

# Codebricks: Um RPG Educacional 2D para Dispositivos Móveis no Ensino de Lógica de Programação com Blocos Visuais

Hany Ahmad Omar Shahin<sup>1</sup>, Heder Filho Silva Santos<sup>1</sup>, Marcos Alves Vieira<sup>1</sup>

<sup>1</sup> Instituto Federal Goiano (IF Goiano) - Campus Iporá - Iporá - GO - Brasil  
{hany.ahmad, heder.filho}@estudante.ifgoiano.edu.br,  
marcos.vieira@ifgoiano.edu.br

**Abstract.** *This paper presents the development of “Codebricks”, a 2D educational RPG game designed to teach programming logic to elementary and high school students. Featuring a visual block-based interface inspired by Scratch, the game allows students to build algorithms to control a character in an interactive environment. The proposal aims to make learning more accessible, engaging, and aligned with the preferences of digital natives, with a focus on usability on mobile devices. The project combines elements of gamification and computational thinking to foster engagement and meaningful learning.*

**Resumo.** *Este trabalho apresenta o desenvolvimento do “Codebricks”, um jogo educativo do tipo RPG 2D voltado ao ensino de lógica de programação para estudantes do ensino fundamental e médio. Com uma interface de blocos visuais inspirada no Scratch, o jogo permite que os alunos construam algoritmos para controlar um personagem em um ambiente interativo. A proposta visa tornar o aprendizado mais acessível, envolvente e compatível com as preferências dos nativos digitais, com ênfase na usabilidade em dispositivos móveis. O projeto integra elementos de gamificação e pensamento computacional para promover o engajamento e a aprendizagem significativa.*

## 1. Introdução

O ensino de programação tem ganhado cada vez mais relevância na sociedade contemporânea, não apenas como uma habilidade valorizada no mercado de trabalho, mas também como um instrumento para o desenvolvimento do raciocínio lógico, da criatividade e da capacidade de solucionar problemas de forma estruturada [Albán Bedoya and Ocaña-Garzón 2021]. Esse reconhecimento está alinhado à noção de pensamento computacional, concebida por [Wing 2006] como uma habilidade cognitiva essencial no século XXI, que engloba habilidades como decomposição, abstração, reconhecimento de padrões e construção de algoritmos.

Tais habilidades tornam-se ainda mais relevantes quando estimuladas desde os primeiros anos escolares. Estudos como o de [Su and Yang 2023] demonstram que abordagens lúdicas e digitais, quando bem planejadas, promovem avanços cognitivos significativos mesmo em crianças na faixa etária de 4 a 6 anos. Diante disso, diversas políticas educacionais têm buscado incorporar o ensino de programação aos currículos escolares, como evidenciado na Base Nacional Comum Curricular (BNCC) [Brasil 2018]. Essa diretriz foi aprofundada pelo Parecer CNE/CEB nº 2/2022, que normatiza a Computação na

Educação Básica e reforça a importância de trabalhar eixos como o pensamento computacional ao longo de toda a formação escolar.

Contudo, como [Araújo and Silva 2024] observam, a implementação dessa diretriz ainda enfrenta dificuldades, pois a inserção da computação nas escolas ainda ocorre de forma limitada e, muitas vezes, desconectada da prática pedagógica. Soma-se a isso a dificuldade de abordar conceitos técnicos como variáveis, estruturas de decisão e laços de repetição, especialmente entre estudantes iniciantes, o que, aliado a abordagens acadêmicas excessivamente técnicas, contribui para a desmotivação e eventual abandono do aprendizado [Grover and Pea 2013]. De acordo com [de Souza et al. 2021], é fundamental considerar o contexto social dos estudantes e adotar estratégias mais acessíveis e motivadoras, capazes de dialogar com suas experiências digitais cotidianas.

Nesse cenário, ferramentas de programação por blocos, como o Scratch<sup>1</sup>, têm se mostrado eficazes ao abstrair a complexidade sintática e possibilitar o aprendizado por meio de recursos visuais e interativos [Resnick et al. 2009]. Já [Bers 2020] enfatiza que esse tipo de abordagem deve ser lúdica e exploratória, ressaltando seu potencial para promover tanto competências cognitivas quanto socioemocionais, como autonomia e persistência. Nesse contexto, uma das formas mais exploradas para potencializar esse caráter lúdico e estruturado é o uso de jogos digitais no processo de ensino-aprendizagem.

Jogos digitais vêm se consolidando como estratégias pedagógicas eficazes, por criarem contextos imersivos em que o erro é parte do processo de aprendizagem, favorecendo o engajamento e a construção significativa do conhecimento [Gee 2003]. Estudos recentes, como o de [Souza et al. 2024], indicam que jogos com mecânicas de *Role-Playing Game* (RPG)<sup>2</sup> e recompensas progressivas favorecem o desenvolvimento do pensamento lógico, a retenção de conteúdo e o engajamento prolongado com o aprendizado. Dado que mais de 80% das crianças e adolescentes brasileiros utilizam dispositivos móveis [CGI.br/NIC.br 2021], torna-se essencial criar ferramentas educacionais compatíveis com essa realidade, como destaca [Raabe and de Araújo Cavalcante 2024].

Nesse contexto, este trabalho apresenta o “Codebricks”, um jogo de RPG em duas dimensões (2D) educativo para celulares, voltado ao ensino de lógica de programação com blocos visuais. Os objetivos incluem: (i) projetar uma narrativa gamificada com foco pedagógico; (ii) desenvolver uma interface de programação por blocos adaptada a telas pequenas; e (iii) avaliar a experiência do usuário por meio do Modelo para a Avaliação de Jogos Educacionais (*Model for the Evaluation of Educational Games* - MEEGA+).

O restante deste trabalho está estruturado da seguinte forma: a Seção 2 apresenta uma revisão dos principais trabalhos relacionados, destacando as abordagens existentes para o ensino de programação com blocos visuais e os desafios enfrentados no contexto educacional brasileiro. A Seção 3 descreve detalhadamente o jogo Codebricks, abordando sua mecânica de programação em blocos, estrutura de fases e interface adaptada para dispositivos móveis. Em seguida, a Seção 4 detalha a metodologia adotada para o desenvolvimento do jogo, incluindo o processo de concepção, prototipação, arquitetura técnica e planejamento da avaliação da experiência do usuário. Por fim, a Seção 5 reúne as considerações finais e os próximos passos do projeto, sintetizando os resultados

---

<sup>1</sup><https://scratch.mit.edu/>

<sup>2</sup>RPG é um jogo em que o jogador interpreta um personagem em um mundo com regras e narrativa.

alcançados até o momento e apresentando as direções futuras para a evolução e validação do Codebricks como ferramenta de apoio ao ensino de lógica de programação.

## 2. Trabalhos Relacionados

Diversos estudos têm explorado o uso da programação em blocos para o desenvolvimento do pensamento computacional. Um mapeamento sistemático conduzido por [Sousa et al. 2020] identificou a predominância de ferramentas *web* (59%) e a baixa presença de soluções para dispositivos móveis (5%) entre 2015 e 2019, com foco no ensino fundamental e médio, além de pouca representatividade brasileira (3,4%), o que reforça a necessidade de iniciativas voltadas ao contexto nacional.

No Brasil, destacam-se o projeto Desenvolver para Codificar [Ricarte et al. 2024], que integrou programação visual ao currículo de escolas públicas na Paraíba, e o Scratchim [Rodrigues et al. 2022], que combinou atividades desplugadas ao ensino de programação em blocos. Ambos evidenciam o potencial pedagógico da abordagem mesmo em ambientes com infraestrutura limitada. O ProgramSE [Silva et al. 2021] utilizou metáforas cotidianas em jogos baseados no Scratch, com resultados positivos, mas com limitações quanto à variedade e personalização de conteúdos.

Estudos como de [Perin et al. 2021] e [Geraldes and Afonseca 2024] reforçam o papel de metodologias ativas e jogos educativos na motivação e retenção de estudantes, especialmente em cursos técnicos. Em nível superior, [Josko and de Assis Zampirolli 2023] demonstraram que a gamificação pode reduzir reprovações e aumentar a motivação.

A importância da gamificação como estratégia para o ensino de lógica também é destacada por [Li et al. 2023], cuja meta-análise com mais de 5 mil participantes mostrou impacto positivo significativo, especialmente no ensino fundamental, com melhores resultados em abordagens com narrativa, desafios e *feedback* visual. O apoio docente também é relevante, como destaca [Brezolin et al. 2023] ao recomendar tutoriais e explicações integradas ao jogo.

Em resposta à predominância de soluções *web*, este trabalho apresenta o Codebrick, um RPG 2D educativo para dispositivos móveis. Seus diferenciais residem na combinação de eixos técnicos e pedagógicos: (i) o uso da progressão por fases com desafios crescentes para contextualizar o aprendizado, fazendo da lógica a ferramenta para superar obstáculos; (ii) a programação por blocos como o sistema de controle do personagem, em vez de uma ferramenta de criação, os blocos são a interface direta para comandar as ações e resolver os desafios do jogo; e (iii) uma interface nativa projetada para telas pequenas, que remove barreiras de acesso e se alinha aos hábitos digitais do público-alvo.

## 3. Codebricks

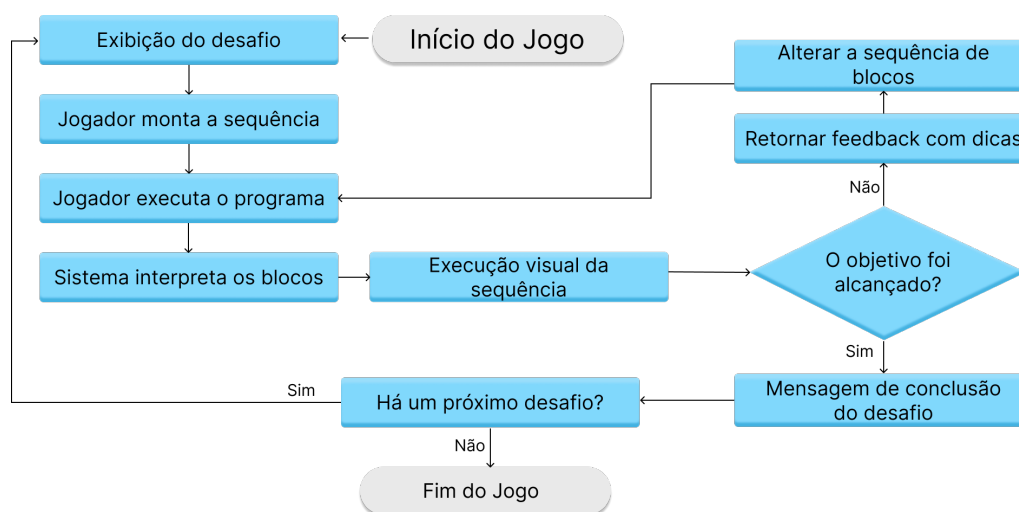
“Codebricks” é um jogo digital educativo do gênero Jogo de Interpretação de Papéis (*Role-Playing Game* - RPG) em duas dimensões (2D), desenvolvido para dispositivos móveis, com o propósito de auxiliar no ensino de lógica de programação para estudantes do ensino fundamental e médio. O jogo foi concebido para proporcionar uma experiência de aprendizado interativa e lúdica, no qual o usuário assume o papel de um personagem que explora um mundo, enfrentando desafios que exigem a aplicação de conceitos de programação para serem superados. A narrativa do jogo guia o estudante através de di-

ferentes fases, cada uma introduzindo novos conceitos e aumentando progressivamente a complexidade dos problemas a serem resolvidos.

### 3.1. Mecânica de Programação em Blocos

A principal mecânica é a interface de programação baseada em blocos visuais, abordada para simplificar a interação em dispositivos móveis com telas menores, eliminando a necessidade de digitação de código e reduzindo a curva de aprendizado inicial associada à sintaxe de linguagens de programação tradicionais. O jogador monta algoritmos em um painel de programação no qual pode arrastar e encaixar blocos que representam comandos específicos, tais como mover para frente, mover para esquerda, coletar item e se (condição) então (ação). A sequência de blocos forma um algoritmo que, ao ser executado, define o comportamento do personagem no cenário do jogo. O sistema oferece retorno visual imediato, permitindo ao jogador acompanhar a execução e identificar falhas lógicas. Essa dinâmica favorece a aprendizagem por tentativa e erro, estimulando o pensamento algorítmico de modo prático e exploratório.

A Figura 1 apresenta um fluxograma representando o funcionamento do sistema de execução no jogo Codebricks. O diagrama ilustra a sequência de etapas desde a apresentação do desafio ao jogador até a verificação do sucesso da tarefa

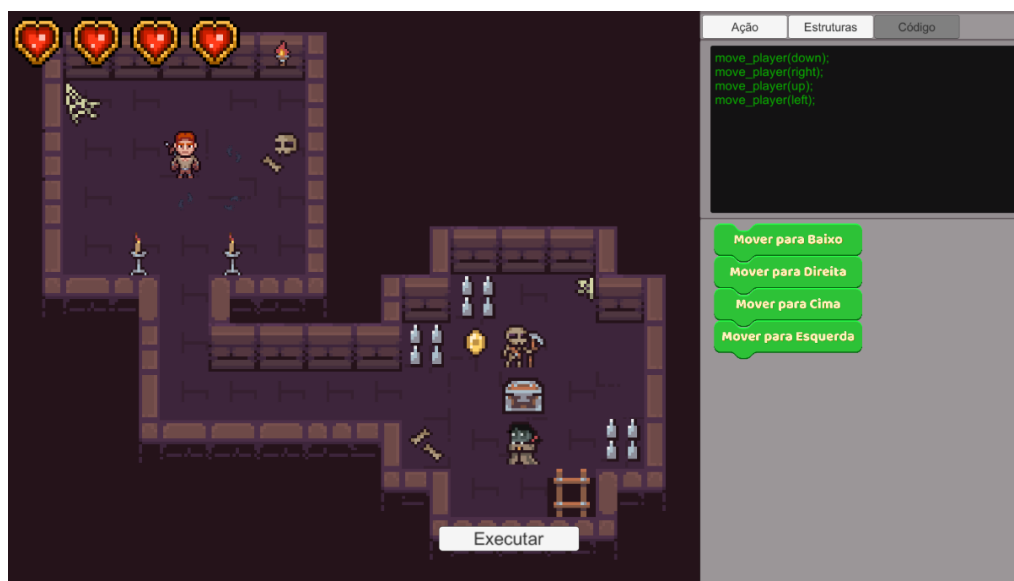


**Figura 1. Fluxograma representando o funcionamento do jogo “Codebricks”.**

Além da montagem por blocos, o sistema oferece uma funcionalidade opcional que exibe, paralelamente, uma tradução textual simplificada da sequência criada (Figura 2). Essa visualização tem como objetivo aproximar o aluno da estrutura das linguagens de programação, facilitando a compreensão dos conceitos subjacentes e incentivando, de forma gradual, a transição para ambientes textuais.

### 3.2. Estrutura de Fases e Progressão

A progressão do jogo é estruturada em fases que introduzem, de forma gradual e crescente em complexidade, os principais conceitos da lógica de programação. Cada fase aborda um único conceito, que deve ser compreendido e aplicado pelo jogador para avançar, promovendo a assimilação por meio da prática em um ambiente interativo.



**Figura 2. Código textual gerado a partir da sequência de blocos**

As duas primeiras fases focam no sequenciamento, utilizando blocos de comandos básicos para movimentar o personagem. Elas introduzem os fundamentos sem recorrer a estruturas de controle.

Em seguida, são trabalhadas estruturas de decisão: uma fase com o condicional “se” (*if*) e outra com “se/senão” (*if/else*), permitindo decisões baseadas em condições lógicas. A lógica de repetição é abordada em fases seguintes: a estrutura “enquanto” (*while*) já está implementada, e a fase com o comando “para” (*for*) encontra-se em estágio final de desenvolvimento.

A Figura 3 ilustra a fase introdutória da estrutura condicional simples. Nela, o jogador deve coletar uma chave para liberar os blocos “Existe\_Chave” e “Abrir\_Porta”, que são usados para abrir os portões e alcançar a escada, objetivo final da fase.

A fase final, que visa reunir os conceitos apresentados ao longo do jogo em desafios mais amplos e integrados, encontra-se atualmente na etapa de planejamento. A proposta é que esse estágio funcione como uma síntese do aprendizado, exigindo a aplicação combinada dos blocos de ação, condicionais e estruturas de repetição.

### 3.3. Interface e Interação em Dispositivos Móveis

Considerando o foco em dispositivos móveis, a interface foi projetada para ser intuitiva e responsiva ao toque. Os blocos de comando são dimensionados para fácil manipulação em telas pequenas, e o sistema de “arrastar e soltar” (*drag-and-drop*) é otimizado para a interação tátil. O *layout* do jogo foi projetado para equilibrar a visualização do mapa com o painel de programação em blocos. Para otimizar o uso da tela, são utilizados painéis semitransparentes, permitindo que todas as informações essenciais estejam acessíveis ao jogador sem comprometer a clareza ou sobrecarregar a interface.

## 4. Metodologia

A metodologia adotada neste trabalho é inspirada no processo iterativo-incremental de *design* de jogos educacionais descrito por [Pires 2021], com adaptações pontuais. A es-



Figura 3. Nível que aborda o conceito do condicional “se” (if)

estrutura em etapas foi baseada no modelo proposto por [Honda et al. 2022], combinando práticas de *design* educacional e *game design* com foco pedagógico. A Figura 4 apresenta uma visão geral das etapas da metodologia, as quais são detalhadas na sequência.



Figura 4. Diagrama de Atividades ilustrando a metodologia adotada.

#### 4.1. Concepção e Definição do Problema

A etapa inicial consistiu em um *brainstorm* para identificar o problema educacional a ser abordado e o perfil do público-alvo. Considerando os desafios enfrentados por estudantes do ensino fundamental e médio no aprendizado de lógica de programação, optou-se por desenvolver um jogo voltado à introdução de estruturas básicas, como sequências, decisões (*if/else*) e repetições (*loops*). A proposta pedagógica inclui o uso da programação por blocos como mecânica de controle no jogo, permitindo que os estudantes aprendam enquanto jogam.

#### 4.2. Planejamento e Elaboração do Documento de *Design* de Jogo Educacional

Na fase de planejamento, foi produzido o Documento de *Design* de Jogo Educacional (*Educational Game Design Document* - EGDD), com base no modelo de [Rogers 2014]. Este documento incluiu elementos como: objetivos de aprendizagem, narrativa, mecânicas de jogo, usabilidade em dispositivos móveis, estrutura das fases, estética, público-alvo, interface e critérios de avaliação pedagógica. Além disso, o EGDD contemplou uma funcionalidade complementar que apresenta uma representação textual dos algoritmos criados com blocos visuais, com o objetivo de promover a transição gradual para a programação textual.

### 4.3. Prototipação de Alta Fidelidade

O desenvolvimento do Codebricks partiu diretamente para a implementação de um protótipo funcional na Unity Engine<sup>3</sup>, caracterizando-se como um protótipo de alta fidelidade. Essa decisão foi tomada com base na necessidade de validar rapidamente as mecânicas de programação por blocos em dispositivos móveis.

A construção do protótipo foi orientada pelo EGDD, previamente elaborado. Esse documento guiou a estruturação das interfaces, mecânicas, narrativa e objetivos pedagógicos do jogo. O protótipo incluiu as funcionalidades essenciais para a jogabilidade: montagem de blocos visuais com sistema de *drag-and-drop*, execução do algoritmo, movimentação do personagem no cenário e retorno visual imediato.

Até o momento, o protótipo não foi testado com usuários reais. A aplicação de instrumentos de avaliação como *System Usability Scale* (SUS)<sup>4</sup> [Brooke et al. 1996], Self-Assessment Manikin (emoti-SAM)<sup>5</sup> [Bradley and Lang 1994] e MEEGA+ [Petri et al. 2019] está prevista para a próxima fase do projeto, após a inclusão de novas fases e estabilização dos principais recursos. Esses testes serão fundamentais para validar a usabilidade, clareza dos conceitos e engajamento dos estudantes com a proposta pedagógica do jogo.

### 4.4. Arquitetura Técnica

O desenvolvimento do jogo está sendo realizado na Unity Engine, uma plataforma amplamente utilizada para a criação de jogos em duas e três dimensões com suporte multiplataforma e bom desempenho em dispositivos móveis. A escolha pela Unity se justifica por sua curva de aprendizado acessível, ampla comunidade de desenvolvedores, vasta disponibilidade de materiais de estudo, além do suporte robusto à linguagem C# e da facilidade na integração de elementos gráficos e interativos.

Até o momento, os testes de execução estão sendo realizados em um ambiente Android específico utilizado durante o desenvolvimento. Etapas futuras incluirão testes com diferentes resoluções de tela e versões de sistema operacional, a fim de validar a compatibilidade e a responsividade do jogo em múltiplos dispositivos móveis.

### 4.5. Avaliação da Experiência do Usuário com MEEGA+

Com o objetivo de avaliar a experiência dos estudantes com o jogo educacional, será utilizado o instrumento MEEGA+ [Petri et al. 2019]. O MEEGA+ é um modelo validado para a avaliação da qualidade de jogos educacionais, com foco principal na experiência do usuário, considerando aspectos como engajamento, imersão, desafio, usabilidade, clareza de objetivos e aprendizado percebido.

Embora a medição direta da aprendizagem apresente desafios, o MEEGA+ permite avaliar o engajamento e a experiência subjetiva dos usuários com o jogo, fornecendo indicadores relevantes sobre sua aceitação e potencial educacional. O modelo é composto por um conjunto de afirmações distribuídas em sete dimensões principais, que devem ser avaliadas pelos usuários por meio de uma escala Likert de cinco pontos.

---

<sup>3</sup><https://unity.com/pt>

<sup>4</sup><https://www.taylorfrancis.com/chapters/edit/10.1201/9781498710411-35/sus-quick-dirty-usability-scale-john-brooke>

<sup>5</sup>[https://doi.org/10.1016/0005-7916\(94\)90063-9](https://doi.org/10.1016/0005-7916(94)90063-9)

Neste trabalho, a aplicação do MEEGA+ ocorrerá após os testes com os alunos do ensino fundamental e médio, permitindo que os dados obtidos identifiquem pontos fortes e fracos da experiência com o Codebricks, auxiliando na melhoria contínua do jogo e na validação de sua eficácia como ferramenta de apoio ao ensino de lógica de programação.

## 5. Considerações Finais e Próximos Passos

Este trabalho apresentou o desenvolvimento do Codebricks, um jogo educacional do tipo RPG 2D voltado ao ensino de lógica de programação com blocos visuais em dispositivos móveis. A proposta surgiu a partir da constatação de que, embora o ensino de programação seja cada vez mais valorizado, ainda são escassas as ferramentas educacionais acessíveis, lúdicas e adaptadas à realidade tecnológica de estudantes do ensino fundamental e médio. Ao unir elementos de gamificação, narrativa interativa e programação por blocos inspirada no Scratch, o Codebricks busca promover o engajamento dos alunos e facilitar o aprendizado de conceitos fundamentais da lógica de programação.

Até o momento, foram implementadas as fases iniciais do jogo, que abordam os conceitos de sequência, estruturas condicionais (*if* e *if/else*) e laços de repetição (*while*), além de uma interface adaptada para telas sensíveis ao toque e um sistema de programação por blocos integrado à narrativa do jogo. A construção do protótipo de alta fidelidade foi realizada na Unity Engine, com base em um Documento de Design de Jogo Educacional (EGDD) e orientada por fundamentos pedagógicos e de *design* instrucional. A experiência do usuário será posteriormente avaliada com o instrumento MEEGA+, considerando aspectos como engajamento, clareza de objetivos e usabilidade.

Entre os próximos passos do projeto, destacam-se: (i) a inclusão de novos desafios para os conceitos já implementados e, futuramente, para o bloco *for*, atualmente em fase final de implementação; (ii) a introdução progressiva de novos blocos de programação, visando contemplar conteúdos mais complexos de pensamento computacional; e (iii) a realização de testes com estudantes em ambientes educacionais reais.

O jogo foi concebido com uma estrutura modular, o que possibilita a expansão gradual de conteúdo, respeitando uma curva de dificuldade que favoreça a progressão do aprendizado. Essa evolução será fundamentada em princípios do pensamento computacional, como decomposição, abstração, reconhecimento de padrões e construção de algoritmos, em consonância com a Base Nacional Comum Curricular (BNCC).

A avaliação do jogo será conduzida em dois eixos complementares. O primeiro, de natureza técnica, visa garantir a estabilidade e compatibilidade do Codebricks em diferentes dispositivos móveis, por meio de testes de desempenho e identificação de possíveis falhas. O segundo, de cunho pedagógico, será realizado com a participação de turmas do ensino fundamental e médio, mediante aprovação de Comitê de Ética. Serão aplicados instrumentos quantitativos e qualitativos para avaliar aspectos como usabilidade, clareza conceitual, engajamento e impacto na aprendizagem.

Ao final do processo de desenvolvimento e avaliação, prevê-se o lançamento público do Codebricks, acompanhado de um material de apoio para professores, contendo orientações metodológicas e sugestões de uso pedagógico. A expectativa é que o jogo se consolide como uma ferramenta eficaz para o ensino de lógica de programação, promovendo uma aprendizagem ativa, acessível e alinhada às vivências digitais dos estudantes.



## Referências

- Albán Bedoya, I. and Ocaña-Garzón, M. (2021). Educational programming as a strategy for the development of logical-mathematical thinking. In *XV Multidisciplinary International Congress on Science and Technology*, pages 309–323. Springer.
- Araújo, K. F. and Silva, T. d. (2024). A inserção do pensamento computacional nos currículos do novo ensino médio no brasil. *Revista e-Curriculum*, 22.
- Bers, M. U. (2020). *Coding as a playground: Programming and computational thinking in the early childhood classroom*. Routledge.
- Bradley, M. M. and Lang, P. J. (1994). Measuring emotion: the self-assessment manikin and the semantic differential. *Journal of behavior therapy and experimental psychiatry*, 25(1):49–59.
- Brasil (2018). Base Nacional Comum Curricular: Educação é a Base. Ministério da Educação. Disponível em <https://basenacionalcomum.mec.gov.br/>.
- Brezolin, C. V. S., Correa, C. M., and Silveira, M. S. (2023). Apoio ao desenvolvimento de recursos de ajuda-voltados para usuários professores-para linguagens de programação visual baseadas em blocos. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1850–1861. SBC.
- Brooke, J. et al. (1996). Sus-a quick and dirty usability scale. *Usability evaluation in industry*, 189(194):4–7.
- CGI.br/NIC.br (2021). Pesquisa sobre o uso da internet por crianças e adolescentes no brasil – tic kids online brasil 2021. Relatório técnico, CGI.br, São Paulo.
- de Souza, F. A., Falcão, T. P., and Mello, R. F. (2021). O ensino de programação na educação básica: uma revisão da literatura. *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1265–1275.
- Gee, J. P. (2003). What video games have to teach us about learning and literacy. *Computers in entertainment (CIE)*, 1(1):20–20.
- Geraldes, W. and Afonseca, U. (2024). Estratégia pedagógica para permanência e êxito na disciplina de algoritmos e técnicas de programação do curso técnico em informática para internet. In *Anais do XXXV Simpósio Brasileiro de Informática na Educação*, pages 3097–3105, Porto Alegre, RS, Brasil. SBC.
- Grover, S. and Pea, R. (2013). Computational thinking in k–12: A review of the state of the field. *Educational researcher*, 42(1):38–43.
- Honda, F., Pires, F., Pessoa, M., and Maia, J. (2022). Cadê minha pizza? um jogo para exercitar matemática e pensamento computacional através de grafos. In *Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 876–885. SBC.
- Josko, J. M. B. and de Assis Zampirolli, F. (2023). Introducing programming logic consistently through gamification. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 486–495. SBC.
- Li, M., Ma, S., and Shi, Y. (2023). Examining the effectiveness of gamification as a tool promoting teaching and learning in educational settings: a meta-analysis. *Frontiers in Psychology*, 14:1253549.

- Perin, A. P. J., Silva, D. E., and Valentim, N. M. C. (2021). Um benchmark de ferramentas de programação em blocos que podem ser utilizadas nas salas de aula do ensino médio. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 1162–1173. SBC.
- Petri, G., Von Wangenheim, C. G., and Borgatto, A. F. (2019). Meega+: Um modelo para a avaliação de jogos educacionais para o ensino de computação. *Revista Brasileira de Informática na Educação*, 27(03):52–81.
- Pires, F. G. d. S. (2021). Thinkted lab, um caso de aprendizagem criativa em computação no nível superior.
- Raabe, A. and de Araújo Cavalcante, E. (2024). Revisão sistemática sobre a formação professores da educação básica para ensinar computação. *Informática na educação: teoria & prática*, 27(2).
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., et al. (2009). Scratch: programming for all. *Communications of the ACM*, 52(11):60–67.
- Ricarte, D. R., Nery, L. D., Castro, R., de Figueiredo, R. V., de Lima Costa, T. K., Cavalcanti, V., de Souza Rebouças, A. D. D., et al. (2024). Desenvolver para codificar: Projeto para o ensino de programação no ensino fundamental em escolas públicas. In *Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 3173–3182. SBC.
- Rodrigues, A., Gomes, K., and Carneiro, M. (2022). Scratchim: uma abordagem para o ensino do pensamento computacional para crianças de forma remota e desplugada. In *Anais do XXXIII Simpósio Brasileiro de Informática na Educação*, pages 1444–1455, Porto Alegre, RS, Brasil. SBC.
- Rogers, S. (2014). *Level Up! The guide to great video game design*. John Wiley & Sons.
- Silva, R. R., Rivero, L., and Santos, R. P. d. (2021). Programse: Um jogo para aprendizagem de conceitos de lógica de programação. *Revista Brasileira de Informática na Educação*, 29:301–330.
- Sousa, L., Farias, E., and Carvalho, W. (2020). Programação em blocos aplicada no ensino do pensamento computacional: Um mapeamento sistemático. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1513–1522, Porto Alegre, RS, Brasil. SBC.
- Souza, M. R. V. B., de Freitas, L. H., Silva, G. d. S., de Carvalho, F. X., and Souza, L. A. M. (2024). Uma proposta para o uso de rpg no ensino de física: A vingança de newton. *arXiv preprint arXiv:2411.17642*.
- Su, J. and Yang, W. (2023). A systematic review of integrating computational thinking in early childhood education. *Computers and Education Open*, 4:100122.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.