

Explorando a Lógica de Programação com Visualização da Execução em um Jogo Educacional Digital

Allan Rodrigues¹, Bruno Vasconcelos¹, Sarah Silva¹
Pedro Henrique Valle², Alessandreia Oliveira¹

¹Departamento de Ciência da Computação - UFJF – Juiz de Fora – MG – Brasil

²Universidade de São Paulo - (IME/USP) – São Paulo – SP – Brasil

{allan.henriques, moraes.bruno, sarah.cristina}@estudante.ufjf.br,
pedrohenriquevalle@usp.br, alessandreia.oliveira@ufjf.br

Abstract. Introduction: Introductory programming courses present high failure rates, associated with difficulties in understanding abstract concepts and low motivation. As an alternative, Digital Educational Games (DEGs) have been used to promote engagement and develop Computational Thinking. **Objective:** To present LostBots, a DEG that proposes teaching programming logic through maze-based challenges, in which students build algorithms using predefined commands and follow their execution. Unlike traditional approaches, the game incorporates a computational cost model through a battery system and efficiency-based scoring, for optimization of solutions. **Methodology:** The evaluation was conducted with 30 students using the MEEGA+ model. **Results:** The results indicate high levels of engagement, usability, and perceived learning, suggesting that the approach is promising. **Keywords:** Programming Logic, Digital Educational Games, Computational Thinking, MEEGA+.

Resumo. Introdução: Disciplinas introdutórias de programação apresentam altas taxas de reprovação, associadas à dificuldade em compreender conceitos abstratos e à baixa motivação. Como alternativa, Jogos Educacionais Digitais (JEDs) têm sido utilizados para promover engajamento e desenvolver o Pensamento Computacional. **Objetivo:** Apresentar o LostBots, um JED que propõe o ensino de lógica de programação por meio de desafios em labirintos, nos quais estudantes constroem algoritmos com comandos pré-definidos e acompanham sua execução. Diferentemente das abordagens tradicionais, o jogo incorpora um modelo de custo computacional por meio de um sistema de baterias e de pontuação por eficiência, para otimização de soluções. **Metodologia:** A avaliação foi realizada com 30 estudantes, utilizando o modelo MEEGA+. **Resultados:** Indicam alto engajamento, usabilidade e aprendizagem percebida, o que sugere que a abordagem é promissora. **Palavras-chave:** Lógica de programação, Jogos educacionais digitais, Pensamento computacional, MEEGA+.

1. Introdução

O ensino de programação é um pilar fundamental na formação de profissionais de tecnologia. No entanto, disciplinas introdutórias apresentam altas taxas de reprovação e evasão [Llanos et al. 2023, Pires et al. 2024], frequentemente associadas às dificuldades dos estudantes em compreender conceitos abstratos e em desenvolver o raciocínio lógico

necessário à construção de algoritmos. Esse cenário pode gerar frustração e desmotivação logo nos primeiros períodos dos cursos, sendo apontado na literatura como um dos principais desafios na Educação em Computação [Pereira e Sousa 2024].

Como alternativa, diferentes estratégias pedagógicas têm sido investigadas, com destaque para o uso de Jogos Educacionais Digitais (JED). A literatura científica indica que o uso de elementos lúdicos pode aumentar o engajamento, a motivação e a autoconfiança dos estudantes [Malliarakis et al. 2017]. Além disso, ambientes visuais interativos permitem a visualização da execução sequencial dos algoritmos, oferecendo *feedback* imediato e favorecendo a construção de modelos mentais mais consistentes [Vahldick et al. 2020, Malliarakis et al. 2017].

Apesar desses avanços, muitas ferramentas existentes concentram-se na resolução de problemas, sem explorar explicitamente aspectos relacionados à eficiência das soluções [Paiva et al. 2022, Keuning et al. 2018]. Dessa forma, conceitos importantes, como custo computacional e otimização, tendem a ser introduzidos apenas em estágios mais avançados da formação, o que pode limitar o desenvolvimento precoce de um pensamento algorítmico mais refinado [Ginat 2001].

Neste contexto, este trabalho apresenta o LostBots, um JED para apoiar o ensino de lógica de programação por meio de desafios em labirintos. No jogo, os estudantes constroem algoritmos com comandos pré-definidos e acompanham sua execução de forma visual e procedural. Como diferencial, o JED incorpora um modelo de custo computacional por meio de um sistema de bateria e pontuação, baseado na eficiência das soluções, incentivando não apenas a resolução dos problemas, mas também sua otimização. O jogo foi desenvolvido como uma aplicação Web, com foco em acessibilidade e facilidade de uso. Para avaliar a proposta, realizou-se um estudo de caso com estudantes de diferentes cursos, cujos dados foram analisados com base no modelo MEEGA+ (*Model for the Evaluation of Educational Games*) [Petri e von Wangenheim 2016]. As principais contribuições deste trabalho são: (i) a proposta de um JED orientado à eficiência algorítmica, incorporando o conceito de custo computacional no processo de aprendizagem; (ii) o *design* de mecânicas de jogo que incentivam a otimização de soluções desde os níveis iniciais de programação; (iii) o desenvolvimento de uma ferramenta acessível, baseada em Web, que permite a visualização detalhada da execução de algoritmos; (iv) uma avaliação da experiência do usuário e da percepção de aprendizagem utilizando o modelo MEEGA+.

Este trabalho está organizado da seguinte forma: a Seção 2 apresenta o embasamento teórico sobre JEDs no ensino de programação e a Seção 3 apresenta os trabalhos relacionados. A Seção 4 descreve o desenvolvimento e a arquitetura do LostBots. A Seção 5 apresenta a avaliação com o modelo MEEGA+ e a Seção 6 apresenta os resultados obtidos. A Seção 7 discute as ameaças à validade, e por fim, a Seção 8, as considerações finais.

2. Fundamentação Teórica

As disciplinas introdutórias de lógica de programação representam um dos principais desafios nos cursos de Computação e áreas afins. Estudantes ingressantes frequentemente enfrentam dificuldades para compreender conceitos abstratos e desenvolver o raciocínio necessário à construção de algoritmos [Paiva et al. 2022]. Essas dificuldades estão associadas à necessidade de transição do pensamento cotidiano para o Pensamento

Computacional [Wing 2006], que exige habilidades como a decomposição de problemas, abstração e a organização lógica de instruções. Outro fator relevante é a dificuldade dos estudantes em compreender a execução dos algoritmos, especialmente quanto à relação entre as instruções e seus efeitos. A ausência de *feedback* visual claro pode dificultar a construção de modelos mentais consistentes, prejudicando a aprendizagem.

Nesse contexto, os JED têm sido utilizados como estratégia para tornar o aprendizado mais engajante e significativo. A literatura aponta que a Aprendizagem Baseada em Jogos pode contribuir para o aumento do engajamento, da motivação e da autoconfiança dos estudantes, além de favorecer a compreensão de conceitos abstratos [Malliarakis et al. 2017]. Ambientes visuais interativos, em particular, permitem a visualização da execução dos algoritmos, proporcionando *feedback* imediato e facilitando a construção do raciocínio lógico [Vahldick et al. 2020]. Além disso, abordagens exploram o uso de tecnologias, como ambientes baseados em blocos e realidade aumentada, para apoiar o ensino de programação, evidenciando o potencial de recursos visuais e interativos na aprendizagem [Sharma et al. 2022].

Apesar desses avanços, observa-se que a maioria das ferramentas educacionais concentra-se na execução correta dos algoritmos, com menor ênfase na qualidade das soluções. Aspectos como eficiência, otimização e custo computacional tendem a ser introduzidos apenas em estágios mais avançados, o que pode limitar o desenvolvimento precoce de um pensamento algorítmico mais refinado. Nesse sentido, abordagens que integrem, desde os níveis iniciais, elementos que incentivem a reflexão sobre a eficiência das soluções podem contribuir para uma compreensão mais completa do processo de construção de algoritmos, ampliando o escopo da aprendizagem em programação.

3. Trabalhos Relacionados

O uso de ambientes baseados em agentes em grade para o ensino introdutório de programação possui uma longa tradição na área de Educação em Computação. Um dos exemplos mais clássicos é o *Karel the Robot* [Roberts 2001], no qual estudantes desenvolvem algoritmos para controlar um robô em um ambiente estruturado. Abordagens mais recentes também exploram ambientes em formato de labirinto, com progressão por fases e execução de código integrada, reforçando o uso de desafios estruturados no ensino de lógica de programação [Neto et al. 2021].

Entre essas ferramentas, destaca-se o *Reeborg's World* [Roberge 2024], um ambiente online que permite a construção e execução de algoritmos por meio da manipulação de um agente em cenários baseados em grade, com suporte a linguagens como Python e JavaScript. Tais ambientes favorecem a compreensão da execução passo a passo dos algoritmos e reduzem a carga cognitiva associada à sintaxe. De forma semelhante, JEDs têm explorado ambientes interativos com execução visual e progressão por níveis, combinando elementos de programação com *feedback* imediato para apoiar a compreensão da lógica [Oliveira et al. 2021, Silva et al. 2022].

Paralelamente, a literatura aponta o uso de JEDs como uma estratégia promissora para aumentar o engajamento e a motivação dos estudantes. Estudos de revisão da literatura indicam que a Aprendizagem Baseada em Jogos pode contribuir positivamente para o ensino de programação, promovendo maior envolvimento e facilitando a compreensão de conceitos abstratos [Malliarakis et al. 2017]. Além disso, abordagens que incorporam elementos visuais e interativos, como ambientes baseados em blocos ou

tecnologias emergentes, também demonstram potencial para aprimorar a aprendizagem [Vahldick et al. 2020, Sharma et al. 2022]. Diversos JEDs têm incorporado mecânicas como narrativa, desafios progressivos e sistemas de recompensa, associadas a *feedback* imediato, como estratégias para aumentar o engajamento e apoiar o aprendizado de conceitos computacionais [Macena et al. 2022, Abreu et al. 2023].

Apesar dos avanços apresentados por essas iniciativas, observa-se que a maioria das ferramentas e dos JEDs concentra-se na execução correta dos algoritmos e na resolução dos problemas propostos. Mesmo em propostas recentes que exploram ambientes interativos, narrativa e *feedback* imediato, o foco permanece predominantemente na correção da solução e na progressão do jogador, enquanto aspectos relacionados à eficiência das soluções, como custo computacional e otimização, são geralmente pouco explorados ou introduzidos apenas em estágios mais avançados da formação [Neto et al. 2021, Oliveira et al. 2021, Macena et al. 2022].

Nesse contexto, o LostBots diferencia-se das abordagens citadas, como Karel the Robot e Reeborg's World, ao preencher uma lacuna ao incorporar explicitamente a noção de custo computacional no processo de aprendizagem. Enquanto a maioria das propostas enfatiza a execução correta e o avanço por fases, sem considerar explicitamente a eficiência das soluções, o LostBots, por meio de mecânicas como o consumo de bateria e a pontuação baseada na eficiência, incentiva os estudantes a não apenas resolver os desafios, mas também a refletir sobre a qualidade e a otimização de suas soluções desde os estágios iniciais do aprendizado de programação.

4. Desenvolvimento e Arquitetura do LostBots

O desenvolvimento do LostBots foi guiado pelos princípios de acessibilidade e portabilidade, resultando em uma solução baseada em tecnologias Web, executável diretamente no navegador, sem necessidade de instalação. Essa abordagem favorece a adoção em diferentes contextos educacionais e em diferentes dispositivos.

4.1. Arquitetura de Software

A Figura 1 ilustra uma visão geral da arquitetura do JED, evidenciando a organização dos principais componentes e o fluxo de interação entre a interface do jogador, o mecanismo de execução lógica e o sistema de avaliação das soluções. De forma geral, a arquitetura segue um modelo distribuído em camadas, organizado em quatro componentes principais: (i) cliente (interface do jogador), (ii) aplicação Web (*front-end*), (iii) serviços (*back-end*) e (iv) persistência de dados. Esses componentes se comunicam por meio de requisições HTTP, permitindo a execução dos algoritmos, a avaliação das soluções e o armazenamento do progresso do jogador.

No nível de implementação, o sistema utiliza tecnologias distribuídas entre essas camadas, com destaque para:

- **Motor Gráfico (Phaser 3):** responsável pela renderização do ambiente, gerenciamento do ciclo de execução e controle de interações no labirinto¹;
- **Interface de Programação:** construída com HTML e CSS sobrepostos ao ambiente do jogo, permitindo a composição de algoritmos por meio de comandos pré-definidos^{2 3};

¹<https://phaser.io/>

²https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API

³<https://developer.mozilla.org/en-US/docs/Web/API/HTMLElement>

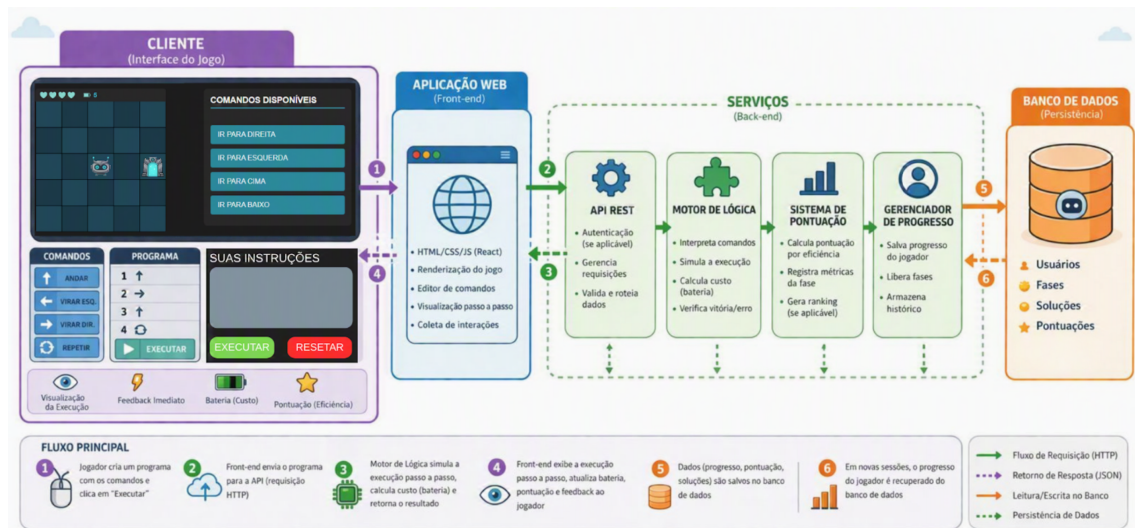


Figura 1. Visão geral da arquitetura.

- **Persistência de Dados (Firebase):** utilizada para autenticação de usuários e manutenção do estado da sessão do jogador (*save game*)⁴.

Essa organização separa as responsabilidades entre a interface, a execução lógica, a avaliação e a persistência, contribuindo para a manutenção e a evolução do sistema.

4.2. Lógica de Execução e Níveis

A lógica do jogo está implementada em um modelo baseado em grade, no qual o agente interage com um ambiente composto por diferentes tipos de células, como caminhos, obstáculos e objetivos. Os níveis são definidos por estruturas que especificam o *layout*, a posição inicial do agente, os comandos disponíveis e os limites de recursos. A execução dos algoritmos ocorre por meio de uma fila de comandos, interpretada sequencialmente. A cada instrução executada, o estado do ambiente é atualizado e apresentado visualmente, permitindo que o estudante acompanhe o comportamento do algoritmo. A Figura 2 exemplifica a execução em um nível inicial.

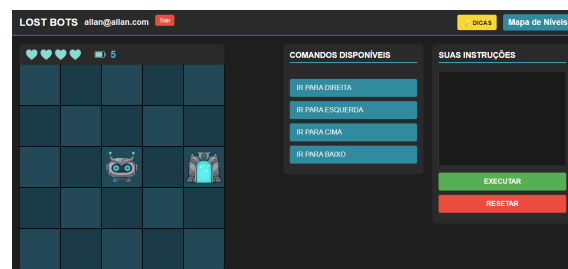


Figura 2. Interface do jogo em um nível inicial.

Além da verificação de corretude, a execução considera o consumo de recursos associado às ações do agente, representado por um sistema de bateria. Esse mecanismo incentiva o planejamento das soluções e a redução de ações desnecessárias. A avaliação dos níveis considera a conclusão do objetivo e o desempenho da solução, expressos por meio de um sistema de pontuação por estrelas, o que incentiva a revisão e o refinamento.

⁴<https://firebase.google.com/>

4.3. Estrutura de Fases e Progressão

O LostBots foi projetado com uma progressão baseada em níveis, organizados de forma a introduzir gradualmente conceitos de lógica de programação. Cada fase apresenta um novo desafio em um ambiente de labirinto, no qual o estudante deve construir uma sequência de comandos para atingir um objetivo específico. À medida que o jogo avança, os desafios tornam-se mais complexos, exigindo maior planejamento e organização das soluções.

Além do aumento da complexidade estrutural dos cenários, a progressão também explora restrições relacionadas ao uso de recursos. A limitação de bateria e o sistema de pontuação incentivam o jogador a refletir sobre a eficiência das soluções, promovendo a revisão e o refinamento dos algoritmos construídos. Essa organização permite que a aprendizagem ocorra de forma incremental, equilibrando desafio e habilidade e contribuindo para o desenvolvimento gradual do raciocínio algorítmico.

4.4. Responsividade e Adaptação Móvel

O JED foi projetado para operar em diferentes dispositivos, permitindo sua utilização em diversos contextos educacionais. A interface adapta-se dinamicamente ao ambiente de execução. Em dispositivos *desktop*, os elementos são organizados lateralmente, enquanto, em dispositivos móveis, verticalmente, priorizando a legibilidade e a interação por toque.

5. Avaliação

Para investigar o potencial do LostBots como apoio ao ensino de programação, foi conduzido um estudo de caso com abordagem quantitativa. A avaliação analisou a experiência do usuário e a percepção de aprendizagem com o MEEGA+. Especificamente, o estudo buscou responder às seguintes questões de pesquisa (QPs):

- **QP1:** O jogo promove engajamento e uma experiência de uso positiva?
- **QP2:** Os estudantes percebem o JED como útil para o aprendizado de lógica de programação?

5.1. Metodologia e Procedimentos de Aplicação

A aplicação prática do jogo ocorreu nas dependências do Laboratório L205 do Instituto de Ciências Exatas (ICE) da Universidade Federal de Juiz de Fora (UFJF). O estudo foi conduzido em duas sessões distintas com alunos de Algoritmos: a primeira turma participou às 10h (18 alunos) e a segunda, às 14h (12 alunos), totalizando 30 participantes.

O convite foi realizado mediante autorização prévia dos professores responsáveis pela disciplina. Antes do início da atividade, foi apresentada uma explicação geral sobre o funcionamento do jogo e os objetivos da pesquisa. Um dos autores deste trabalho atuou como mediador e observador durante a aplicação. Todos os participantes tiveram acesso ao Termo de Consentimento Livre e Esclarecido (TCLE)⁵, garantindo anonimato e participação voluntária. O procedimento experimental seguiu as seguintes etapas:

1. **Apresentação Inicial:** contextualização da atividade e instruções de uso do jogo;
2. **Sessão de Interação:** os estudantes utilizaram o LostBots por aproximadamente 25 minutos, explorando livremente os desafios propostos;

⁵Disponível em: <https://tinyurl.com/4fzdfrbc>

3. **Coleta de Dados:** aplicação de um questionário digital via *Google Forms*⁶, com questões de caracterização e itens do modelo MEEGA+.

Durante a atividade, foram observados alguns imprevistos. Parte dos alunos ingressou na sessão após o início da explicação, o que tornou necessárias orientações individuais adicionais. Além disso, foi identificado um problema técnico no *cache* do navegador, que levou ao carregamento de versões desatualizadas em algumas máquinas. O problema foi resolvido com a atualização forçada da página (*hard refresh*).

5.2. Caracterização dos Participantes

A amostra analisada compreendeu 30 estudantes de graduação. O perfil dos participantes é predominantemente composto por alunos ingressantes, com maior concentração na faixa etária de 18 a 21 anos (26 alunos), característica típica de disciplinas introdutórias.

Observou-se uma distribuição de gênero não paritária (19 participantes do gênero masculino e 11 do feminino), além de diversidade de cursos, incluindo Engenharia Civil (17 alunos), Engenharia Elétrica (8 alunos), Ciências Biológicas (2), Computação (1), Engenharia Computacional (1) e Mestrado em Modelagem Computacional o que contribui para a heterogeneidade da amostra.

A heterogeneidade da amostra, composta majoritariamente por estudantes de cursos fora da área de Computação, como Engenharias e Biologia, constituiu um ponto estratégico para a validação do JED. Essa composição permitiu analisar o potencial de disseminação do Pensamento Computacional em contextos interdisciplinares, bem como avaliar a eficácia do LostBots como ferramenta de apoio ao aprendizado além do domínio específico da Computação. Ao testar o jogo com alunos ingressantes que não têm contato direto e constante com programação, o estudo evidencia que o jogo é capaz de apoiar o desenvolvimento do raciocínio lógico de pessoas com perfis de conhecimento diversos, independentemente da formação acadêmica.

Em relação à experiência prévia, a maioria dos participantes declarou possuir pouco (9 alunos) ou nenhum conhecimento (16 alunos) em programação, o que indica a adequação do público-alvo ao propósito do jogo. Esse aspecto é relevante, uma vez que o LostBots foi projetado para apoiar o aprendizado em níveis iniciais. A percepção dos estudantes quanto à facilidade de compreensão de programação apresentou distribuição variada (Figura 3), enquanto o nível de conforto atual indicou tendência levemente positiva (Figura 4). Adicionalmente, a maioria relatou familiaridade com jogos digitais, informando que joga diariamente (8 alunos) ou frequentemente (9), embora o uso prévio de jogos educacionais não seja predominante, conforme indicado por 22 alunos. Esses fatores indicam que o contexto de aplicação é representativo de cenários reais de ensino introdutório, permitindo avaliar o potencial do jogo em condições próximas às práticas educacionais usuais.

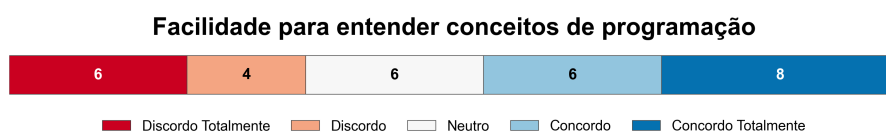


Figura 3. Percepção de facilidade para compreender conceitos de programação.

⁶Disponível em: <https://tinyurl.com/3xkru784>

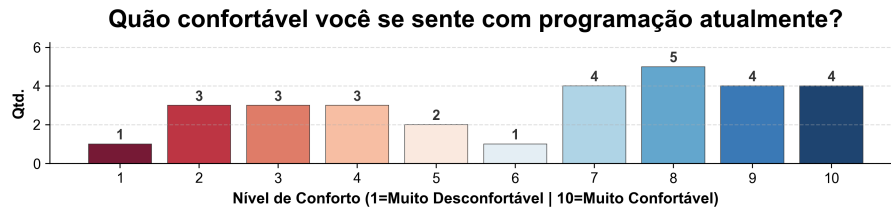


Figura 4. Nível de conforto atual com programação (Escala 1-10).

6. Resultados da Avaliação

Esta seção apresenta os resultados da aplicação do modelo MEEGA+⁷, organizados conforme as QPs definidas na Seção 5. Os dados correspondem à distribuição de respostas em escala Likert (1 = Discordo Totalmente a 5 = Concordo Totalmente).

6.1. Experiência do usuário, engajamento e usabilidade (QP1)

Os resultados indicam que o LostBots promove uma experiência de uso positiva, com destaque para os aspectos de engajamento, imersão, diversão e usabilidade. Quanto ao engajamento, conforme apresentado na Figura 5, observa-se forte concentração de respostas nas categorias “Concordo” e “Concordo Totalmente”, indicando que o jogo foi eficaz em manter a atenção e o interesse dos participantes durante a atividade. Resultados semelhantes são observados na dimensão Imersão, embora com maior proporção de respostas neutras em itens relacionados à perda de noção do tempo. Esse comportamento é esperado, considerando o tempo limitado da sessão (25 minutos), o que tende a restringir a imersão a níveis mais superficiais.

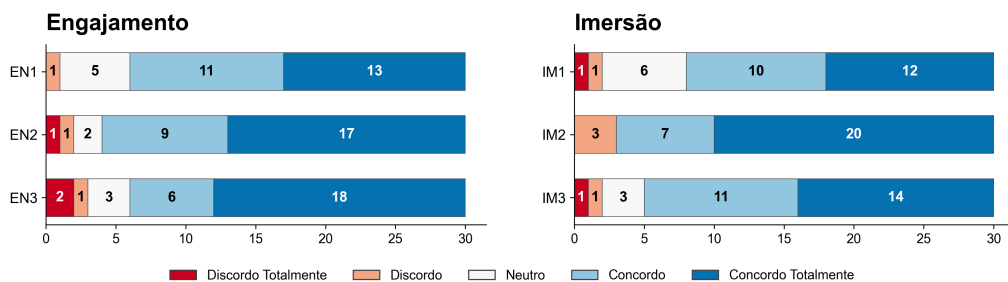


Figura 5. Resultados das dimensões de Engajamento e Imersão.

A dimensão Diversão (Figura 6) obteve uma das avaliações mais positivas, com predominância de respostas nas categorias superiores da escala. Esse resultado sugere que o jogo conseguiu tornar a experiência de aprendizagem mais atraente e menos árida. Em relação à usabilidade, a maioria avaliou positivamente a clareza das instruções e a facilidade de uso. As poucas avaliações neutras ou negativas podem estar associadas a problemas técnicos pontuais. De forma geral, os resultados associados à QP1 indicam que o LostBots proporciona uma experiência positiva, engajando e oferecendo uma interface adequada ao público iniciante. Esses achados reforçam o potencial da ferramenta em contextos educacionais reais, especialmente por sua acessibilidade, uma vez que pode ser usada sem instalação prévia e em diferentes dispositivos.

⁷Disponível em: <https://tinyurl.com/5n6w2aub>

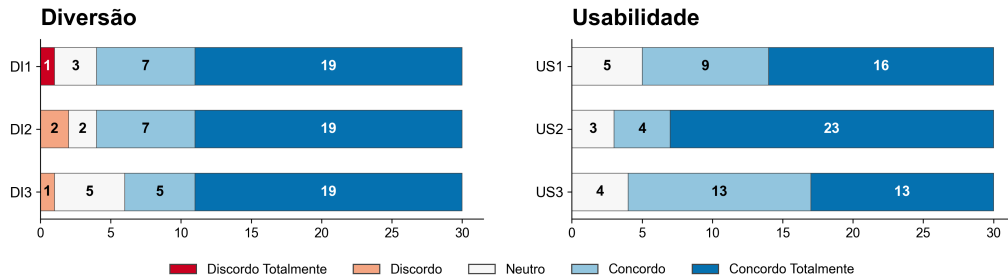


Figura 6. Resultados das dimensões de Diversão e de Usabilidade.

6.2. Percepção de aprendizagem e motivação (QP2)

A análise da percepção de aprendizagem indica resultados fortemente positivos. Conforme ilustrado na Figura 7, a grande maioria concorda que o jogo contribuiu para a compreensão dos conceitos de lógica de programação. Além disso, a dimensão Motivação também apresentou resultados positivos, indicando que os participantes demonstraram interesse em continuar aprendendo programação após a experiência com o jogo.

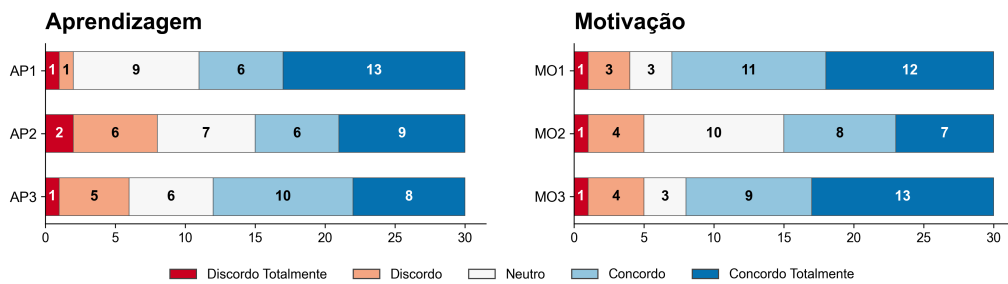


Figura 7. Resultados das dimensões de Aprendizagem e Motivação.

Esses resultados sugerem que o LostBots não apenas facilita a compreensão inicial dos conceitos, mas também contribui para o aumento do interesse dos estudantes pela área, aspecto fundamental em disciplinas introdutórias.

6.3. Síntese dos Resultados

Em conjunto, os resultados indicam que o jogo apresenta desempenho positivo nas dimensões avaliadas pelo modelo MEEGA+, especialmente em engajamento, diversão, usabilidade e percepção de aprendizagem. Embora a avaliação baseie-se na percepção dos estudantes, os achados são consistentes com o objetivo da proposta, que é tornar o aprendizado mais acessível e motivador. Além disso, os resultados reforçam o potencial do jogo como ferramenta de apoio em contextos introdutórios, particularmente para estudantes com pouca experiência em programação, evidenciando sua adequação a cenários educacionais com diferentes perfis de alunos e limitações de infraestrutura.

Adicionalmente, observa-se que a estrutura do jogo, baseada na execução passo a passo e na necessidade de planejamento das ações, pode contribuir para o desenvolvimento do raciocínio algorítmico. Ainda que aspectos relacionados à eficiência das soluções não tenham sido avaliados diretamente neste estudo, a mecânica de limitação de recursos (bateria) configura-se como um elemento promissor para incentivar práticas de otimização, a ser investigado em estudos futuros. Embora não tenha sido realizada uma avaliação direta de desempenho (e.g., pré e pós-teste), os resultados obtidos fornecem evidências iniciais sobre a aceitação e o potencial educacional da ferramenta.

7. Ameaças à Validade

A interpretação dos resultados deve considerar possíveis ameaças à validade [Yin 2009]. Essa análise permite maior transparência na interpretação dos achados. Em relação à **validade interna**, a aplicação ocorreu em duas turmas distintas, em horários diferentes, o que pode ter introduzido variações decorrentes do cansaço e da atenção. Para mitigar esse efeito, o protocolo foi padronizado. Sobre a **validade de construto**, a avaliação baseou-se no modelo MEEGA+, que privilegia aspectos da experiência do usuário e da percepção de aprendizagem. Assim, os resultados refletem a visão subjetiva dos participantes, o que não permite afirmar ganhos efetivos de aprendizagem. Além disso, embora o jogo incorpore conceitos de eficiência algorítmica, esse aspecto não foi avaliado diretamente, o que torna necessária a adoção de métricas objetivas em estudos futuros.

Quanto à **validade externa**, a amostra de 30 estudantes de uma instituição limita a generalização. Ainda assim, a diversidade de cursos e o perfil de iniciantes contribuem para a representatividade do público-alvo. Além disso, a aplicação ocorreu em ambiente controlado, podendo diferir de cenários de uso autônomo. Quanto à **validade de conclusão**, os resultados baseiam-se em análise descritiva, sem testes estatísticos inferenciais. Embora indiquem tendências, não permitem generalizações robustas, sendo interpretados como evidências iniciais sobre a aceitação e o potencial da ferramenta.

8. Considerações Finais

Este trabalho apresentou o desenvolvimento e a avaliação do LostBots, projetado para apoiar o ensino de lógica de programação. Os resultados obtidos com graduandos indicam que o LostBots foi bem recebido nas dimensões avaliadas pelo MEEGA+, com destaque para engajamento, diversão, usabilidade e percepção de aprendizagem. Esses achados sugerem que a abordagem baseada em jogos, aliada à visualização passo a passo da execução, contribui para tornar o aprendizado mais acessível, interativo e motivador. Como principal diferencial, incorpora a noção de custo computacional por meio de mecânicas como a limitação de recursos e a pontuação baseada na eficiência das soluções, incentivando não apenas a resolução correta, mas também a reflexão sobre a qualidade e a otimização dos algoritmos desde os estágios iniciais. Destaca-se, ainda, seu caráter acessível, por ser uma aplicação Web, o que favorece seu uso em diferentes contextos.

Apesar dos resultados promissores, este estudo apresenta limitações, como o tamanho da amostra, a aplicação em uma única instituição e o uso de medidas baseadas nas percepções dos participantes. Além disso, aspectos relacionados à eficiência das soluções não foram avaliados diretamente, o que representa uma oportunidade para estudos futuros. Como trabalhos futuros, pretende-se expandir o LostBots com a inclusão de estruturas de controle de fluxo, como condicionais e laços de repetição, bem como investigar seu impacto no desempenho por meio de métricas objetivas, como análise de soluções e estudos com pré- e pós-teste. Adicionalmente, pretende-se explorar mecanismos que ampliem a análise e comparação de soluções, fortalecendo o desenvolvimento do pensamento algorítmico e da eficiência desde os níveis iniciais.

Declaração sobre uso de Inteligência Artificial

Ferramentas de Inteligência Artificial foram utilizadas exclusivamente para apoio editorial. Todas as decisões de pesquisa, métodos, análises e interpretações foram realizadas integralmente pelos autores, sem que essas ferramentas influenciassem o conteúdo científico ou as conclusões do trabalho.

Referências

- Abreu, J., Castro, H., Reis, C., e Sousa, P. (2023). Poké prog: Jogo educacional para o ensino dos conceitos básicos de programação. In *Anais Estendidos do XXII Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 881–891, Porto Alegre, RS, Brasil. SBC.
- Ginat, D. (2001). Misleading intuition in algorithmic problem solving. In *Proceedings of the Thirty-Second SIGCSE Technical Symposium on Computer Science Education*, SIGCSE '01, pages 21–25, New York, NY, USA. Association for Computing Machinery.
- Keuning, H., Jeuring, J., e Heeren, B. (2018). A systematic literature review of automated feedback generation for programming exercises. *ACM Transactions on Computing Education*, 19(1).
- Llanos, J. et al. (2023). Early prediction of student performance in cs1 programming courses. *PeerJ Computer Science*, 9:e1655.
- Macena, J., Pires, F., Pessoa, M., e Melo, R. (2022). Hello food: um jogo para praticar conceitos de algoritmos para iniciantes na computação. In *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 1066–1075, Porto Alegre, RS, Brasil. SBC.
- Malliarakis, C., Satratzemi, M., e Xinogalos, S. (2017). Cmx: The effects of an educational mmorpg on learning and teaching computer programming. *IEEE Transactions on Learning Technologies*, 10(2):219–235.
- Neto, J. G., Marcolino, P. P., Ferreira, P., e Barbosa, D. (2021). Maze code: Retórica procedural aplicada ao ensino de lógica de programação. In *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 519–528, Porto Alegre, RS, Brasil. SBC.
- Oliveira, Y., Toledo, C., e Pereira, L. (2021). Project Éden: platform for introductory programming concepts. In *Anais Estendidos do XX Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 566–573, Porto Alegre, RS, Brasil. SBC.
- Paiva, J. C., Leal, J. P., e Figueira, A. (2022). Automated assessment in computer science education: A state-of-the-art review. *ACM Computing Surveys*, 55(8).
- Pereira, K. C. e Sousa, R. R. (2024). Dificuldades de aprendizagem e fatores que motivam para o estudo para programação: o que mudou desde os estudos de 2018 no curso de ti da ufersa, campus pau dos ferros? *RECIMA21 - Revista Científica Multidisciplinar*, 5(4):e545151.
- Petri, G. e von Wangenheim, C. G. (2016). Meega+: Systematic model for the evaluation of educational games. Technical Report INCoD/GQS.03.2016.E, Universidade Federal de Santa Catarina (UFSC), Florianópolis, Brazil. Technical Report.
- Pires, J., Brito Correia, F., Gomes, A., Borges, A., e Bernardino, J. (2024). Predicting student performance in introductory programming courses. *Computers*, 13(9):219.
- Roberge, A. (2024). Reeborg's world. Acesso em: 2026.
- Roberts, E. (2001). *Karel the Robot Learns Java*. Stanford University.
- Sharma, V., Bhagat, K. K., Huang, H.-H., e Chen, N.-S. (2022). The design and evaluation of an ar-based serious game to teach programming. *Computers & Graphics*, 103:1–18.

- Silva, A., Nascimento, G., Pereira, J., Neto, L. M., Araújo, M., e Oliveira, V. (2022). Codeland: um jogo para o ensino de lógica de programação com python. In *Anais Estendidos do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital*, pages 643–652, Porto Alegre, RS, Brasil. SBC.
- Vahldick, A., Farah, P. R., Marcelino, M. J., e Mendes, A. J. (2020). A blocks-based serious game to support introductory computer programming in undergraduate education. *Computers in Human Behavior Reports*, 2:100037.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Yin, R. K. (2009). *Case study research: Design and methods*, volume 5. sage.