

Desenvolvendo o Pensamento Computacional por Meio de um Jogo Interativo de Lógica de Programação

Anderson Lima dos Santos¹, Claudia P. Pereira^{1,2}

¹Engenharia de Computação – Universidade Estadual de Feira de Santana (UEFS)
Caixa Postal 44.036-900 – Feira de Santana – BA – Brasil

²Programa de Pós-Graduação em Ciência da Computação (PGCC)
Universidade Estadual de Feira de Santana (UEFS) - Feira de Santana, BA - Brasil

{alsantos}@ecomput.uefs.br

{caupinto.sena}@gmail.com

Abstract. *This article presents PaintRobot, a digital game developed to foster computational thinking (CT) in children aged 9 to 12. Based on the BNCC and principles recognized in the literature, the game uses drag-and-drop commands, loops, and conditionals, with progressive feedback. A WEB version is available at itch.io and is in the process of being published on the Play Store. Preliminary validation involved 28 elementary school students, with pre- and post-tests and a learning perception questionnaire. The results indicated understanding of algorithms and patterns, as well as high engagement, suggesting PaintRobot's potential as a playful, motivating, and promising educational tool.*

Resumo. *Este artigo apresenta o PaintRobot, um jogo digital desenvolvido para estimular o Pensamento Computacional (PC) em crianças de 9 a 12 anos. Baseado na BNCC e em princípios reconhecidos na literatura, o jogo utiliza comandos de arrastar e soltar, laços e condicionais, com feedback progressivo. Uma versão WEB está disponível no itch.io e encontra-se em processo de publicação na Play Store. A validação preliminar envolveu 28 estudantes do ensino fundamental, com pré e pós-testes e questionário de percepção de aprendizagem. Os resultados indicaram compreensão de algoritmos e padrões, além de alto engajamento, sugerindo potencial do PaintRobot como ferramenta educativa lúdica, motivadora e promissora.*

1. Introdução

O Pensamento Computacional (PC) tem ganhado destaque na educação como uma competência essencial para as novas gerações. Inicialmente introduzido em [Papert 1980], o conceito ganhou visibilidade com novas abordagens, incluindo discussões apresentadas em [Wing 2006]. Com o avanço das tecnologias e a integração de ferramentas digitais na sociedade, o PC passou a ser incorporado aos currículos escolares em diversos países [Kafai e Proctor 2022, Liu 2023].

No Brasil, a Base Nacional Comum Curricular (BNCC) [Brasil 2017] e seu complemento de 2022 [Brasil 2022] reconhecem o PC como eixo central da educação em computação. Nesse contexto, diferentes estratégias têm sido exploradas para seu ensino, como a computação desplugada [Brackmann et al. 2017], a robótica educacional [de Mattos Vogel e Pereira 2023, Silva e Bertoni 2023] e o uso de jogos educativos

[Kutay e Oner 2022, Ch'ng et al. 2019]. Os jogos digitais têm se mostrado eficazes no ensino de PC, por promoverem motivação, engajamento e melhor desempenho em diversas áreas [Kishimoto 1995, Gee 2003, Al-Tarawneh 2016, Nadeem et al. 2023].

Inspirado nos trabalhos apresentados em [de Mattos Vogel e Pereira 2023, Silva e Bertoni 2023], este estudo propõe o desenvolvimento e a avaliação de um jogo digital — o *PaintRobot* — como ferramenta para estimular habilidades de PC entre estudantes de 9 a 12 anos. Dentre as habilidades, espera-se que o *PaintRobot* seja capaz de estimular a abstração, a decomposição, o reconhecimento de padrões e o conceito de algoritmos, conforme definidos na BNCC, além de servir como material didático de apoio docente. Uma validação preliminar foi realizada para observar o potencial educacional do jogo, indicando avanços especialmente nos conceitos de algoritmos e padrões, embora aspectos mais abstratos tenham apresentado evolução limitada, o que aponta caminhos claros para aprimoramentos futuros.

Além desta introdução, a Seção II apresenta a fundamentação teórica; a Seção III descreve os procedimentos metodológicos; a Seção IV analisa os resultados obtidos; e a Seção V discute as considerações finais e perspectivas futuras.

2. Fundamentação Teórica

2.1. Pensamento Computacional

O Pensamento Computacional (PC) tem ganhado cada vez mais destaque na educação, sendo discutido por diversos autores ao longo das últimas décadas. Sua origem remonta a Seymour Papert, que propôs o conceito de "Pensamento Procedural" como forma de desenvolver habilidades cognitivas por meio da programação de computadores [Papert 1980]. A proposta de Papert indicava que tais habilidades poderiam influenciar a forma de pensar, mesmo fora do contexto computacional. Posteriormente, Jeannette Wing popularizou o termo "Pensamento Computacional", defendendo-o como uma habilidade fundamental para todos, comparável à leitura, escrita e aritmética [Wing 2006]. Em atualização posterior, a autora definiu PC como o processo mental envolvido na formulação de problemas e suas soluções, de maneira que possam ser representadas e executadas por um agente computacional [Wing 2011].

Embora ainda não exista um consenso absoluto sobre sua definição, há elementos amplamente aceitos. Em [The Royal Society 2012] enfatiza-se o reconhecimento de aspectos computacionais no mundo real, enquanto em [Aho 2012] foca-se na formalização de problemas e suas soluções via algoritmos. Em [Grover e Pea 2013] o conceito é expandido ao incluir abstração, modularização, lógica condicional, fluxo de controle e depuração sistemática.

Entre as tentativas de sistematizar o conceito, destaca-se a proposta apresentada em [Brackmann et al. 2017], que organiza o PC em quatro pilares principais: *decomposição*, *reconhecimento de padrões*, *abstração* e *design de algoritmos*. A decomposição refere-se à capacidade de dividir problemas complexos em partes menores e mais manejáveis. O reconhecimento de padrões envolve identificar semelhanças em dados ou processos, o que facilita a generalização de soluções. A abstração permite focar nos aspectos essenciais do problema, ignorando detalhes irrelevantes. Já o design de algoritmos corresponde à criação de sequências ordenadas de instruções para resolver problemas de forma eficiente e sistemática.

No cenário brasileiro, a Base Nacional Comum Curricular (BNCC) reconhece o PC como uma competência essencial, envolvendo habilidades como compreender, modelar, automatizar e resolver problemas de maneira lógica e estruturada [Brasil 2017]. A atualização de 2022 [Brasil 2022] especifica objetivos de aprendizagem para os níveis de ensino infantil e fundamental, destacando competências como a criação de algoritmos, a identificação de padrões, o uso de lógica condicional e a decomposição de problemas complexos.

Neste trabalho, adota-se a abordagem de Brackmann [Brackmann et al. 2017] em conjunto com as diretrizes da BNCC como base teórica. Essa combinação oferece um alicerce conceitual sólido para orientar o desenvolvimento de estratégias pedagógicas que promovam o PC no contexto da educação básica.

2.2. Aprendizado baseado em jogos

O aprendizado baseado em jogos (GBL - *Game-Based Learning*) é uma abordagem que utiliza jogos como ferramenta pedagógica, com o objetivo de promover o engajamento dos estudantes, estimular a motivação intrínseca e facilitar a construção do conhecimento de forma lúdica e significativa. Essa metodologia tem ganhado espaço no contexto educacional por proporcionar um ambiente de aprendizagem envolvente e interativo [Plass et al. 2015].

Em [Kishimoto 1995], argumenta-se que o jogo é essencial para o desenvolvimento cognitivo, social e emocional das crianças, pois estimula habilidades como resolução de problemas, cooperação e tomada de decisões. Destaca-se que o jogo deve ser planejado e inserido intencionalmente no currículo escolar, indo além de uma atividade meramente recreativa.

Com o avanço das tecnologias digitais, o potencial educativo dos jogos foi ampliado. Em [Prensky 2001], afirma-se que jogos bem projetados despertam o interesse dos alunos ao reunir elementos como desafio, curiosidade e controle, favorecendo uma aprendizagem ativa. Em [Gee 2003], ressalta-se que jogos criam contextos autênticos nos quais os estudantes podem explorar conceitos, errar e tentar novamente, promovendo uma aprendizagem mais profunda. [Qian e Clark 2016] apontam que os jogos podem possibilitar melhorias significativas no desempenho acadêmico, motivação e retenção do conhecimento. Em [Dahalan et al. 2024], discute-se que a integração de elementos de jogos no ensino contribui para o envolvimento dos estudantes e exige a formação adequada dos professores para uma aplicação efetiva.

No ensino de PC, os jogos também se mostram eficazes. [Kafai e Burke 2015] destacam que, ao incorporarem princípios de programação e resolução de problemas, os jogos permitem aos alunos praticar habilidades como decomposição, reconhecimento de padrões, abstração e criação de algoritmos de forma intuitiva e divertida. Em [Theodoropoulos e Lepouras 2020], reforça-se que a interatividade e o engajamento dos jogos digitais favorecem o aprendizado de conceitos fundamentais do PC.

2.3. Trabalhos correlatos

Os trabalhos correlatos, apresentados nesta seção, exploram diferentes abordagens para o desenvolvimento do PC, com foco em jogos digitais e interfaces tangíveis. Esses estudos contribuem para compreender como diferentes tipos de jogos e tecnologias podem

promover habilidades ligadas ao PC, destacando estratégias aplicáveis ao contexto educacional.

[Kutay e Oner 2022] investigaram o desenvolvimento do PC em estudantes do ensino médio por meio da programação no jogo *Minecraft*. As atividades envolveram práticas como experimentação, teste, depuração, abstração e modularização. Os resultados, obtidos por meio de testes e entrevistas, indicaram melhora significativa na compreensão e aplicação dos conceitos de PC, demonstrando o potencial do jogo na promoção dessas habilidades.

Em [Corrêa 2017], avaliou-se o impacto do jogo *Counter-Strike: Global Offensive* no desenvolvimento do PC. A pesquisa revelou que, apesar da popularidade do jogo, não houve correlação significativa entre o hábito de jogar e o desenvolvimento das habilidades de PC. O estudo destaca que nem todo jogo eletrônico favorece esse desenvolvimento, sendo necessário que o jogo proponha desafios que envolvam raciocínio lógico e planejamento estratégico.

Em [Nipo et al. 2022], analisaram-se jogos de entretenimento como *The Lost Vikings*, *Overcooked* e *Desperados*, observando sua relação com os pilares do PC: decomposição, reconhecimento de padrões, abstração e algoritmos. Os autores destacam como cada jogo contribui para o desenvolvimento dessas habilidades, sugerindo que, mesmo não sendo desenvolvidos com fins educativos, jogos de entretenimento podem ser aproveitados pedagogicamente com mediação adequada.

Por fim, [de Mattos Vogel e Pereira 2023] e [Silva e Bertoni 2023] propuseram estudos sobre o uso de interface tangível para o ensino de programação de robôs, utilizando blocos físicos que representam comandos de movimento, laço e condicional. Os testes com crianças entre 8 e 9 anos indicaram avanço no entendimento de algoritmos e boa aceitação da proposta. Essa abordagem serviu de inspiração para o desenvolvimento do PaintRobot, que transpõe a lógica dos blocos físicos para um ambiente digital, mantendo o foco pedagógico na estruturação algorítmica.

Além desses trabalhos, destaca-se o jogo educativo *Lightbot*¹, amplamente utilizado na introdução à lógica de programação. O jogo aborda conceitos como sequência, procedimentos e laços por meio de desafios progressivos que exigem planejamento algorítmico. Também é pertinente mencionar plataformas voltadas ao ensino de programação e do PC, como *Scratch*², *CodeMonkey*³ e os ambientes do *Code.org*⁴. Embora ofereçam interfaces visuais baseadas em blocos e permitam explorar conceitos de forma ampla e criativa, funcionam como ambientes abertos, exigindo maior autonomia do estudante na formulação de problemas e na construção de soluções. Assim, diferem de jogos educativos estruturados, nos quais o aprendizado ocorre por meio de desafios curtos e totalmente gamificados.

Nesse cenário, o PaintRobot busca se destacar ao incorporar funcionalidades específicas para aprofundar o aprendizado e suprir limitações observadas em jogos similares, como o *Lightbot*. Um diferencial chave é o comando “Condicional”, que introduz

¹<https://lightbot.com/>

²<https://scratchbrasil.org.br/>

³<https://www.codemonkey.com/>

⁴<https://code.org/>

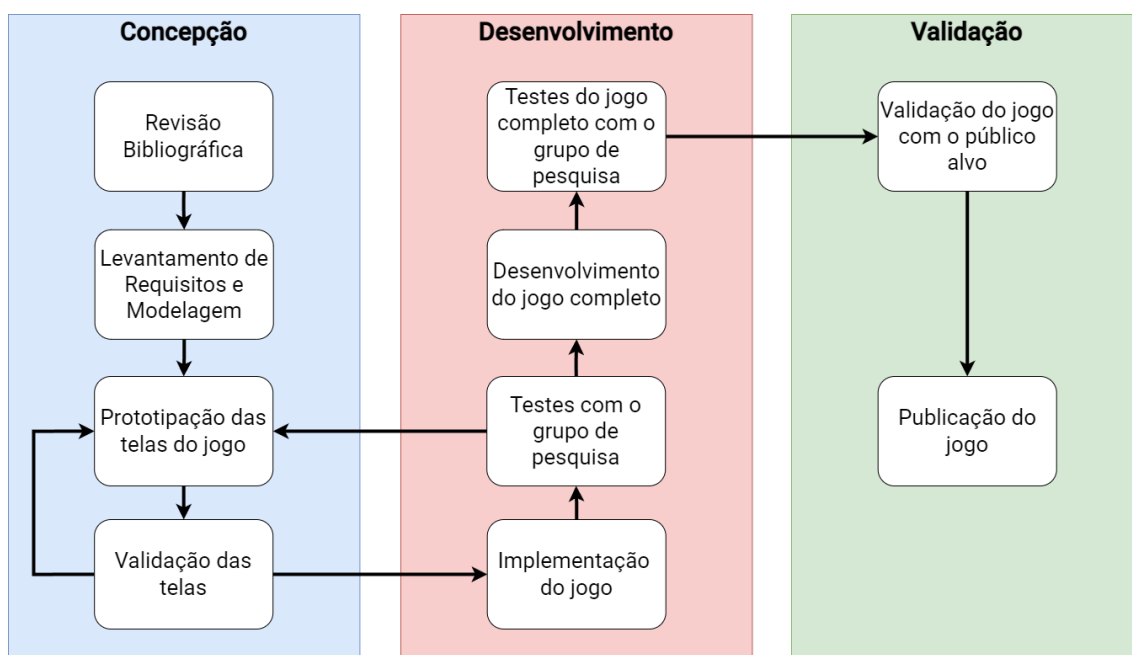


Figura 1. Desenvolvimento do produto

de forma prática a lógica condicional e a noção de verdadeiro ou falso; os jogadores devem utilizá-lo para que o robô reaja a elementos dinâmicos das fases, como pisos que desaparecem, verificando condições para evitar erros e avançar.

Adicionalmente, o PaintRobot oferece um sistema de *feedback* personalizado ao final de cada fase: caso o jogador não conclua o desafio, recebe mensagens que indicam possíveis erros; se bem-sucedido, seu desempenho é avaliado com uma pontuação em estrelas baseada na eficiência do algoritmo, incentivando soluções mais otimizadas. Por fim, o jogo introduz laços do tipo *for*, diferentemente de abordagens baseadas apenas em procedimentos, aproximando o estudante de estruturas comuns em linguagens contemporâneas de programação. Tais características representam aspectos centrais do PaintRobot no contexto de jogos educativos para o PC.

3. Metodologia

Este trabalho originou-se no programa Monitoria de Algoritmos e Programação (MAP) da Universidade Estadual de Feira de Santana (UEFS) em 2023, com a proposta de criar um jogo digital para o ensino de pensamento computacional, inspirado na abordagem tangível de [de Mattos Vogel e Pereira 2023] e [Silva e Bertoni 2023]. O PaintRobot foi concebido como um jogo sério com gráficos isométricos para dispositivos móveis, visando um aprendizado lúdico e acessível. Suas características incluem interface amigável, abordagem de diversos conceitos de programação e *feedback* intuitivo. Adotou-se uma metodologia ágil com entregas iterativas ao longo de um ano, permitindo flexibilidade e ajustes contínuos baseados em *feedbacks* [Sommerville 2011]. O desenvolvimento seguiu três etapas: concepção (pesquisa, requisitos, prototipação), desenvolvimento (implementação, testes) e validação (pré/pós-testes, sessões com estudantes, questionários). Um resumo das etapas seguidas é apresentado na Figura 1.

3.1. Concepção

A concepção iniciou-se com uma revisão bibliográfica para definir os conceitos de Pensamento Computacional (PC) a serem incorporados ao jogo. Os quatro pilares de [Brackmann et al. 2017] (decomposição, reconhecimento de padrões, abstração e algoritmos) e os objetivos de aprendizagem da BNCC [Brasil 2022] foram referências centrais. Realizou-se o levantamento de requisitos funcionais e não funcionais, e utilizou-se o *Unified Game Canvas* (UGC) [Sarinho 2017] para modelar os elementos do jogo. Protótipos das interfaces, desenvolvidos no Canva, foram validados iterativamente com o grupo de pesquisa, e um formulário de avaliação inicial foi aplicado a estudantes de Engenharia de Computação nessa etapa. Assim, o PaintRobot foi concebido como um jogo educativo para estimular o PC em crianças, com mecânica de arrastar e soltar comandos para controlar um robô pintor em fases progressivas, visando desenvolver habilidades como laços e condicionais.

3.2. Desenvolvimento

O jogo foi desenvolvido na *engine Unity* com a linguagem C#. O processo iterativo envolveu validações contínuas pelo grupo de pesquisa. Elementos visuais, como o *sprite* do robô (obtido de recursos gratuitos), foram inicialmente prototipados no Canva e refinados no Adobe Illustrator⁵. As mecânicas centrais (arraste de comandos, movimentação do robô) e as interfaces (menus, telas de ajuda) foram implementadas, juntamente com as fases jogáveis de dificuldade progressiva. O jogo foi compilado para Android (APK) e Windows (EXE).

3.3. Validação

A validação utilizou um teste de PC (pré e pós-teste) para avaliar a eficácia do jogo e um questionário de experiência do usuário para percepções qualitativas.

Para mensurar o PC, aplicou-se o teste de [Román-González et al. 2015], adaptado por [Brackmann et al. 2017]. Do total de 28 questões originais, foram selecionadas 5 com base em três critérios: (i) cobertura dos pilares do PC [Brackmann et al. 2017], garantindo variedade entre decomposição, reconhecimento de padrões, abstração e algoritmos; (ii) compatibilidade com os conceitos trabalhados no PaintRobot, incluindo laços e condicionais; e (iii) dificuldade progressiva, adequada ao tempo disponível para aplicação. As questões selecionadas (1, 3, 6, 8 e 14 da versão original, renumeradas de 1 a 5) apresentavam formatos com comandos simples (setas) ou blocos lógicos similares ao Scratch. O teste completo encontra-se no zenodo.

Para **validar a experiência do usuário**, aplicou-se um questionário que avaliava a percepção dos usuários sobre o jogo e os conceitos de PC. Foram incluídas quatro questões fechadas sobre os pilares do PC (decomposição, reconhecimento de padrões, abstração, algoritmos), com respostas em escala visual de cinco emojis, e três perguntas abertas sobre pontos positivos, pontos de melhorias e comentários adicionais. O instrumento completo está em zenodo.

O experimento foi conduzido pelo pesquisador do projeto com alunos voluntários do 5º e 6º anos (10-13 anos) de um colégio local, durante o horário de aula em dois dias

⁵<https://www.adobe.com/br/products/illustrator.html>

consecutivos. No primeiro dia, após introdução ao Pensamento Computacional e uma dinâmica lúdica de engajamento, os alunos realizaram o pré-teste. No segundo dia, exploraram o PaintRobot em grupos e participaram de uma discussão coletiva sobre os conceitos abordados, finalizando com o pós-teste e o questionário de experiência do usuário. Todo o fluxo foi planejado para integrar exploração do jogo, reflexão guiada e avaliação dos conceitos de PC, garantindo coerência pedagógica e ética no procedimento. O material completo do experimento pode ser encontrado em zenodo. A análise dos dados foi realizada no Jamovi⁶, e o projeto seguiu os protocolos do Comitê de Ética, incluindo sua submissão em agosto de 2024.

4. Resultados

Apresentam-se os resultados do desenvolvimento e avaliação do jogo educativo **PaintRobot**, concebido para estimular o pensamento computacional (PC).

4.1. Requisitos e Planejamento

O desenvolvimento do jogo foi guiado por requisitos funcionais que incluíram: uma interface intuitiva para crianças, um editor de comandos com funcionalidade de arrastar e soltar (contemplando comandos básicos, laços e condicionais), a execução visual dos algoritmos criados e o fornecimento de *feedback* claro durante o jogo. As fases foram projetadas com desafios progressivos e tutoriais, incentivando a aplicação prática de estruturas de programação. Requisitos não funcionais focaram na compatibilidade com Android e Windows e na realização de testes com o público-alvo. A lista detalhada de requisitos pode ser acessada em zenodo.

A concepção do jogo foi estruturada utilizando o *Game Design Canvas* (GDC) [Sarinho 2017], que ajudou a definir o público-alvo (estudantes a partir de 8 anos), as mecânicas centrais (controlar um robô com comandos para pintar pisos), o estilo visual (isométrico) e a progressão dos desafios. Inspirado na mecânica do LightBOT⁷, o PaintRobot foi idealizado como um *puzzle* para o ensino de lógica e algoritmos. O GDC completo está disponível em zenodo.

4.2. Descrição do Jogo, Protótipos e Interfaces

O PaintRobot é um jogo educativo que visa ensinar PC de forma interativa, através do qual o jogador controla um robô pintor. As mecânicas centrais incluem comandos de movimentação (“Andar”, “Girar”), repetição (“Laço”) e lógica (“Condicional”), estruturas diretamente derivadas dos pilares de algoritmos e lógica condicional definidos em [Brackmann et al. 2017] e [Brasil 2022]. Este último é um diferencial, permitindo ao jogador lidar com obstáculos dinâmicos, como pisos que desaparecem, exigindo a verificação de condições para avançar. O jogo, compatível com *Android* e *Windows*, possui fases progressivas (“Básico”, “Laços”, “Condicionais”), tutoriais e pontuação por estrelas. Inspirado também no Coding4Carrots⁸, o PaintRobot se destaca pelo uso prático do comando “Condicional” e pelo *feedback* personalizado ao final de cada fase.

⁶<https://www.jamovi.org/>

⁷<https://lightbot.com/>

⁸<https://doodles.google/doodle/celebrating-50-years-of-kids-coding/>

Protótipos iniciais foram validados com 15 estudantes de Engenharia de Computação da Universidade Estadual de Feira de Santana (BA), resultando em melhorias, especialmente na clareza do comando “Condicional” através da adição de elementos dinâmicos nas fases. O formulário e as respostas estão disponíveis em zenodo.

As interfaces finais seguem um fluxo intuitivo: tela inicial, seleção de conteúdo e fases, além da tela principal do jogo, mostradas respectivamente na Figura 2. Através da interface principal, os jogadores montam algoritmos. As fases foram alinhadas aos objetivos de aprendizagem de PC da BNCC. A relação dos objetivos da BNCC em relação a cada fase do jogo está disponível em zenodo.

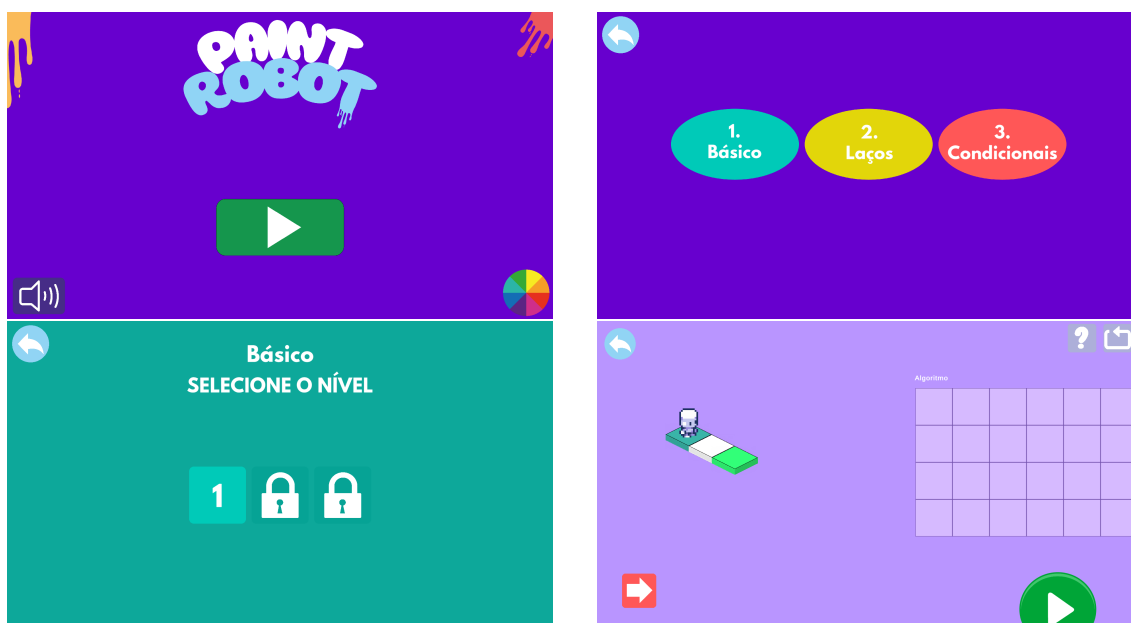


Figura 2. Interfaces do PaintRobot: tela inicial, seleção de conteúdo, seleção de nível e interface principal do jogo.

A proposta do jogo PaintRobot é simples: os jogadores arrastam comandos para formar algoritmos que completem os desafios de cada fase, trabalhando assim conceitos de PC através da lógica empregada. Uma versão WEB do jogo está disponível no itch.io, e o jogo encontra-se em processo de publicação na Google Play Store.

Como exemplo, uma fase da seção “Laços” desafia o jogador a guiar um robô por uma linha de sete pisos usando os comandos “Andar” e “Laço”. Esta atividade explora diversas competências do PC, como o reconhecimento de padrões (ao identificar a necessidade de repetir a ação “Andar”), a criação e simulação de algoritmos com repetições (usando o comando “Laço” para otimizar a solução), a organização de etapas de forma clara (ao construir o algoritmo), e a decomposição de problemas maiores em partes menores. O jogo também incentiva a análise da eficiência de diferentes abordagens (comparando o uso repetitivo de “Andar” versus o uso do “Laço”) e a correção de falhas em algoritmos. A Figura 3 ilustra um exemplo de desafio da seção “Laços” (a), sua solução ideal (b) e uma alternativa (c), evidenciando as duas principais estratégias algorítmicas para o desafio: a repetição manual do comando “Andar” sete vezes e o uso otimizado do comando “Laço” para executar a mesma ação.

Ao final de cada tentativa, o jogador recebe um *feedback* personalizado. Um algoritmo incorreto resulta em uma mensagem de erro, enquanto uma solução correta é avaliada com uma a três estrelas, dependendo da quantidade de comandos utilizados em relação ao ideal para aquela fase. Este sistema de *feedback*, exemplificado na Figura 4, visa reforçar o aprendizado: ao errar, o jogador recebe indicações e pode tentar novamente, criando um ciclo de experimentação que favorece uma aprendizagem mais profunda [Gee 2003], enquanto a pontuação por estrelas sustenta o engajamento por meio do desafio e do controle sobre a solução [Prensky 2001]. As soluções ideais para todas as fases estão disponíveis em zenodo.

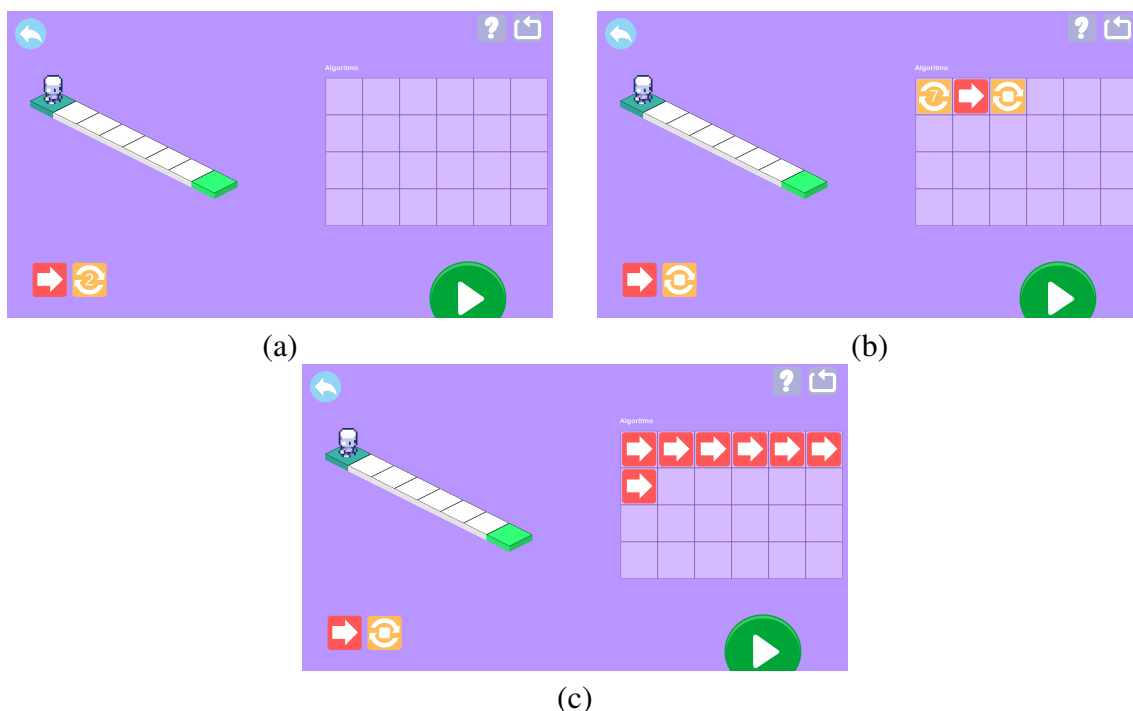


Figura 3. Exemplo da segunda fase da seção “Laços”: desafio (a), solução ideal (b) e solução alternativa (c).

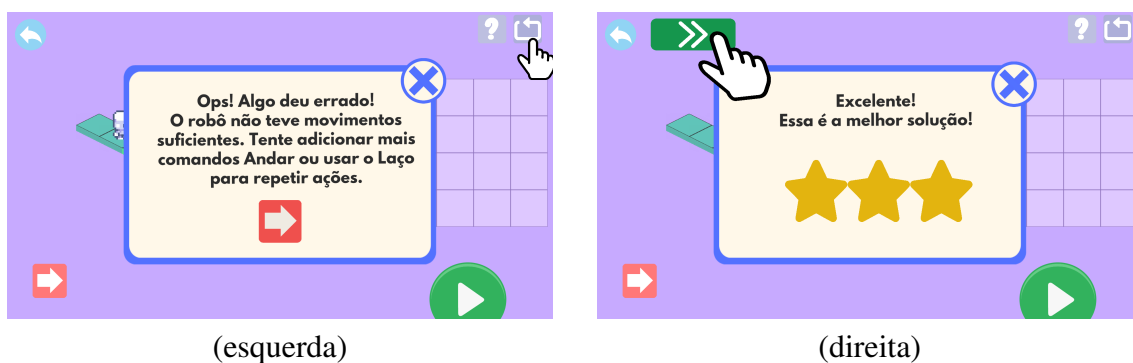


Figura 4. Feedback no jogo: mensagem para algoritmo incorreto (esquerda) e avaliação com estrelas para algoritmo correto (direita).

4.3. Validação do Jogo

A validação envolveu 28 crianças do 5º e 6º anos (entre 10–12 anos, com 57,1% aos 11 anos e distribuição de gênero equitativa) de um Colégio do interior do estado da Bahia. Utilizou-se o Teste de Pensamento Computacional adaptado em [Brackmann et al. 2017] (pré e pós-teste) e um questionário de experiência do usuário.

4.3.1. Estímulo do Pensamento Computacional

No pré-teste, 100% dos alunos acertaram a primeira questão, mas a quarta questão teve apenas 14,3% de acertos. No pós-teste, houve melhora nas questões 2 (de 42,9% para 53,6% de acertos) e 4 (de 14,3% para 21,4%). O Teste Exato de Fisher [Fisher 1922], analisado com Jamovi, indicou diferenças estatisticamente significativas ($p < 0,05$) para as questões 2, 3 e 5, mas não para a questão 4 ($p = 0,549$). A média geral de acertos aumentou levemente de 2,89 para 2,96. O impacto limitado pode ser atribuído ao curto tempo de intervenção (cerca de 30 minutos de jogo), o que impossibilita inferir conclusivamente qualquer ganho efetivo de aprendizado em pensamento computacional, embora os resultados individuais por questão já indicam melhorias do pós teste comparado ao pré teste. Para uma avaliação mais robusta, seriam necessárias sessões mais longas e repetidas ao longo do tempo. As respostas dos participantes estão disponíveis em zenodo.

4.3.2. Avaliação da aprendizagem percebida pelos alunos

Para avaliar a percepção dos alunos sobre sua própria aprendizagem dos conceitos de Pensamento Computacional (PC) durante o uso do PaintRobot, foi aplicado um questionário com quatro afirmações (em escala likert com 5 opções de avaliação), cada uma correspondendo a um dos pilares do PC:

- O jogo ajudou você a entender como dividir a solução de um problema em partes menores (Decomposição).
- O jogo ajudou você a identificar padrões ou semelhanças em problemas diferentes para facilitar a solução (Reconhecimento de Padrões).
- O jogo ajudou você a entender como focar nos detalhes mais importantes de um problema, ignorando informações irrelevantes (Abstração).
- O jogo ajudou você a entender como criar uma sequência de passos ou instruções para resolver um problema de forma eficiente (Algoritmos).

As respostas dos participantes, refletindo sua percepção de aprendizagem, estão apresentadas na Tabela 1 e disponíveis em zenodo.

Tabela 1. Frequência das respostas dos participantes em relação às afirmações sobre PC.

Categoria	Concordo Totalmente	Concordo Parcialmente	Indiferente	Discordo Parcialmente	Discordo Totalmente
Decomposição	18	8	2	0	0
Reconhecimento de Padrões	15	9	4	0	0
Abstração	16	4	7	1	0
Algoritmos	23	5	0	0	0

Observa-se que a maioria dos alunos relatou ter compreendido os conceitos abordados, especialmente Algoritmos (23 respostas “Concordo Totalmente” + 5 “Concordo Parcialmente”) e Decomposição (18 “Concordo Totalmente” + 8 “Concordo Parcialmente”). O Reconhecimento de Padrões também apresentou concordância majoritária, enquanto Abstração teve respostas mais distribuídas entre os níveis de concordância e neutro, indicando menor clareza percebida nesse conceito. É importante ressaltar que Abstração, naturalmente, é um conceito mais complexo, até para idades maiores (de 18 a 20 anos, por exemplo), com jovens que estão iniciando programação introdutória em cursos de computação e estão em estágio superior de desenvolvimento cognitivo.

Embora a intervenção tenha sido breve (~30 minutos de jogo), os relatos indicam que os alunos interagiram de forma mais estratégica com os comandos à medida que exploravam os desafios, especialmente o comando de Laço, inicialmente ignorado ou mal compreendido. Esses dados sugerem que o PaintRobot pode apoiar a percepção de compreensão dos conceitos de PC e estimular a reflexão sobre estratégias algorítmicas, mesmo que não seja possível inferir ganhos efetivos ou duradouros de aprendizagem a partir deste estudo. Para além dos dados quantitativos, as perguntas abertas do questionário revelaram percepções bastante positivas sobre o PaintRobot:

- **O que você mais gostou do jogo?:** Grande parte das crianças destacou a jogabilidade, enfatizando que, apesar de simples, o jogo é muito divertido. Um participante comentou: *“Que tem que ter muito raciocínio para resolver as perguntas”* (E1). Muitos apreciaram o estímulo ao raciocínio lógico para avançar nas fases, como indicado na fala: *“Gostei mais das partes de algoritmos, me lembra muito de jogos que já joguei”* (E5). Outros citaram aspectos específicos, como o robô (*“No design do personagem”* – E13) ou a oportunidade de aprender conceitos de PC como decomposição (*“Gostei de aprender a decomposição”* – E9). Vários expressaram satisfação geral, dizendo que gostaram de tudo.
- **O que poderia ser melhorado no jogo?:** A principal sugestão foi adicionar mais fases, especialmente com maior dificuldade, como apontado: *“Assim eu acho que poderia ter mais fases com níveis mais difíceis”* (E24). Alguns mencionaram melhorias na jogabilidade, como o uso de comandos clicando diretamente neles em vez de arrastá-los: *“Invés de colocar as setas dentro do quadro, o jogador poderia apenas clicar na seta”* (E6). Houve pedidos por uma versão para iPhone (*“Eu gostei, mas poderia ter uma versão para iPhone”* – E15) e personalização: *“Escolher personagem, ter decoração”* (E15). Apesar disso, algumas crianças afirmaram que o jogo já está excelente: *“O jogo está ótimo, para mim não precisa de melhoras”* (E9).
- **Gostaria de fazer mais algum comentário?:** Muitos elogiaram o jogo com termos como *“incrível”* (E14) e *“amei muito”* (E1). Outros pediram novos jogos no mesmo estilo: *“O jogo é um dos melhores que já joguei e gostaria que fizessem mais jogos como este.”* (E4). Alguns optaram por não responder, mas aqueles que comentaram reforçaram o impacto positivo, dizendo: *“Está muito bom o seu jogo, parabéns”* (E15).

De forma geral, os comentários e as respostas dos alunos indicam que o PaintRobot foi bem recebido, sendo percebido como divertido e útil para explorar conceitos de PC. As sugestões fornecem ideias interessantes para futuras melhorias, como fases

mais desafiadoras e ajustes na jogabilidade, reforçando o potencial do jogo para engajar, ensinar e motivar o aprendiz de maneira lúdica.

4.3.3. Impacto Educacional Percebido

Embora a intervenção tenha sido breve, em função também da necessidade de adequação aos horários disponibilizados pela escola e à rotina das crianças, as observações qualitativas indicam que os alunos interagiram de forma mais estratégica com os comandos, especialmente o comando de **Laço**, à medida que exploravam os desafios.

Essas evidências preliminares sugerem que o PaintRobot possui potencial para apoiar a percepção de compreensão de conceitos de PC e estimular reflexão sobre estratégias algorítmicas. Além disso, o uso de feedback por estrelas e fases progressivas permite estudar como crianças ajustam suas soluções, oferecendo um mecanismo replicável para observar o engajamento e experimentação em algoritmos.

Dessa forma, os resultados apontam que jogos educativos com mecânicas de exploração, design iterativo e feedback progressivo podem favorecer o engajamento e a experimentação com estruturas de programação. O PaintRobot, portanto, nesta perspectiva, contribui não apenas como ferramenta educativa, mas também como material didático e objeto de pesquisa para investigar estratégias de ensino de PC. Estudos futuros, com amostras maiores e metodologias controladas, serão necessários para avaliar impactos concretos na aprendizagem e validar essas hipóteses.

5. Conclusão

Este trabalho evidenciou o PaintRobot como uma ferramenta educativa concebida para estimular o pensamento computacional (PC) em crianças, construída com base nos pilares amplamente reconhecidos na literatura sobre PC e alinhada às competências da BNCC. O jogo foi desenvolvido para oferecer uma experiência lúdica, progressiva e acessível, com funcionalidades como *feedback* visual, comandos de repetição e lógica condicional, incentivando a criação de algoritmos mais eficientes. Os relatos e respostas dos alunos indicam que o PaintRobot é percebido como divertido e útil para explorar conceitos de PC, reforçando seu potencial pedagógico.

Apesar do curto período de intervenção durante a validação, que limitou a mensuração de ganhos de aprendizagem de forma robusta, os dados qualitativos sugerem que o jogo conseguiu engajar os alunos e fomentar reflexão sobre estratégias algorítmicas. Para trabalhos futuros, sugere-se a realização de estudos longitudinais, com amostras maiores e a inclusão de grupos-controle, além da ampliação das fases e funcionalidades planejadas, o que permitirá avaliar de forma mais consistente o impacto do PaintRobot no desenvolvimento do pensamento computacional.

Agradecimentos

Agradecemos aos participantes da pesquisa preliminar e ao Colégio Monteiro Lobato pela infraestrutura e pelo tempo cedido para a realização das atividades.

Uso de Inteligência Artificial

Declaro que foram utilizadas ferramentas de Inteligência Artificial Generativa neste trabalho para auxílio na revisão gramatical e verificação de adequação ao formato do evento. A concepção da pesquisa, análise dos dados e escrita final do conteúdo são de responsabilidade integral dos autores.

Referências

- Aho, A. V. (2012). Computation and computational thinking. *The Computer Journal*, 55(7):832–835. Disponível em: <https://doi.org/10.1093/comjnl/bxs074>. Acesso em: 30 de Maio de 2024.
- Al-Tarawneh, M. H. (2016). The effectiveness of educational games on scientific concepts acquisition in first grade students in science. *Journal of Education and Practice*, 7(3):31–37. Disponível em: https://www.researchgate.net/publication/315516554_The_Effectiveness_of_Educational_Games_on_Scientific_Concepts_Acquisition_in_First_Grade_Students_in_Science. Acesso em: 16 de Maio de 2024.
- Brackmann, C. P. et al. (2017). Computational thinking unplugged: Teaching and evaluation in spanish primary education. *15*. Disponível em: <https://doi.org/10.1145/3137065.3137069>. Acesso em: 21 de Março de 2024.
- Brasil (2017). Base nacional comum curricular (BNCC). http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=83875-bncc-20dez-site-pdf&category_slug=dezembro-2017-pdf&Itemid=30192. Acesso em: 21 de Março de 2024.
- Brasil (2022). Normas sobre Computação na Educação Básica – Complemento à BNCC. http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=89123-normas-sobre-computacao-na-educacao-basica-2022-pdf&category_slug=janeiro-2022-pdf&Itemid=30192. Acesso em: 21 de Março de 2024.
- Ch'ng, S. I., Low, Y. C., Lee, Y. L., Chia, W. C., e Yeong, L. S. (2019). Video games: A potential vehicle for teaching computational thinking. In Kong, S. C. e Abelson, H., editors, *Computational Thinking Education*, pages 247—260. Springer, Singapore. Disponível em: https://doi.org/10.1007/978-981-13-6528-7_14. Acesso em: 2 de Maio de 2024.
- Corrêa, E. F. S. (2017). Jogos eletrônicos e o desenvolvimento do pensamento computacional: um estudo de caso. *Universidade de Brasília*, Monografia. Disponível em: <https://bdm.unb.br/handle/10483/21061>. Acesso em: 13 de Junho de 2024.
- Dahalan, F., Alias, N., e Shaharom, M. S. (2024). Gamification and game based learning for vocational education and training: A systematic literature review. *Education and Information Technologies*, 29:1279–1317. Disponível em: <https://doi.org/10.1007/s10639-022-11548-w>. Acesso em: 30 de Maio de 2024.
- de Mattos Vogel, B. G. e Pereira, C. P. (2023). Trabalhando o pensamento computacional através de uma interface tangível para programação de robôs. *Revista Brasileira de*

- Informática na Educação*, 31:117–148. Disponível em: <https://journals-sol.sbc.org.br/index.php/rbie/article/view/2824>. Acesso em: 2 de Maio de 2024.
- Fisher, R. A. (1922). On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94. Disponível em: <https://doi.org/10.2307/2340521>. Acesso em: 14 de Novembro de 2024.
- Gee, J. P. (2003). *What Video Games Have to Teach Us About Learning and Literacy*. Palgrave Macmillan.
- Grover, S. e Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational Researcher*, 42(1):38–43. Disponível em: <https://doi.org/10.3102/0013189X12463051>. Acesso em: 30 de Maio de 2024.
- Kafai, Y. B. e Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational Psychologist*, 50(4):313–334. Disponível em: <https://doi.org/10.1080/00461520.2015.1124022>. Acesso em: 30 de Maio de 2024.
- Kafai, Y. B. e Proctor, C. (2022). A reevaluation of computational thinking in k–12 education: Moving toward computational literacies. *Educational Researcher*, 51(2):146–151. Disponível em: <https://doi.org/10.3102/0013189X211057904>. Acesso em: 21 de Março de 2024.
- Kishimoto, T. M. (1995). *Jogos infantis: o jogo, a criança e a educação*. Vozes, Petrópolis. Acesso em: 16 de Maio de 2024.
- Kutay, E. e Oner, D. (2022). Coding with minecraft: The development of middle school students’ computational thinking. *ACM Trans. Comput. Educ.*, 22(2). Disponível em: <https://doi.org/10.1145/3471573>. Acesso em: 2 de Maio de 2024.
- Liu, Z. L. (2023). Integrating computational thinking into k-12 education: Bridging the gap between theories and practices. *STEM Education Review*. Disponível em: <https://doi.org/10.54844/stemer.2023.0467>. Acesso em: 21 de Março de 2024.
- Nadeem, M., Oroszlanyova, M., e Farag, W. (2023). Effect of digital game-based learning on student engagement and motivation. *Computers*, 12(9):177. Disponível em: <https://doi.org/10.3390/computers12090177>. Acesso em: 16 de Maio de 2024.
- Nipo, D., Rodrigues, R., e França, R. (2022). Jogando e pensando: Aprendendo pensamento computacional com jogos de entretenimento. In *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*, pages 573–584. SBC. Disponível em: <https://sol.sbc.org.br/index.php/sbie/article/view/22441>. Acesso em: 13 de Junho de 2024.
- Papert, S. (1980). *Mindstorms: Children, computers, and powerful ideas*. Basic Books, Inc.
- Plass, J. L., Homer, B. D., e Kinzer, C. K. (2015). Foundations of game-based learning: A synthesis of ideas in key theoretical models. *Educational Psychologist*, 50(4):314–331.

Disponível em: <https://doi.org/10.1080/00461520.2015.1122533>.
Acesso em: 30 de Maio de 2024.

Prensky, M. (2001). *Digital Game-Based Learning*. McGraw-Hill.

Qian, M. e Clark, K. R. (2016). Game-based learning and 21st century skills: A review of recent research. *Computers in Human Behavior*, 63:50–58. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0747563216303491>. Acesso em: 30 de Maio de 2024.

Román-González, M., Pérez-González, J. C., e Jiménez-Fernández, C. (2015). Test de pensamiento computacional: diseño y psicometría general. In *III Congreso Internacional sobre Aprendizaje, Innovación y Competitividad (CINAIC 2015)*, pages 1–6. Acesso em: 27 de Junho de 2024.

Sarinho, V. T. (2017). Uma proposta de game design canvas unificado. *XVI Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 141–148. Disponível em: <https://www.sbgames.org/sbgames2017/papers/ArtesDesignFull/175107.pdf>. Acesso em: 16 de Maio de 2024.

Silva, A. e Bertoni, F. (2023). Interface tangível para o ensino de algoritmos, programação e prática do pensamento computacional. In *XXVII Seminário de Iniciação Científica Da UEFS Semana Nacional de Ciência e Tecnologia*. Acesso em: 25 de Julho de 2024.

Sommerville, I. (2011). *Engenharia de Software*. Pearson, 9 edition.

The Royal Society (2012). Shut down or restart? the way forward for computing in uk schools. Report, The Royal Society, London, UK.

Theodoropoulos, A. e Lepouras, G. (2020). Digital game-based learning and computational thinking in p-12 education: A systematic literature review on playing games for learning programming. In Kalogiannakis, M. e Papadakis, S., editors, *Handbook of Research on Tools for Teaching Computational Thinking in P-12 Education*, pages 159–183. IGI Global.

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35. Disponível em: <https://doi.org/10.1145/1118178.1118215>. Acesso em: 21 de Março de 2024.

Wing, J. M. (2011). Computational thinking: What and why? *The Link Magazine*, pages 20–23. Disponível em: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>. Acesso em: 16 de Maio de 2024.