

Desenvolvimento do Pensamento Computacional de Estudantes da Educação Básica com MIT App Inventor

Sivaldo Joaquim^{1,2}, Fernando Silvio Cavalcante Pimentel¹,
Alan Pedro da Silva^{1,2}, Ig Ibert Bittencourt^{2,3}

¹Comunidades Virtuais Ufal – Centro de Educação (CEDU)
Universidade Federal de Alagoas (UFAL) – Maceió, AL – Brasil

²Núcleo de Excelência em Tecnologias Sociais (NEES) – Instituto de Computação (IC)
Universidade Federal de Alagoas (UFAL) – Maceió, AL – Brasil

³Harvard Graduate School of Education, Harvard University – Cambridge, MA – USA

{sivaldojoaquim, ig.ibert, alanpedro}@ic.ufal.br,
fernando.pimentel@cedu.ufal.br

Resumo. *A implementação do Ensino de Computação na Educação Básica busca favorecer o desenvolvimento do Pensamento Computacional e estimular a construção do raciocínio lógico, crítico, criativo e resolução de problemas, habilidades essenciais para o progresso acadêmico e profissional dos estudantes. No entanto, isso ainda é um desafio nas escolas brasileiras. Essa investigação tem por objetivo relatar a experiência de utilização da plataforma MIT App Inventor com 46 estudantes do 8º ano do Ensino Fundamental II. Para isso, foi realizada uma pesquisa quantitativa com abordagem descritiva na avaliação de 32 projetos desenvolvidos pelos estudantes para dispositivos móveis. Como resultados, observou-se desempenho significativos em critérios como construção de telas, nomeação de componentes e procedimentos, eventos, criação de variáveis, o uso de strings, operadores aritméticos, relacionais e expressões lógicas.*

Abstract. *The implementation of Computer Teaching in Basic Education search to support the development of Computational Thinking and to stimulate the construction of logical, critical, creative reasoning and problems solving, essential skills for the academic and professional progress of students. However, this is still a challenge in Brazilian schools. This investigation aims to report the experience of using the MIT App Inventor platform with 46 students from the 8th year of Elementary School II. For this, a quantitative research was performed with a descriptive approach in the evaluation of 32 projects developed by students for mobile devices. As results, it was observed significant performance in criteria's as building screens, components and procedures appointment, events, variables creation, strings usage, arithmetic operators, relational and logical expressions.*

1. Introdução

As transformações resultantes do desenvolvimento das Tecnologias Digitais (TD) requerem habilidades para atender os desafios dos atuais e novos modelos de negócios. Segundo um relatório do *World Economic Forum* (2016), 65% das crianças que ingressam na Educação Básica hoje, ao concluir seus estudos, irão trabalhar em funções completamente novas, que ainda não existem. Neste contexto, para atender as atuais e futuras demandas do mercado, surgem diversos movimentos e iniciativas para implementação do ensino de informática nas escolas (COMMITTEE ON EUROPEAN COMPUTING EDUCATION, 2017; DAGIENÉ *et al.*, 2022; CASPERSEN *et al.*, 2023; INFORMATICS FOR ALL, 2023; HOUR OF CODE, 2023; CODE.ORG, 2023; GOOGLE FOR EDUCATION, 2023; PROGRAMAÊ, 2023). Nesse sentido, a *International Society for Technology in Education (ISTE)* tem proposto padrões para orientar professores, gestores educacionais e formuladores de políticas públicas no desenho e implementação de currículos para potencializar o uso adequado das TD na prática pedagógica (CROMPTON e SYKORA, 2021; ISTE, 2023) com ênfase no desenvolvimento do Pensamento Computacional (PC) dos estudantes, visando desenvolver habilidades essenciais do século XXI (CASPERSEN *et al.*, 2019; OECD, 2020), *e.g.*, criatividade, pensamento crítico, raciocínio lógico e resolução de problemas.

No Brasil, a utilização de TD na escola com foco no PC dos estudantes da Educação Básica, de forma crítica, reflexiva e significativa em diferentes níveis e contextos educacionais têm chamado atenção da comunidade acadêmica e científica (KOHLENER *et al.*, 2021). Além disso, o Ministério da Educação (MEC) incluiu a “Cultura Digital” como uma das dez competências gerais na Base Nacional Curricular (BNCC) (BRASIL, 2018). A Sociedade Brasileira de Computação (SBC) tem discutido e construído diretrizes e itinerário formativo para o ensino de Computação na Educação Básica em três eixos: PC, mundo digital e cultura digital (SBC, 2019a). O Centro de Inovação para a Educação Brasileira (CIEB) elaborou um Currículo de Referência em Tecnologia e Computação alinhado às competências e habilidades propostas na BNCC (CIEB, 2019) e mais recentemente, o Conselho Nacional de Educação (CNE) aprovou as “Normas sobre Computação na Educação Básica – complemento à BNCC” (BRASIL, 2022).

Segundo Wing (2017), PC envolve o processo de formulação de problemas e soluções de forma que possam ser executados por computadores ou humanos. Neste contexto, a pesquisadora destaca três pontos importantes: (i) humanos computam; (ii) as pessoas podem aprender PC sem o uso de computadores; e (iii) os computadores atualmente combinam inteligências dos humanos e máquinas. Para SBC, o termo PC faz referência à capacidade de “compreender, definir, modelar, comparar, solucionar, automatizar e analisar problemas (e soluções) de forma metódica e sistemática, através da construção de algoritmos” (SBC, 2019b, p. 5). Diante disso, desenvolver atividades e projetos que envolvam os estudantes da Educação Básica na decomposição de tarefas complexas, abstração, análises e resolução de problemas do mundo real com base em conceitos da área de Ciência da Computação torna-se essencial para o desenvolvimento do PC. Além disso, a BNCC propõe para essa etapa de formação estudantil a compreensão das TD, inclusive para “utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva”

(BRASIL, 2018, p. 9), que requer não só a inclusão de uma disciplina, mas o repensar do currículo.

Segundo Santana e Oliveira (2019), programar um computador requer habilidades fundamentais como analisar problemas, planejar, desenvolver e testar soluções computacionais (*e.g.*, aplicativos e jogos digitais). Nesse sentido, o presente artigo relata a experiência de utilização da plataforma MIT App Inventor (PATTON; TISSENBAUM e HARUNANI, 2019) com objetivo de desenvolver o PC dos estudantes da Educação Básica em uma Escola de Educação Infantil, Ensino Fundamental e Médio, situada ao Sul do Estado de Alagoas. Para isso, foi realizada uma pesquisa do tipo quantitativa, com abordagem descritiva na avaliação de 32 projetos desenvolvidos pelos estudantes para dispositivos móveis. Os principais resultados obtidos mostram que houve desempenho significativos dos estudantes em critérios relacionados ao PC, tais como: construção de telas, nomeação de componentes e procedimentos, eventos, criação de variáveis, o uso de *strings*, operadores aritméticos, relacionais e expressões lógicas.

Este artigo apresenta a seguinte estrutura: na seção 2 apresentamos de forma sucinta alguns trabalhos relacionados essenciais para fundamentação teórica do presente estudo; na seção 3 descrevemos sobre os materiais e métodos aplicados na condução dessa pesquisa; na seção 4 apresentamos os resultados e discussão; e por último, na seção 5 as considerações finais.

2. Trabalhos Relacionados

Diferentes estudos têm abordado o ensino de programação de computadores com estudantes da Educação Básica visando investigar o desenvolvimento do PC (GARCÍA-PEÑALVO E MENDES, 2018; MEDEIROS; VON WANGENHEIM e HAUCK, 2021; RODRIGUES; SILVA e CARNEIRO, 2021). O objetivo se concentra em estimular habilidades como criatividade, trabalho colaborativo, atenção e concentração com o uso de recursos como papel na realização de atividades “desplugadas”, jogos digitais e ambientes de desenvolvimento *online* com ênfase na construção de algoritmos computacionais, tais como: Code.org, Scratch e MIT App Inventor (WERLICH *et al.*, 2018; SANTANA e SANTOS, 2021; KOHLER *et al.*, 2021; SILVA *et al.*, 2021).

Nesse contexto, Rodrigues, Silva e Carneiro (2021) relatam o desenvolvimento do PC com 15 estudantes do quinto ano do Ensino Fundamental de uma escola da rede pública de ensino do município de Monte Carmelo, MG. Para isso, foram realizadas oficinas de “Computação Desplugadas” e com o uso do ambiente de programação Scratch, durante um período de duas semanas, totalizando seis aulas com 1h30min de duração cada aula. Os pesquisadores destacam que houve uma maior aceitação das atividades de “Computação Desplugadas” em relação ao uso do Scratch. Além disso, segundo Rodrigues, Silva e Carneiro (2021, p. 234) os estudantes “se sentiram mais motivados e empolgados” e resolveram os exercícios propostos de forma mais fácil. Durante a realização da oficina, observou-se o desenvolvimento de trabalho em equipe e interação social entre os participantes. No entanto, com o uso do “Scratch” os meninos conseguiram assimilar mais facilmente os conceitos do que as meninas. De acordo com os autores, isso se deve pelo fato de que os meninos possuíam familiaridade com o ambiente de programação e com os “jogos de videogame” (RODRIGUES; SILVA e CARNEIRO, 2021).

Brackmann, Caetano e Silva (2020) realizaram um estudo do tipo quase-experimental com aplicação de pré-teste e pós-teste visando verificar eficácia de intervenções do PC desplugado com 63 estudantes do 5º ano e 6º ano do Ensino Fundamental. O referido estudo foi realizado em duas escolas da rede pública de ensino, durante cinco semanas, com duas horas de aulas por semana, totalizando 10 aulas no período de 2016 e 2017, divididos em dois grupos: Intervenção e Controle, no município de Santa Maria, RS. Os principais resultados mostraram uma melhoria estaticamente significativa de 17,03% no desempenho de aprendizagem dos estudantes do grupo intervenção. Os estudantes do grupo de controle obtiveram um aumento de 2,89% no desempenho.

Santana e Santos (2021) conduziram um curso de computação criativa com objetivo de verificar o desenvolvimento de habilidades como criatividade e colaboração no ano de 2018 com 45 estudantes do 8º ano e 9º ano; e em 2019 com 73 estudantes do 6º ano e 7º ano do Ensino Fundamental, sendo 72 estudantes de escolas da rede pública de ensino e 46 de escolas da rede particular com duração de 30 horas, na cidade de Piritiba, BA. Os principais resultados evidenciaram o desenvolvimento de habilidades essenciais do PC, tais como: criatividade, colaboração e pensamento crítico. Além disso, observou-se o desenvolvimento pessoal dos participantes e maior interesse em relação aos aspectos computacionais.

Diferentemente das abordagens utilizadas por Brackmann, Caetano e Silva (2020); Santana e Santos (2021); Rodrigues, Silva e Carneiro (2021) que utilizaram atividades desplugadas e o ambiente de programação Scratch para construção de jogos em períodos relativamente curtos com estudantes da Educação Básica. O presente estudo foi realizado com estudantes do Ensino Fundamental II por meio da plataforma MIT App Inventor durante o primeiro semestre de 2022 (período de janeiro a junho) para a criação de *Apps* educacionais. Além disso, para verificar o desenvolvimento do PC foi utilizada uma ferramenta *online* gratuita validada por (VON WANGENHEIM *et al.*, 2018) para avaliação automatizada dos projetos criados pelos estudantes com objetivo de analisar conceitos essenciais de programação e design de interface dos *Apps*.

3. Materiais e Métodos

Este estudo foi realizado com 46 estudantes de duas turmas do 8º ano do Ensino Fundamental II. Desse total, temos 18 do gênero masculino e 28 do gênero feminino na faixa etária entre 13 e 15 anos de idade, com a utilização da plataforma MIT App Inventor, em uma escola de Educação Básica, situada na região do Baixo do São Francisco, localizado ao sul do Estado de Alagoas. Como método de pesquisa, foi utilizada uma abordagem quantitativa. Segundo Mattar e Ramos (2021, p. 133) “o objetivo geral das pesquisas quantitativas é explicar e prever fenômenos”. Neste contexto, é fundamental descrever as etapas e procedimentos aplicados na coleta e análises de dados para tomadas de decisão. Mattar e Ramos (2021, p. 133) ressaltam que “a construção de instrumentos e a definição de procedimentos para a coleta de dados precisam ser concebidas considerando essas tomadas de decisão prévias, que afetam as possibilidades de análise e os tipos de resultados que podem ser alcançados”. Para isso, foram executadas as seguintes etapas: (i) escolha da ferramenta de desenvolvimento; (ii) seleção dos materiais de apoio aos processos de ensino e aprendizagem; (iii) execução das aulas/atividades; (iv) coleta de dados; e (v) análise dos resultados e discussão.

Na primeira etapa, realizou-se uma pesquisa visando selecionar com base na literatura científica uma ferramenta gratuita para desenvolvimento de aplicativos (*Apps*) para dispositivos móveis que oferecesse aos estudantes maior interatividade e usabilidade nos processos de ensino e aprendizagem (LEÔNCIO *et al.*, 2017; CHING; HSU e BALDWIN, 2018). Como resultado, foi selecionado a plataforma MIT App Inventor¹, um ambiente de programação visual *online* e intuitivo que possibilita a criação de aplicativos funcionais para *smartphones* e *tablets* com sistemas operacionais Android e iOS. Desenvolvido originalmente pelo *Google* e atualmente mantido pelo *Massachusetts Institute of Technology (MIT)* (LEÔNCIO *et al.*, 2017; ALVES; VON WANGENHEIM; HAUCK, 2019), presente em 195 países, com 18,1 milhões de usuários registrados e mais de 85,4 milhões de *Apps* criados na plataforma. O MIT App Inventor é um ambiente de programação baseado em blocos para desenvolvimento de *Apps* complexos e de alto impacto por estudantes da Educação Básica com pouco conhecimento na construção de *Apps* em um tempo significativamente menor, quando comparado aos ambientes de programação tradicionais (MIT APP INVENTOR, 2023).

Na segunda etapa, buscou-se na *web* materiais de referências para apoiar os processos de ensino e de aprendizagem durante o desenvolvimento de *Apps* com o uso da plataforma MIT App Inventor. Nesta fase, foram selecionados e adotados como materiais para condução das aulas e atividades o livro “*App Inventor 2: Create your own Android Apps*” (WOLBER *et al.*, 2014). Além disso, foram consultados tutoriais disponíveis nos sites do MIT App Inventor³ e do projeto de extensão “Aprenda a Programar Jogando”⁴ da Universidade Federal do Rio de Janeiro (UFRJ), Campus Macaé.

Na terceira etapa, foram conduzidas aulas e atividades no laboratório de informática da própria escola no período de fevereiro a junho de 2022, com duração de 50 minutos/aula, uma vez por semana. Nesse período, os estudantes conheceram a plataforma de programação e exploraram alguns recursos disponíveis no site do MIT App Inventor. Durante as aulas, foram abordados conceitos introdutórios sobre o ambiente de programação, as características, depuração, algoritmos, variáveis, estruturas condicionais simples e composta, estruturas de repetição, operadores lógicos e matemáticos. Os estudantes realizaram várias atividades durante o semestre e como projeto final foi proposto o desenvolvimento de um *App* educacional individualmente no formato de um *quiz*, contendo no mínimo dez questões sobre conhecimentos gerais ou relacionado alguma disciplina específica (*e.g.*, Língua Portuguesa, Matemática, História, Geografia, Língua Inglesa).

Na quarta etapa, realizou-se a coleta de dados a partir dos projetos desenvolvidos na plataforma MIT App Inventor *online*. Nessa fase, os estudantes fizeram o *download* dos projetos e salvaram em um *pen drive* entregue pelo professor para posterior avaliação dos *Apps* desenvolvidos no laboratório de informática.

Por último, na quinta etapa, foi utilizado o *software* CodeMaster⁵ para realizar análise quantitativa dos projetos desenvolvidos pelos estudantes na plataforma MIT App Inventor (VON WANGENHEIM *et al.*, 2018; ALVES; VON WANGENHEIM; HAUCK,

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

² Disponível em: <http://www.appinventor.org/book2>

³ Disponível em: <https://appinventor.mit.edu/explore/ai2/tutorials>

⁴ Disponível em: <https://aprendaprogramar.macaue.ufrj.br/index.php/atividades/tutoriais/>

⁵ Disponível em: <http://apps.computacaonaescola.ufsc.br/codemaster/>

2019; ALVES *et al.*, 2020). Segundo Von Wangenheim *et al.* (2018) o CodeMaster é uma ferramenta *web* gratuita utilizada para avaliar e classificar automaticamente projetos desenvolvidos em um contexto de aprendizado baseado em problemas através das plataformas MIT App Inventor e Snap. Na avaliação automática dos projetos, a ferramenta CodeMaster atribuiu uma pontuação para cada critério relacionado ao PC com base na “Rubrica de avaliação do App Inventor⁶” (ALVES *et al.*, 2020). Além disso, os dados gerados através do CodeMaster foram tabulados e processados com o uso do *software R*⁷.

4. Resultados e Discussão

Dos 46 estudantes, 32 entregaram os projetos no prazo definido pelo professor. Neste artigo, foram avaliados $N = 32$ aplicativos desenvolvidos em formato de *quiz* com uma abordagem interdisciplinar, como parte do projeto proposto na disciplina de Informática. Na Tabela 1, observa-se a estatística descritiva contendo as medidas de tendência central e dispersão dos critérios relacionados aos conceitos do PC obtidos a partir dos dados gerados pela ferramenta *CodeMaster* (VON WANGENHEIM *et al.*, 2018; ALVES; VON WANGENHEIM; HAUCK, 2019; ALVES *et al.*, 2020).

Tabela 1. Sumarização dos dados relacionados aos critérios do PC.

Critérios	Mínimo	1° Quartil	Mediana (2° Quartil)	Média	3° Quartil	Máximo	Variância	Desvio Padrão
Telas	1.000	3.000	3.000	2.625	3.000	3.000	0.629	0.793
Nomeação: componentes, variáveis e procedimentos	1.000	2.000	3.000	2.562	3.000	3.000	0.383	0.618
Eventos	1.000	3.000	3.000	2.875	3.000	3.000	0.177	0.421
Abstração de procedimentos	0.000	0.000	0.000	0.281	0.000	3.000	0.789	0.888
Laços	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Condicionais	0.000	0.000	0.000	0.313	0.000	3.000	0.737	0.859
Listas	0.000	0.000	0.000	0.188	0.000	2.000	0.350	0.592
Persistência de dados	0.000	0.000	0.000	0.063	0.000	2.000	0.125	0.353
Sensores	0.000	0.000	0.000	0.125	0.000	2.000	0.177	0.421
Desenho e animação	0.000	0.000	0.000	0.250	0.000	3.000	0.645	0.803
Operadores	0.000	1.000	1.000	1.156	1.000	3.000	0.652	0.807
Variáveis	1.000	2.000	2.000	1.875	2.000	2.000	0.112	0.336
<i>Strings</i>	0.000	1.000	1.000	1.031	1.000	2.000	0.095	0.309
Sincronização	0.000	0.000	0.000	0.094	0.000	1.000	0.087	0.296
Mapas	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
Extensões	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000

Conforme podemos observar na Tabela 1, a média obtida no desenvolvimento das “telas” foi $\bar{X} = 2.625$, mediana (Md) = 3.000 e desvio padrão (S) = 0.793. Esse resultado mostra que 81% dos aplicativos criados pelos estudantes possuem mais de quatro telas, sendo que as mesmas foram programadas para alterar seus estados durante o período de execução do *App* e 19% tem apenas uma tela. Segundo García-Peñalvo e Mendes (2018); Rodrigues, Silva e Carneiro (2021), esse pensamento lógico para resolução de problemas

⁶ Disponível em: http://apps.computacaonaescola.ufsc.br:8080/rubrica_appinventor.jsp

⁷ Disponível em: <https://www.r-project.org/>

por meio de uma abordagem ativa, configura-se como elemento fundamental do PC no processo de construção de artefatos reais ou virtuais. Resultado semelhante foi obtido no critério de “**nomeação: componentes, variáveis e procedimentos**”, com $\bar{X} = 2.562$, $Md = 3.000$ e $S = 0.618$. Isso revela que 90% dos nomes de componentes, variáveis e procedimentos foram alterados do padrão para resolver problemas utilizando o PC (GARCÍA-PEÑALVO E MENDES, 2018). Além disso, observou-se a atenção dos estudantes com relação a nomeação dos componentes durante a criação do *design* dos aplicativos para facilitar a implementação da lógica de programação. De acordo com Leôncio *et al.* (2017), esses achados mostram o desenvolvimento do PC dos estudantes na elaboração do *layout* das telas e inserção dos componentes com foco na sequência lógica dos códigos para o funcionamento dos aplicativos.

No critério “**eventos**” obteve-se uma $\bar{X} = 2.875$, $Md = 3.000$ e $S = 0.421$. Esse resultado mostra que 90,6% dos aplicativos contêm mais de dois tipos de manipuladores de eventos. 6,3% possuem exatamente dois tipos de manipuladores e apenas 3,1% implementou um único tipo de evento. Segundo Medeiros, Von Wangenheim e Hauck (2021), isso mostra o desenvolvimento de habilidades essenciais dos estudantes da Educação Básica para criação de *apps* totalmente funcionais. No geral, os estudantes implementaram nos *apps* eventos do tipo ao clicar sobre um botão na tela principal, carrega uma segunda tela contendo a primeira questão e alternativas do *quiz*, da mesma forma quando o usuário seleciona uma alternativa, um evento chama a próxima questão e assim sucessivamente, no final o aplicativo mostra a pontuação total e se desejar, o usuário pode reiniciar o *quiz*. Santana e Santos (2021) destacam o uso do conceito de eventos no desenvolvimento de projetos com o ambiente de programação *Scratch* pelos estudantes da Educação Básica para tornar os aplicativos mais interativos.

Entretanto, no critério de “**abstração de procedimentos**” nota-se um baixo desempenho com $\bar{X} = 0.281$, $Md = 0.000$ e $S = 0.888$. Isso significa que 90,6% dos aplicativos não existem procedimentos e apenas 9,4% dos estudantes programaram nos *Apps* procedimentos para decompor problemas complexos, reduzir a duplicação e reutilização do código (blocos) na plataforma MIT App Inventor. Resultados semelhantes foram identificados no estudo realizado por Santana e Oliveira (2019). Segundo os pesquisadores, existem um baixo desempenho em “habilidades de abstrair e decompor problemas em partes menores, de modo que torne mais fácil a compreensão do algoritmo e facilite as futuras correções e/ou alterações no código” (SANTANA E OLIVEIRA, 2019, p. 164).

Outro critério observado nos projetos diz respeito aos “**laços**” com $\bar{X} = 0.000$, $Md = 0.000$ e $S = 0.000$. Neste caso, nota-se que não foi implementado estrutura de repetição do tipo “*while*” e “*for each*” no desenvolvimento dos projetos. Santana e Oliveira (2019) observam poucas implementações do tipo estrutura de repetição (*e.g.*, sempre, repita “x” vezes, e repita até que “x”) para controlar o comportamento dos personagens nos jogos desenvolvidos pelos estudantes do Ensino Fundamental. De acordo com Brackmann, Caetano e Silva (2019), Santana e Santos (2021) os estudantes da Educação Básica apresentam muitas dificuldades em implementar nos projetos estruturas de repetição (laços). O mesmo acontece com as estruturas “**condicionais**” ($\bar{X} = 0.313$, $Md = 0.000$ e $S = 0.859$). Isso mostra que 88% dos estudantes não implementaram estruturas condicionais, enquanto que 6% utilizaram estrutura condicional composta “*if... then... else*” e 6% condicionais em sequência “*if... else... if*”. Segundo Carlos, Godinho e

Gomide (2018) uma das maiores dificuldades encontradas pelos estudantes está relacionada a compreensão e utilização de estruturas condicionais.

Resultados semelhantes foram identificados nos critérios “**listas**” ($\bar{X} = 0.188$, $Md = 0.000$ e $S = 0.592$), 90,6% não implementaram listas e apenas 9,4% usaram listas unidimensional do tipo “*make a list*” e “*select list item*”; “**persistência de dados**” ($\bar{X} = 0.063$, $Md = 0.000$ e $S = 0.353$), identificou-se que 97% dos estudantes criaram variáveis para guardar os dados temporariamente e 3% implementaram um único componente de banco de dados “*TinyDB*” para armazenamento no dispositivo móvel do usuário; “**sensores**” ($\bar{X} = 0.125$, $Md = 0.000$ e $S = 0.421$), 91% não usaram, 6% um único tipo de sensor “*Clock*” e 3% implementaram dois tipos de sensores “*AccelerometerSensor*” e “*Clock*” nos *Apps*; “**desenho e animação**” ($\bar{X} = 0.250$, $Md = 0.000$ e $S = 0.803$), 90,6% não implementaram comandos de animação, 3,1% fez uso de comando de animação com componente pré-definido e 6,3% implementaram comando de animação com imagem nos aplicativos. Diante disso, ressaltamos que essa foi a primeira experiência dos estudantes com a criação de aplicativos na plataforma MIT App Inventor e por tanto, ainda estão em processo de desenvolvimento das habilidades do PC (WERLICH *et al.*, 2018; KOHLER *et al.*, 2021).

No entanto, nos critérios “**operadores**” ($\bar{X} = 1.156$, $Md = 1.000$ e $S = 0.807$) observa-se que em 12,5% dos aplicativos não foi utilizado, 71,9% fez uso de operadores aritméticos (adição, subtração, multiplicação, divisão), 3,1% utilizou operadores relacionais (maior que, menor que, igualdade, diferente de) e 12,5% possui operadores *booleanos* (lógicos) “*and, or, not*”. De acordo com Santana e Oliveira (2019), esses projetos estão acima da média por apresentar em suas estruturas, operadores que combinam as condições, de forma a mostrar resultados diferentes dependendo das situações implementadas; “**variáveis**” nota-se um resultado significativo ($\bar{X} = 1.875$, $Md = 2.000$ e $S = 0.336$), em 12,5% dos aplicativos utilizaram variáveis predefinidas e 87,5% foram criadas pelos estudantes para o armazenamento de dados; e “**strings**” ($\bar{X} = 1.031$, $Md = 1.000$ e $S = 0.309$), 97% dos estudantes utilizaram comando de criação de *string* para alterar textos de elementos e apenas 3% não utilizaram *string* no desenvolvimento dos aplicativos. Santana e Santos (2021) relatam que os estudantes tiveram muitas dificuldades na compreensão de conceitos de variáveis e operadores relacionais.

Porém, nota-se um baixo desempenho no critério “**sincronização**” ($\bar{X} = 0.094$, $Md = 0.000$ e $S = 0.296$). Esse resultado mostra que 90,6% dos estudantes não utilizaram temporizadores para sincronização e apenas 9,4% implementaram essa técnica no desenvolvimento dos aplicativos. Santana e Santos (2021) relatam que os estudantes tiveram dificuldades em implementar tarefas sincronizadas no desenvolvimento de aplicativos. Além disso, não foram identificados o uso de comandos associados aos componentes “**mapas**” e “**extensões**” no MIT App Inventor.

5. Limites do Estudo

Como toda pesquisa científica, o presente estudo possui algumas limitações, tais como: os resultados apresentados podem ser representativos apenas na região onde foi realizada a investigação. O período de condução das aulas no laboratório de informática pode ter sido insuficiente para o desenvolvimento do PC dos estudantes e esse fator pode ter contribuído para ausência ou baixa implementação de aspectos fundamentais, como: abstração de procedimentos, laços, condicionais, listas, persistência de dados, o uso de

sensores, desenho e animação, mapas e extensões nos projetos desenvolvidos pelos estudantes da Educação Básica.

6. Considerações Finais

O presente estudo relatou o uso da plataforma de desenvolvimento de aplicativos para dispositivos móveis MIT App Inventor com 46 estudantes do 8º ano do Ensino Fundamental II. Os principais resultados mostraram desempenho significativos em aspectos do PC, como: construção de telas, nomeação de componentes, variáveis e procedimentos, na implementação de eventos, *strings*, operadores aritméticos, relacionais e expressões lógicas. Entretanto, há necessidade de melhorar outros critérios, tais como: abstração de procedimentos, laços, condicionais, listas, persistência de dados, o uso de sensores, desenho e animação, a utilização de mapas e extensões no ambiente de programação MIT App Inventor.

Como trabalhos futuros, pretende-se realizar oficinas com estudantes da rede pública de ensino em localidades geograficamente diferentes visando desenvolver o raciocínio lógico para implementação de soluções computacionais com impacto social na comunidade.

Referências

- Alves, N. D. C., von Wangenheim, C. G., Hauck, J. C. R., & Borgatto, A. F. (2020). A large-scale evaluation of a rubric for the automatic assessment of algorithms and programming concepts. In *Proceedings of the 51st ACM technical symposium on computer science education* (pp. 556-562).
- Alves, N., von Wangenheim, C. G., e Hauck, J. (2019). Um Modelo de Avaliação do Pensamento Computacional na Educação Básica através da Análise de Código de Linguagem de Programação Visual. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* (Vol. 8, No. 1, p. 1021).
- Brackmann, C. P., Caetano, S. V. N., & Silva, A. R. (2020). Pensamento Computacional Desplugado: ensino e avaliação na educação primária brasileira. *RENOTE*, 17(3), 636–647. <https://doi.org/10.22456/1679-1916.99894>
- Brasil, Ministério da Educação. (2018). Base Nacional Comum Curricular. Disponível em: <http://basenacionalcomum.mec.gov.br/>. Acesso em: 31 mai. 2022.
- Brasil, Ministério da Educação. Conselho Nacional de Educação. (2022). Normas sobre Computação na Educação Básica – Complemento à BNCC. Disponível em: <https://bit.ly/3wZ2aWe>. Acesso em: 31 mai. 2022.
- Carlos, L., Godinho, J., & Gomide, J. (2018). Um Relato de Experiência da Escola de Verão de Programação para Crianças. In *Anais do XXIV Workshop de Informática na Escola*, (pp. 41-50). Porto Alegre: SBC. doi:10.5753/cbie.wie.2018.41
- Caspersen, M. E., Gal-Ezer, J., McGettrick, A., & Nardelli, E. (2019). Informatics as a fundamental discipline for the 21st century. *Communications of the ACM*, 62(4), 58-58.
- Caspersen, M., Gal-Ezer, J., McGettrick, A., & Nardelli, E. (2023). Informatics Education for School: A European Initiative. *ACM Inroads*, 14(1), 49-53. <https://doi.org/10.1145/3583088>

- Ching, YH., Hsu, YC. & Baldwin, S. (2018). Developing Computational Thinking with Educational Technologies for Young Learners. *TechTrends* 62, 563–573. <https://doi.org/10.1007/s11528-018-0292-7>
- CIEB, Centro de Inovação para a Educação Brasileira. (2019). Currículo de Referência em Tecnologia e Computação. Disponível em: <http://curriculo.cieb.net.br/>. Acesso em: 31 mai. 2022.
- Code.org. (2023). Aprenda Ciência da Computação. Mude o mundo. Disponível em: <https://code.org/>. Acesso em: 30 jun. 2023.
- Committee on European Computing Education (2017). Informatics Education in Europe: Are We All in the Same Boat? Technical Report. *Association for Computing Machinery*, New York, NY, USA. <https://doi.org/10.1145/3106077>
- Crompton, H. (2017). ISTE Standards for Educators: A Guide for Teachers and Other Professionals. *Teaching & Learning Faculty Books*. 24. https://digitalcommons.odu.edu/teachinglearning_books/24
- Crompton, H., & Sykora, C. (2021). Developing instructional technology standards for educators: A design-based research study. *Computers and Education Open* v. 2, <https://doi.org/10.1016/j.caeo.2021.100044>
- Dagienė, V., Jevsikova, T., Stupurienė, G., & Juškevičienė, A. (2022). Teaching computational thinking in primary schools: Worldwide trends and teachers' attitudes. *Computer Science and Information Systems*, 19(1), 1-24. <https://doi.org/10.2298/CSIS201215033D>
- García-Peñalvo, F. J., & Mendes, A. J. (2018). Exploring the computational thinking effects in pre-university education. *Computers in Human Behavior*, 80, 407-411. <https://doi.org/10.1016/j.chb.2017.12.005>
- Google for Education. (2023). Every student deserves the chance to explore, advance, and succeed in computer science. Disponível em: https://edu.google.com/intl/ALL_br/code-with-google/. Acesso em: 27 mai. 2023.
- Hour of Code. (2023). Celebre a ciência da computação onde quer que esteja!. Disponível em: <https://hourofcode.com/pt/pt>. Acesso em: 27 jun. 2023.
- Informatics for All. (2023). The Informatics for All Coalition. Disponível em: <https://www.informaticsforall.org/>. Acesso em: 20 jun. 2023.
- ISTE. International Society for Technology in Education. (2023). The ISTE Standards. Disponível em: <https://iste.org/iste-standards> acesso em: 30 jun. 2023.
- Kohler, L., Mattos, M., Lopes, M., Fronza, L., Silveira, H., Fibrantz, G., Rosa, V., & Son, L. (2021). Análise dos Resultados de um Estudo de Caso Aplicando Pensamento Computacional no Ensino Fundamental com Foco na Produção de Algoritmos. In *Anais do XXVII Workshop de Informática na Escola*, (pp. 106-115). Porto Alegre: SBC. doi:10.5753/wie.2021.218645
- Leôncio, N. N., de Sousa, C. C., de Sousa, R. P., & de Melo, R. F. (2017). Programação em blocos com o Mit App Inventor: Um relato de experiência com alunos do ensino médio. In *Anais do Workshop de Informática na Escola* (Vol. 23, No. 1, pp. 1159-1163).
- Mattar, J., e Ramos, D. K. (2021). Metodologia da Pesquisa em Educação: abordagens qualitativas, quantitativas e mistas. 1. Ed. – São Paulo: Edições 70.

- Medeiros, G. A. S., Von Wangenheim, C. G., & Hauck, J. C. R. (2021). O protagonismo de estudantes da Educação Básica a partir do desenvolvimento de aplicativos para smartphone. *Perspectiva*, 39(1), 1-18. <https://doi.org/10.5007/2175-795X.2021.e66049>.
- MIT App Inventor. Disponível em: <https://appinventor.mit.edu/>. Acesso em: 25 jun. 2023.
- OECD. (2020). *What Students Learn Matters: Towards a 21st Century Curriculum*, OECD Publishing, Paris. <https://doi.org/10.1787/d86d4d9a-en>.
- Patton, E.W., Tissenbaum, M., Harunani, F. (2019). MIT App Inventor: Objectives, Design, and Development. In: Kong, SC., Abelson, H. (eds) *Computational Thinking Education*. Springer, Singapore. https://doi.org/10.1007/978-981-13-6528-7_3
- Programaê. (2023). Disponível em: <https://programae.org.br/> acesso em: 20 jun. 2023.
- Rodrigues, A. K. M., Silva, A. P. M., & Carneiro, M. G. (2022). Ensino de pensamento computacional para alunos do ensino básico usando Computação Desplugada e Scratch. *Revista Em Extensão*, 20(2), 228–240. <https://doi.org/10.14393/REE-v20n22021-62305>
- Santana, B., & Santos, J. (2021). Pensamento Computacional para alunos do ensino básico do sertão baiano. In *Anais do XXVII Workshop de Informática na Escola*, (pp. 01-10). Porto Alegre: SBC. doi:10.5753/wie.2021.217174
- Santana, S., & Oliveira, W. (2019). Desenvolvendo o Pensamento Computacional no Ensino Fundamental com o uso do Scratch. In *Anais do XXV Workshop de Informática na Escola*, (pp. 158-167). Porto Alegre: SBC. doi:10.5753/cbie.wie.2019.158
- SBC, Sociedade Brasileira de Computação. (2019a). Diretoria de Educação Básica. Disponível em: <http://www.sbc.org.br/educacao/diretoria-de-educacao-basica>. Acesso em: 16 jun. 2022.
- SBC, Sociedade Brasileira de Computação. (2019b). Diretrizes para ensino de Computação na Educação Básica. Disponível em: <https://www.sbc.org.br/documentos-da-sbc/send/203-educacao-basica/1220-bncc-em-itinerario-informativo-computacao-2>. Acesso em: 16 jun. 2022.
- Silva, R., Souza Oliveira, F. C., Silva Martins, D. J., & Almeida Brito, J. (2021). Uma abordagem lúdica no ensino de pensamento computacional para crianças. *RENOTE*, 19(1), 543–553. <https://doi.org/10.22456/1679-1916.118546>
- Von Wangenheim, C. G., Hauck, J. C., Demetrio, M. F., Pelle, R., da Cruz Alves, N., Barbosa, H., & Azevedo, L. F. (2018). CodeMaster - Automatic Assessment and Grading of App Inventor and Snap! Programs. *Informatics in Education*, 17(1), 117-150.
- Werlich, C., Crema, C., Karczinski, A., & Gasparini, I. (2018). Pensamento Computacional no Ensino Fundamental I: um estudo de caso utilizando Computação Desplugada. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, 7(1), 719. doi:<http://dx.doi.org/10.5753/cbie.wcbie.2018.719>
- Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, 25(2), 7-14. doi: 10.17471/2499-4324/922
- Wolber, D., Abelson, H., Spertus, E., Looney, L. (2014). *App Inventor 2: Create Your Own Android Apps*. 2nd edition, O'Reilly Media, Inc.
- World Economic Forum. (2016). The future of jobs: Employment, skills and workforce strategy for the fourth industrial revolution. Global Challenge Insight Report.