, guiada por algoritmos, Inteligência Artificiais, processos automatizados e robôs [García-Peñalvo 2018]. Por isso, buscando amenizar o impacto econômico dessa implicação, diversos países na última década vêm tentando integrar em seu sistema educacional o aprendizado em programação e acima de tudo o desenvolvimento do pensamento computacional, que possibilita a criação de novas tecnologias.

O ensino de programação tem apoiado o desenvolvimento do pensamento computacional [Belmar 2022]. Isso acontece pois o pensamento promove uma maior capacidade de se pensar de maneira sistemática [Lye and Koh 2014]. Além disso, o uso de abordagens mais visuais utilizando blocos permite aos estudantes um foco maior na lógica e estrutura do programa a ser escrito [Belmar 2022]. Isso facilita o aprendizado pois permite tirar o foco das mecânicas de escrita do mesmo se utilizando linguagens de programação como Java e C++.

Uma iniciativa do Instituto de Massachusetts (MIT) foi a criação do MIT App Inventor¹, um ambiente online que permite a criação de aplicativos utilizando blocos de construção de código. Esses blocos permitem aos estudantes uma forma mais intuitiva de se visualizar conceitos como operadores lógicos e aritméticos, manipulação de variáveis e estruturas de repetição. Além disso, estudos apontam uma boa receptividade por parte dos estudantes para o aprendizado de programação utilizando MIT App Inventor no ensino médio e superior, como visto em [St. Georgiev 2019].

Neste contexto, o Grupo de Educação Tutorial da Engenharia Computacional (GET-EngComp)², conduziu oficinas para estimular o desenvolvimento do pensamento computacional em estudantes de ensino médio através do MIT App Inventor. Portanto, esse trabalho relata a experiência obtida dessa atividade desenvolvida na Escola Estadual Nyrce Villa Verde Coelho De Magalhães na cidade de Juiz de Fora (MG). Este trabalho foi organizado em seis seções contando com a introdução. A Seção 2 consiste na fundamentação teórica da aplicação feita, na Seção 3 é apresentada a metodologia do trabalho. A Seção 4 apresenta um relato da experiência seguido das lições aprendidas na Seção 5. Por fim, conclui-se o trabalho na Seção 6.

2. Fundamentação Teórica

O pensamento computacional é definido como uma forma de pensar que envolve a formulação de problemas e suas resoluções [Papert 1980, Wing 2006]. Além disso, ele deve ser estruturado de uma forma que possa ser efetivamente executada por um agente de processamento de informações [Wing 2011]. Entretanto, ele também é definido como o conjunto de habilidades como abstração, decomposição, reconhecimento de padrões, transformação de processos em algoritmos, depuração e resolução de problemas, que podem ser estendidos à habilidade do pensamento crítico [Belmar 2022]. Assim, sabendo da aplicação dessas habilidades, principalmente na área de tecnologia, ciência, engenharia e matemática, observou-se que o pensamento computacional é uma habilidade que todos devem ter ou deveriam trabalhar para adquirir [Lockwood and Mooney 2017, Belmar 2022].

¹Disponível em: <https://appinventor.mit.edu/>

²Disponível em: <https://www2.ufjf.br/getengcomp/>

Existem diferentes iniciativas para introduzir o pensamento computacional em sala de aula [García-Peñalvo and Mendes 2018]. Assim, quatro abordagens principais de se introduzir o pensamento computacional foram propostas [García-Peñalvo and Mendes 2018]:

1. Focando na semântica da linguagem, mais que a própria sintaxe da linguagem de programação;
2. Ambientes variados de programação sendo algo visual como MIT App Inventor ou linhas de código;
3. Usando controladores de robôs;
4. Por meio da construção de conjuntos físicos para controlar as coisas;

O ensino da programação contribui para o desenvolvimento do pensamento computacional, portanto é o meio normalmente escolhido para se envolver os alunos [Belmar 2022]. O ambiente de programação pode ser tanto visual, onde o usuário constrói uma aplicação com blocos de código, assim como no Scratch³ e no MIT App Inventor, quanto escrito, como utilizando linguagens Java e Python. Porém, os estudantes possuem uma maior facilidade e afinidade para uma programação mais visual do que escrita [St. Georgiev 2019]. Para tanto, aplicativos como Scratch e MIT App Inventor são normalmente a escolha para essa introdução da programação e do pensamento computacional.

O ensino de programação tende a seguir duas veredas principais [Belmar 2022]. A primeira é o ensino por meio de jogos, indo do design, construção e até o uso do mesmo para o aprendizado. Essa abordagem mostrou um efeito promissor no aprendizado de programação via jogos para alunos do ensino médio [Pellas and Vosinakis 2018]. A segunda, por meio da robótica educacional, com o uso de robôs de aprendizagem por interação oral em inglês [Lin and Hsu 2020].

Trabalhos relacionados ao pensamento lógico e computacional estão recebendo atenção nos últimos anos [Lockwood and Mooney 2017]. Entretanto, a aplicação do ensino do pensamento computacional vem crescendo de forma rápida [Tang et al. 2020]. Apesar da melhora no pensamento lógico trazido por essa capacidade, ainda há muita dificuldade de separar seu ensino das áreas relacionadas à ciência da computação. Isso se torna ainda mais complicado quando diversos países no mundo continuam no processo de introduzi-lo no seu currículo em todos os níveis educacionais [Lockwood and Mooney 2017]. Nesse contexto, o Brasil já estabeleceu na quinta competência da Base Nacional Comum Curricular (BNCC) o pensamento computacional como um eixo de ensino em sua base [BNCC 2018]. Porém, ainda há a necessidade de iniciativas e políticas públicas para promover a inclusão digital nas escolas dada a falta de projetos na área.

Um estudo que se atenta à essa necessidade no Brasil foi conduzido pelo projeto Escola de Games UFJF [Genesio et al. 2023], cujo objetivo é ensinar conceitos de lógica de programação para crianças e jovens por meio do desenvolvimento de jogos digitais. Assim, por meio de oficinas, o projeto promove para os estudantes da rede pública inclusão digital e desenvolvimento de habilidades importantes como o pensamento computacional.

³<https://scratch.mit.edu/>

3. Metodologia

Para alcançar o objetivo proposto, o GET-EngComp conduziu oficinas utilizando o MIT App Inventor na Escola Estadual Nyrce Villa Verde Coelho De Magalhães na cidade de Juiz de Fora (MG). Para o desenvolvimento dessas oficinas, seguiram-se as seguintes ações:

- **Planejamento da Ação:** o grupo planejou as oficinas considerando os tópicos iniciais que deveriam ser abordados com os alunos da escola pública. Nesse mesmo momento, verificou-se a quantidade total de aulas a serem ministradas para uma divisão de todo conteúdo a ser abordado.
- **Desenvolvimento dos Materiais:** o material da primeira aula foi desenvolvido pelos membros do GET considerando o desconhecimento do perfil dos alunos. Com isso, o material faz uma introdução sobre o computador e seus componentes, passando também sobre o que é a internet e aplicativos em geral. O restante dos materiais foram desenvolvidos pelo grupo nas semanas anteriores à aplicação, seguindo um planejamento flexível que listava todos os conteúdos a serem abordados. Dessa forma, quaisquer imprevistos ou aplicações canceladas poderiam ser contornados considerando que existia um número limitado de aulas. As aulas do projeto foram divididas entre os membros do GET, com responsáveis pela elaboração de material e apresentação aos alunos. Para cada semana foram definidos os responsáveis por desenvolver o material e também por conduzir as oficinas utilizando o material desenvolvido.
- **Aplicação:** com o planejamento feito e os responsáveis determinados as aulas puderam ser aplicadas na escola. Todas as exposições foram ministradas no laboratório de informática da escola.
- **Avaliação:** essa etapa considerou uma análise do impacto das aulas nos entendimentos dos alunos e em sua formação.

4. Relato de Experiência

A fim de aplicar a metodologia descrita na seção anterior, as aulas foram ministradas pelo GET aos alunos de Ensino Médio, que não possuíam experiência prévia na área de programação. Para descrever melhor as condições de aplicação das oficinas, o grupo coletou os dados demográficos dos alunos presentes e também dos instrutores, que estão expostos nas Tabelas 1 e 2.

Tabela 1. Informações sobre os estudantes.

Faixa etária	Idades de 16 a 18 anos
Número de estudantes	9 estudantes
Localidade	Escola Estadual Nyrce Villa Verde Coelho De Magalhães, Juiz de Fora
Informações socioeconômicas	Alunos de escola pública

Nas subseções abaixo serão também explicados os processos de desenvolvimento dos materiais e algumas experiências vividas e observadas na aplicação.

Tabela 2. Informações sobre os instrutores.

Faixa etária	Idades de 18 a 26 anos
Número de instrutores	11 instrutores
Educação	Estudantes de graduação em Engenharia Computacional

4.1. Materiais Didáticos

Para facilitar a compreensão das aulas, o grupo desenvolveu slides que contém o conteúdo de cada oficina com linguagem simples e exemplos próximos do cotidiano dos alunos. Esses slides utilizados no curso estão disponíveis no site do GET-EngComp⁴. A criação dos slides se mostrou muito importante para o acompanhamento dos alunos. Eles puderam compreender os conceitos apresentados e cada aluno pôde aplicá-los na criação de um aplicativo próprio. A divisão das aulas aplicadas e o conteúdo ofertado em cada uma delas está apresentado nas Tabelas 3 e 4.

Tabela 3. Título das aulas.

Número	Título
Aula 01	Introdução ao MIT App Inventor
Aula 02	Conceitos de interface e design
Aula 03	Introdução a elementos lógicos e de controle
Aula 04	Armazenamento de dados e loops
Aula 05	Introdução às listas
Aula 06	Sensores e multimídia
Aula 07	Finalização e apresentação dos aplicativos

4.2. Ambiente de Aplicação

As oficinas foram ministradas pelo GET no laboratório de informática da Escola Estadual Nyrce Villa Verde Coelho De Magalhães na cidade de Juiz de Fora/MG. Essas aulas ocorriam semanalmente às sextas-feiras no horário livre dos estudantes de Ensino Médio, logo após o almoço oferecido na escola. No mínimo dois membros do GET deveriam ir como ministrantes da aula. O laboratório da escola era equipado com 30 computadores em ótimo estado, que não apresentaram problemas durante a aplicação das aulas. Para a apresentação dos materiais, feitos em slides, foi utilizado o projetor do próprio grupo. Embora a escola tivesse seu próprio equipamento, o mesmo possuía uma qualidade de imagem inferior ao utilizado. Um compilado de informações relacionadas à aplicação do projeto podem ser vistas na Tabela 5.

Como já mencionado, a ferramenta utilizada foi o MIT App Inventor. Ela é uma ferramenta de criação de aplicativos que funciona pelo navegador do computador. Além disso, é necessário um aplicativo para celular chamado MIT AI2 Companion⁵, uma vez que é nele que o projeto desenvolvido é executado e testado. Com o objetivo de simplificar o processo de codificação, o MIT App Inventor utiliza da programação em blocos, que

⁴<https://www2.ufjf.br/getengcomp/projeto-mit-app-inventor/>

⁵<https://play.google.com/store/apps/details?id=edu.mit.appinventor.aicompanion3>

Tabela 4. Conteúdo das aulas aplicadas na escola.

Número	Conteúdo
Aula 01	Apresentação do GET. Apresentação da Engenharia Computacional e da Ciência da Computação. Apresentação do projeto Escola de Games. Dinâmica de apresentação dos alunos. Apresentação do computador e componentes. Apresentação do curso a ser dado. Apresentação do MIT App Inventor. Explicação do ambiente de desenvolvimento. Exportação e sincronização de projetos com o celular.
Aula 02	Design de interfaces de usuário. Criação de interfaces interativas. Exercícios práticos de design.
Aula 03	Introdução a elementos lógicos e de controle. Exercício práticos.
Aula 04	Conceitos de armazenamento de dados em aplicativos. Introdução a variáveis. Loops. Exercícios de variáveis e de loops.
Aula 05	Listas. Exercícios sobre Listas e loops.
Aula 06	Uso de sensores do celular. Multimídia. Início do desenvolvimento do projeto individual.
Aula 07	Auxílio aos alunos para terminarem seus aplicativos sem erros ou dúvidas. Apresentação dos aplicativos desenvolvidos por cada aluno.

facilita a visualização das etapas da programação. É válido ressaltar que apesar de o idioma padrão da ferramenta ser o inglês, a mesma possuía tradução para o Português Brasileiro, o que eliminou boa parte da barreira linguística para o aprendizado.

4.3. Atividades Desenvolvidas

Como é possível observar na Tabela 4, todas as aulas são divididas em duas partes: a parte teórica, em que é explicado o conteúdo da oficina, e a parte prática, em que é realizada a consolidação do ensino por meio de exercícios. Nas duas subseções abaixo estão

Tabela 5. Informações sobre a aplicação do projeto.

Tipo de aulas	Aulas expositivas seguindo slides com foco no desenvolvimento de aplicativos/jogos.
Parceria com o Escola de Games	Agosto de 2023
Parceria com a escola	Setembro de 2023
Ferramenta utilizada	MIT App Inventor
Duração das aulas	7 aulas (de 13h às 15h30 - 2h30min de duração)
Número de alunos	9 alunos em média
Período de aplicação	28 de setembro a 8 de dezembro
Recursos necessários	Computadores e <i>smartphones</i> com acesso à internet e aplicativo mobile MIT AI2 Companion.
Tempo de preparação	1 mês

detalhadas as atividades envolvidas em cada uma das partes.

4.3.1. Conteúdo Teórico

Todas as oficinas tiveram seu início na parte teórica, com explicações orais e com o auxílio dos slides sobre a matéria abordada na aula. Para conceitos mais complexos de serem explicados, como o conceito de listas, *loops*, ou variáveis de controle, os ministrantes também utilizaram o quadro branco para permitir a visualização dos conceitos ensinados.

Para aulas especialmente conceituais, como a primeira aula, que apresenta o computador e seus componentes aos alunos, o grupo observou a dificuldade de prender a atenção dos estudantes. Para remediar este problema, exemplos do cotidiano dos alunos foram utilizados para exemplificar os conceitos. Por exemplo, na segunda aula, os ministrantes apresentaram a interface de jogos famosos como exemplos de interfaces gráficas.

Vale ressaltar que todos os alunos presentes já apresentavam certa familiaridade com as ferramentas de um computador. Por esse motivo, os conceitos apresentados nas primeiras aulas foram assimilados mais facilmente.

4.3.2. Conteúdo Prático

Na segunda parte de todas as aulas, os responsáveis realizavam atividades práticas para reforçar os conteúdos aprendidos na parte teórica. Por exemplo, na primeira aula, o grupo apresentou um aplicativo simples facilmente replicável para que os alunos se acostumassem com a ferramenta MIT App Inventor.

Como é uma etapa mais dinâmica não houve muitos problemas relacionados à atenção dos alunos, todos ficaram engajados no processo de criação e dessa forma as dúvidas sobre os conceitos explicados apareceram.

Os ministrantes também aplicaram exercícios que incentivam a interação entre os alunos. Por exemplo, a sexta aula, em que, ao aprenderem o conceito de loops e listas, os alunos foram incentivados a criar um aplicativo que utilizaria destes conceitos para montar um quiz sobre algum assunto, como o filme ou jogo favorito.

4.4. Criação dos Aplicativos

Durante o processo de criação de aplicativos, ficou evidente a criatividade dos alunos. Cada um criou uma interface com elementos e organizações diferentes. Além disso, todos pensaram em mecânicas e funcionalidades diferentes para seus aplicativos. As aplicações demandavam desenvolvimento específico de cada projeto, o que expandiu o escopo das possibilidades de criação para os alunos.

A evolução dos alunos na utilização da ferramenta MIT App Inventor foi gratificante, todos se lembravam de técnicas explicadas em aulas passadas, alguns foram além, utilizando blocos que só seriam explicados em aulas futuras.

4.5. Absorção dos Conceitos

O principal objetivo das aulas era estimular o aprendizado de conceitos de computação através da criação de aplicativos com o MIT App Inventor. Nas aulas iniciais era visível o desconhecimento dos alunos com relação às ferramentas oferecidas no *software*. Isso se deve principalmente ao fato de que os blocos oferecidos para desenvolvimento no aplicativo já se aproximavam muito do que seria um código escrito em uma linguagem como C++ ou Python. Entretanto, nas aulas finais, era visível a maior compreensão dos alunos ao utilizarem cada bloco lógico. Isso demonstrou que não só haviam adquirido conhecimento sobre a utilização do MIT App Inventor, mas também sobre conceitos computacionais como estruturas lógicas e de repetição. É válido, portanto, ressaltar a importância de abordagens próximas ao cotidiano e interesse dos alunos. Dessa forma, um estímulo educacional diferente, como o desenvolvimento de aplicativos sem conhecimento prévio de programação, possibilita uma adesão maior e mais motivada aos conteúdos apresentados.

5. Lições Aprendidas

Diversos ensinamentos foram extraídos das oficinas aplicadas. Primeiramente, é importante adaptar as atividades e os materiais didáticos de acordo com o contexto dos estudantes. Isso não apenas os deixou mais envolvidos, mas também criou um ambiente mais propício ao aprendizado. Dessa forma, ao proporcionar um ambiente adequado, os estudantes se sentiram mais à vontade para fazer perguntas, o que contribuiu significativamente para o entendimento dos conceitos abordados.

A realização dessas atividades também evidenciou os benefícios do uso do MIT App Inventor no ensino de conceitos básicos de programação, especialmente para aqueles sem experiência prévia na área. A possibilidade de utilização do software diretamente no navegador facilitou a utilização. Além disso, a simplicidade do aplicativo potencializa o ensino da programação.

Além da criação de slides, o grupo percebeu que a gravação de videoaulas como material de apoio entre as aulas auxiliaria na fixação dos conhecimentos. Os vídeos podem prover explicação adicional sobre o conteúdo de cada aula, além de ser uma fonte de

conteúdo para os alunos ausentes. Estas videoaulas já estão sendo gravadas no Centro de Educação a Distância (CEAD) da UFJF e serão utilizadas em novas versões do curso.

O GET também notou a necessidade de uma avaliação sobre a metodologia conduzida. Apesar disso, o grupo observou como todos os alunos demonstraram estarem satisfeitos com o aprendizado adquirido. A experiência indicou que os alunos são capazes de desenvolver seus próprios aplicativos com os materiais didáticos adequados. Isso foi feito estabelecendo uma conexão prática e teórica entre os conceitos de jogos e a lógica de programação.

6. Considerações Finais

Esse trabalho descreveu um relato de experiência mostrando as observações realizada no ensino de programação utilizando MIT App Inventor para alunos do Ensino Médio. A atividade desenvolvida pelos integrantes do grupo GET Engenharia Computacional promoveu uma aproximação dos alunos da Escola Estadual Nyrce Villa Verde Coelho De Magalhães a essa realidade mais digital. Dessa forma, instigou também através das atividades trabalhadas o desenvolvimento do pensamento computacional nos 9 alunos participantes. A escolha da utilização do MIT App Inventor se mostrou bastante assertiva, por se tratar de uma abordagem mais amigável e visual ao usuário, facilitando o processo de ensino durante a experiência.

Outro fator a se levantar é a replicabilidade da experiência realizada. Portanto, as informações apresentadas no artigo podem ser utilizadas como referência por outros professores e grupos que também estejam interessados em atividades que venham estimular o ensino de programação e desenvolvimento do pensamento computacional em suas instituições de ensino. Assim, as informações, de todos os desafios enfrentados e as estratégias utilizadas podem servir como um guia de como conduzir uma experiência semelhante de forma bem-sucedida. Dessa forma, fica claro também a importância de compartilhar a experiência tida, pois ela serve não só para o incentivo de novos interessados em aplicar essa abordagem, mas também preparar alunos para essa realidade mais tecnológica.

Uma limitação encontrada nesse estudo é a falta de um trabalho de análise e avaliação dos resultados do ensino dos conceitos, assim como uma avaliação de satisfação dos alunos para com o curso ministrado. Isso se deve ao fato de as avaliações conduzidas acerca desses tópicos não terem sido documentadas, prejudicando a argumentação posta na Seção 4.5. Porém, isso é algo que será retificado nas próximas edições, buscando aplicar metodologias de avaliação de cursos e oficinas similares disponíveis na literatura.

Por fim, tem-se que a experiência afetou de maneira positiva os estudantes. Isso torna claro ao ver como a experiência incentivou dois dos estudantes que participaram a ingressarem no ensino superior. Principalmente a influência do desenvolvimento do pensamento computacional nesses alunos, já que um deles se sentiu confortável a escolher o curso de ciências exatas, onde essa habilidade é essencial.

Referências

Belmar, H. (2022). Review on the teaching of programming and computational thinking in the world. *Frontiers in Computer Science*, 4.

- BNCC (2018). Base nacional comum curricular. Ministério da Educação.
- García-Peñalvo, F. J. (2018). Editorial computational thinking. *IEEE Revista Iberoamericana de Tecnologías del Aprendizaje*, 13(1):17–19.
- García-Peñalvo, F. J. and Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80:407–411.
- Genesio, N., Menezes, M., Almeida, J., Boaventura, A., and Valle, P. (2023). Aprendendo lógica de programação desenvolvendo jogos digitais: Um relato de experiência. In *Anais do XXIX Workshop de Informática na Escola*, pages 375–386, Porto Alegre, RS, Brasil. SBC.
- Lin, Y.-T. and Hsu, T.-C. (2020). Effects of using mobile phone programs to control educational robots on the programming self-efficacy of the third grade students. In *Proceedings of International Conference on Computational Thinking Education*.
- Lockwood, J. and Mooney, A. (2017). Computational thinking in education: Where does it fit? A systematic literary review. *CoRR*, abs/1703.07659.
- Lye, S. Y. and Koh, J. H. L. (2014). Review on teaching and learning of computational thinking through programming: What is next for k-12? *Computers in Human Behavior*, 41:51–61.
- Manovich, L. (2013). *Software takes command*. Bloomsbury Academic.
- Papert, S. (1980). *Mindstorms: children, computers, and powerful ideas*. Basic Books, Inc., USA.
- Pellas, N. and Vosinakis, S. (2018). The effect of simulation games on learning computer programming: A comparative study on high school students’ learning performance by assessing computational problem-solving strategies. *Education and Information Technologies*, 23(6):2423–2452.
- St. Georgiev, T. (2019). Students’ viewpoint about using mit app inventor in education. In *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, pages 611–616.
- Tang, X., Yin, Y., Lin, Q., Hadad, R., and Zhai, X. (2020). Assessing computational thinking: A systematic review of empirical studies. *Computers & Education*, 148:103798.
- Wing, J. (2011). Research notebook: Computational thinking—what and why. *The link magazine*, 6:20–23.
- Wing, J. M. (2006). Computational thinking. *Commun. ACM*, 49(3):33–35.