

Desenvolvimento de Jogos Digitais no Ensino de Algoritmos sob uma Perspectiva Construtivista: Um Relato de Experiência

Péricles Miranda¹, Pedro Henrique Dias Valle², Alessandra Marta de Oliveira³

¹Universidade Federal Rural de Pernambuco – Depto de Computação – UFRPE, Brasil

²Universidade de São Paulo – IME/USP – São Paulo, SP, Brasil

³Universidade Federal de Juiz de Fora – DCC/UFJF – Juiz de Fora, MG, Brasil

pericles.miranda@ufrpe.br, pedrohenriquevalle@usp.br

alessandraia.oliveira@ufjf.br

Abstract. *Teaching algorithms in higher education presents challenges related to abstraction and low engagement. This work presents an experience report of an introductory programming course based on a constructivist approach, in which students develop digital games as a learning strategy. The proposal organizes instruction through progressively practical activities that articulate fundamental programming concepts. The experience took place between 2023 and 2025 and involved 131 students. The results indicate increased engagement, autonomy, and understanding of these concepts, as evidenced by quantitative data and qualitative feedback. The findings suggest that the use of digital games helps reduce abstraction and promote more meaningful learning.*

Resumo. *O ensino de algoritmos no ensino superior enfrenta desafios relacionados à abstração e ao baixo engajamento. Este trabalho apresenta um relato de experiência de uma disciplina introdutória de programação, baseada em uma abordagem construtivista, na qual os estudantes desenvolvem jogos digitais como estratégia de aprendizagem. A proposta organiza o ensino em atividades práticas progressivas que articulam fundamentos de programação. A experiência foi aplicada entre 2023 e 2025, envolvendo 131 estudantes. Os resultados indicam aumento do engajamento, da autonomia e da compreensão desses conceitos, evidenciado por dados quantitativos e feedbacks qualitativos. Os achados sugerem que o uso de jogos digitais contribui para reduzir a abstração e promover uma aprendizagem mais significativa.*

1. Introdução

O ensino de algoritmos constitui um desafio recorrente no ensino superior [Morais et al. 2020, Elegbeleye and Isong 2025], especialmente entre ingressantes em cursos de STEM (Science, Technology, Engineering and Mathematics). Embora esses estudantes costumem apresentar afinidade com raciocínio lógico e matemático, a natureza abstrata dos conceitos algorítmicos, aliada a abordagens tradicionais centradas na exposição teórica e em exercícios descontextualizados, frequentemente resulta em dificuldades de aprendizagem, baixa motivação e engajamento limitado. Esse cenário tem motivado a adoção de metodologias ativas, que promovem maior participação, autonomia e envolvimento com o conteúdo de programação [Córdova-Esparza et al. 2024, Díaz-Lauzurica and Moreno-Salinas 2025].

Dentre essas abordagens, destacam-se as baseadas em jogos digitais [Wang et al. 2022, Videnovik et al. 2023], tanto pelo uso de jogos educacionais como recurso didático ProgramADAs [Vilarinho et al. 2026] [Silva et al. 2022, Gundersen and Lampropoulos 2025], quanto por perspectivas construtivistas que incentivam os próprios estudantes a desenvolverem jogos [DiSalvo et al. 2014, Hoover et al. 2016, Kafai 2006]. Em particular, a criação de jogos mostra-se promissora ao integrar algoritmos, lógica de programação e resolução de problemas a elementos lúdicos, criativos e colaborativos, favorecendo a aprendizagem pela prática e pela experimentação.

Diante desse cenário, este trabalho apresenta um relato de experiência docente no ensino de algoritmos ao longo de vários anos, fundamentado em uma abordagem construtivista na qual os estudantes desenvolvem jogos digitais como estratégia central de aprendizagem. Diferentemente de abordagens que utilizam jogos de forma pontual ou complementar, a proposta estrutura a disciplina a partir do desenvolvimento progressivo de jogos, articulando os conteúdos de programação a desafios práticos de complexidade crescente e a elementos da Aprendizagem Baseada em Problemas (*Problem-Based Learning* – PBL) [Wood 2003], ao posicionar os jogos como problemas abertos e contextualizados. Assim, o desenvolvimento de jogos atua como eixo estruturante da disciplina, meio de construção do conhecimento e instrumento de avaliação. Como contribuição, este artigo descreve e analisa essa abordagem aplicada de forma contínua em diferentes turmas, discutindo sua organização, aplicação em sala e impactos no engajamento e na aprendizagem, de modo a oferecer subsídios para sua replicação e adaptação em outros contextos de ensino de algoritmos no ensino superior.

Este artigo está organizado da seguinte forma: a Seção 2 apresenta a fundamentação que embasa a proposta; a Seção 3, os trabalhos relacionados; a Seção 4, a metodologia adotada; a Seção 5, a discussão dos resultados; a Seção 6, as limitações da abordagem; e a Seção 7, as considerações finais.

2. Fundamentação Teórica

2.1. Ensino de Algoritmos e Dificuldades de Aprendizagem

O ensino de algoritmos e de introdução à programação tem sido reconhecido como um dos principais desafios no ensino superior, especialmente nos primeiros períodos dos cursos das áreas de Computação e STEM. Estudos indicam que as dificuldades enfrentadas pelos estudantes não se limitam à sintaxe das linguagens de programação, mas também estão fortemente associadas à abstração, à decomposição de problemas e à construção de raciocínio algorítmico [Morais et al. 2020, Elegbeleye and Isong 2025]. Nesse contexto, abordagens tradicionais centradas na exposição teórica tendem a agravar essas dificuldades ao oferecer poucas oportunidades de experimentação, contextualização e construção ativa do conhecimento, o que limita o engajamento e a compreensão prática dos conceitos.

2.2. Metodologias Ativas e Aprendizagem Baseada em Problemas

Diante desse cenário, metodologias ativas têm sido propostas para promover maior engajamento e participação discente. Entre elas, a PBL organiza o processo de ensino a partir de problemas autênticos e desafiadores, nos quais os estudantes assumem um papel central na construção do conhecimento [Wood 2003]. Na PBL, o aprendizado ocorre de forma iterativa, envolvendo investigação, colaboração e reflexão, o que favorece o desenvolvimento de habilidades cognitivas de ordem superior, como análise e resolução de problemas, aspectos relacionados ao desenvolvimento do raciocínio algorítmico.

2.3. Jogos Digitais e a Perspectiva Construtivista

Paralelamente, abordagens baseadas em jogos digitais têm ganhado destaque na Educação em Computação. A Aprendizagem Baseada em Jogos (*Game-Based Learning*) [Prensky 2003] utiliza jogos para engajar estudantes e apoiar a compreensão de conceitos, explorando desafios, *feedback* imediato e progressão de dificuldade [Wang et al. 2022, Videnovik et al. 2023], além de proporcionar ambientes seguros para a experimentação, nos quais os erros integram o processo de aprendizagem e contribuem para reduzir a ansiedade e a resistência iniciais frequentes em disciplinas de programação. Uma vertente relevante é a abordagem construtivista que envolve os estudantes no desenvolvimento de seus próprios jogos [Kafai 2006, Kafai and Burke 2015], na qual os alunos deixam de ser consumidores e passam a atuar como criadores, aplicando conceitos de forma integrada, tomando decisões e resolvendo problemas reais. Nesse sentido, o desenvolvimento de jogos pode ser compreendido como uma forma de aprendizagem baseada em projetos, articulando aspectos técnicos e criativos e favorecendo a integração de múltiplos conceitos de programação em atividades significativas.

No ensino de algoritmos, a integração entre PBL e desenvolvimento de jogos mostra-se promissora: enquanto a PBL fornece a estrutura baseada em problemas e investigação, o desenvolvimento de jogos oferece um contexto concreto e motivador para aplicar esses conceitos. Essa combinação favorece a aprendizagem ativa, o engajamento e a construção de conhecimento, ao mesmo tempo em que alinha objetivos de aprendizagem, atividades práticas e avaliação, aproximando conceitos abstratos de situações contextualizadas. Apesar desse potencial, ainda são limitados os trabalhos que sistematizam, de forma estruturada, a integração entre Aprendizagem Baseada em Problemas e desenvolvimento de jogos no ensino de algoritmos no ensino superior, o que torna relevante investigar e documentar experiências que orientem docentes na adoção dessas estratégias, especialmente quando utilizadas de forma contínua e articulada ao longo da disciplina.

3. Trabalhos Relacionados

Jogos digitais no ensino de programação têm sido explorados sob diferentes perspectivas. Uma linha concentra-se no uso de jogos como apoio ao ensino, na qual os estudantes interagem com jogos previamente desenvolvidos. Nesses casos, mesmo sem o envolvimento na construção ativa, os estudantes demonstram níveis significativos de engajamento e motivação [Silva and Dantas 2014, Silva et al. 2021, Silva et al. 2022, Gundersen and Lampropoulos 2025, Vilarinho et al. 2026].

Abordagens baseadas no desenvolvimento de jogos pelos próprios estudantes têm ganhado destaque na literatura [Games and Kane 2011, Denner et al. 2012, DiSalvo et al. 2014, Hoover et al. 2016]. Trabalhos fundamentados na perspectiva construtivista [Kafai 2006, Kafai and Burke 2015] indicam que a criação de jogos pode promover maior engajamento, criatividade e compreensão de conceitos computacionais, uma vez que exige a aplicação prática de conhecimentos, como variáveis, estruturas condicionais e de repetição, em contextos significativos. Além disso, esses estudos evidenciam o desenvolvimento de habilidades, como a resolução de problemas e o Pensamento Computacional [Wing 2006], durante a construção dos jogos.

No entanto, apesar desses avanços, muitos desses estudos tratam o desenvolvimento de jogos como atividade pontual ou projeto isolado dentro da disciplina, sem necessariamente integrá-lo de forma estruturante ao currículo ou articulá-lo com metodologias pedagógicas mais amplas, o que limita a continuidade e o aprofundamento da aprendiza-

gem ao longo do processo formativo. Adicionalmente, observa-se que grande parte dessas experiências ocorre em contextos específicos ou de curta duração, não necessariamente em disciplinas formais de algoritmos no ensino superior.

Diante dessas lacunas, este trabalho se diferencia ao apresentar uma abordagem que posiciona o desenvolvimento de jogos digitais como elemento central da disciplina, articulado aos princípios da PBL e integrado, de forma sistemática, ao planejamento, à execução e à avaliação do processo de ensino-aprendizagem. Diferentemente de abordagens pontuais, a proposta organiza toda a disciplina a partir do desenvolvimento progressivo de jogos, alinhando conteúdos, atividades e avaliação. Ao fazê-lo, busca-se não apenas relatar uma experiência, mas também contribuir para a sistematização de práticas que possam ser replicadas e adaptadas a diferentes contextos educacionais.

4. Metodologia

4.1. Contexto e Caracterização da Amostra

Esta experiência foi conduzida na disciplina Introdução à Programação I do curso de Universidade Federal Rural de Pernambuco (UFRPE), ofertada a alunos do primeiro período da graduação em Ciência da Computação. Trata-se de uma disciplina obrigatória, com foco no desenvolvimento do raciocínio lógico e dos fundamentos de programação em linguagem C, abrangendo desde conceitos introdutórios, como variáveis e estruturas de decisão, até tópicos mais avançados, como ponteiros e manipulação de arquivos. Esta abordagem pedagógica vem sendo aplicada continuamente desde 2015, caracterizando-se como uma estratégia consolidada em diferentes ofertas. Os estudantes têm, em sua maioria, entre 18 e 24 anos e geralmente ingressam sem conhecimento prévio de programação, com turmas de perfis heterogêneos em familiaridade com tecnologia e lógica computacional, o que torna o processo de ensino-aprendizagem particularmente desafiador nos períodos iniciais. Nesse contexto, a disciplina busca equilibrar fundamentos teóricos com atividades práticas que estimulem o engajamento e reduzam a percepção de abstração frequentemente associada ao ensino de algoritmos.

A amostra analisada corresponde às três ofertas realizadas entre 2023 e 2025, com turmas de tamanho e perfil comparáveis, de aproximadamente 50 estudantes matriculados por oferta, predominância da faixa de 18 a 24 anos e sem experiência prévia em programação. A participação na avaliação foi voluntária e anônima, e os 131 respondentes, de um total de 150 matriculados (41 em 2023, 46 em 2024 e 44 em 2025), resultam em uma taxa de resposta global de 87,3% (82% em 2023, 92% em 2024 e 88% em 2025). Não houve seleção amostral por critérios de desempenho, de modo que a amostra reflete o conjunto natural das turmas ao longo dos três anos.

4.2. Organização Pedagógica e Sequência de Jogos

Do ponto de vista pedagógico, a disciplina segue uma estrutura baseada na alternância contínua entre teoria e prática. Essa organização é planejada de forma incremental e progressiva, acompanhando a evolução dos tópicos previstos na ementa. Paralelamente às aulas expositivas, os alunos realizam atividades práticas em laboratório, nas quais são incentivados a aplicar os conceitos aprendidos na resolução de problemas computacionais. Dentro dessa abordagem, o desenvolvimento de jogos é utilizado como estratégia central para contextualizar os conteúdos de algoritmos e de programação.

O uso de jogos consiste em propor desafios práticos mais completos, abstratos e abertos, permitindo que os estudantes desenvolvam diferentes estratégias de resolução

para um mesmo problema. Essa abordagem busca estimular a criatividade, o raciocínio lógico e a autonomia no processo de aprendizagem. Além disso, os jogos favorecem a integração de múltiplos conceitos de programação em uma única atividade prática. Por exemplo, no desenvolvimento do jogo da forca, os alunos precisam utilizar variáveis, estruturas condicionais, de repetição e manipulação de *strings* para implementar a lógica do jogo. À medida que novos conteúdos são introduzidos na disciplina, os desafios propostos também aumentam em complexidade, caracterizando uma progressão pedagógica estruturada, incorporando conceitos como vetores, funções, ponteiros e arquivos.

Com o objetivo de favorecer a replicabilidade da proposta, a Tabela 1 sistematiza a sequência de jogos utilizada e os conceitos de programação associados a cada etapa. A mesma sequência base foi adotada nas três ofertas analisadas (2023 a 2025), preservando a ordem dos conceitos e o nível crescente de complexidade. As variações entre turmas restringiram-se ao projeto final aberto, cujo tema é de livre escolha do estudante, e a eventuais substituições pontuais de jogos equivalentes dentro de uma mesma etapa (por exemplo, o uso do jogo da velha ou do jogo da memória para introduzir matrizes), sem alteração dos conceitos trabalhados.

Tabela 1. Sequência de jogos e conceitos de programação associados

Etapa	Jogos propostos	Conceitos associados
1	Adivinhação numérica, cálculo de pontuação, menus interativos, simulações textuais	Entrada e saída, variáveis, tipos primitivos, operadores aritméticos
2	Pedra-papel-tesoura, batalha naval simplificada, jogo da forca	Estruturas condicionais e de repetição, manipulação de <i>strings</i>
3	Jogo da velha, jogo da memória, jogo da dama	Vetores e matrizes, validação de posições
4	Refatoração dos jogos anteriores em módulos	Funções e modularização
5	Extensões dos jogos (mecânicas mais elaboradas)	Ponteiros e manipulação de memória
6	Quiz com <i>ranking</i> , RPG textual, sistemas de progresso	Manipulação de arquivos, persistência de dados

A avaliação da disciplina é predominantemente prática. Ao longo do semestre, os alunos realizam atividades de programação em laboratório, além de projetos e exercícios relacionados aos conteúdos trabalhados em sala. As avaliações formais são realizadas diretamente no computador e frequentemente usam o desenvolvimento de jogos ou de desafios inspirados em jogos como instrumentos avaliativos. Dessa forma, busca-se avaliar não apenas o domínio sintático da linguagem C, mas também a capacidade dos estudantes de estruturar soluções algorítmicas para problemas mais complexos e contextualizados.

Além das avaliações práticas, a condução da disciplina também envolve observações contínuas sobre o desempenho e engajamento dos estudantes durante as atividades propostas. Essas observações foram consideradas como fonte de dados qualitativos, incluindo participação, dificuldades recorrentes, qualidade das soluções e *feedbacks* informais, e foram utilizadas para apoiar a análise apresentada na Seção 5.

4.3. Instrumento de Avaliação

A percepção dos estudantes foi coletada por meio de um questionário elaborado pelos autores a partir da literatura sobre metodologias ativas e Aprendizagem Baseada em Jogos, organizado em quatro dimensões, motivação e engajamento, aprendizagem e compreensão de conceitos, criatividade e autonomia, e percepção prática da programação, com escala Likert de cinco pontos, de “Discordo totalmente” a “Concordo totalmente”. Além dos itens fechados, incluiu campos abertos para comentários, que deram origem aos *feedbacks* qualitativos discutidos na Seção 5.

O instrumento foi aplicado ao final de cada oferta, entre 2023 e 2025, de forma *online*, voluntária e anônima, após as atividades avaliativas, para não interferir no desempenho dos estudantes. Os dados fechados foram analisados por estatística descritiva, com médias por item e por ano (Tabelas 2 e 3), e, nesta versão, por uma análise cruzada entre ofertas, percepções de aprendizagem e desempenho das turmas, descrita na Seção 5. As respostas abertas foram tratadas por análise de conteúdo, com agrupamento temático dos comentários recorrentes.

Vale salientar que este estudo, conduzido em atividades regulares de sala de aula, não exige submissão a um Comitê de Ética, conforme as resoluções do CNS.

4.4. Relato da Experiência

A disciplina Introdução à Programação I inicia-se com os conceitos fundamentais de algoritmos e lógica de programação. Nas primeiras semanas, os estudantes têm contato com conceitos básicos da linguagem C, como entrada e saída de dados, variáveis, tipos primitivos e operadores aritméticos, já acompanhados de exemplos simples de jogos que evidenciam a aplicação dos conceitos e reduzem a percepção de abstração. À medida que os conteúdos avançam, os exercícios, inicialmente direcionados e de escopo reduzido (adivinhação numérica, menus interativos e simulações textuais), ganham interatividade com as estruturas condicionais e de repetição, e jogos como pedra-papel-tesoura, batalha naval simplificada e forca passam a integrar múltiplos conceitos, exigindo manipulação de *strings*, controle de estados, repetição e condicionais. Jogos com estruturas bidimensionais, como dama, velha e memória, introduzem e consolidam o uso de matrizes.

O desenvolvimento dos projetos ocorre progressivamente: conforme novos conteúdos são apresentados, os alunos expandem ou refatoram seus jogos em um processo iterativo alinhado à proposta construtivista, como a reorganização do código em módulos quando funções são introduzidas. A linguagem utilizada é o C, com ambientes simples de desenvolvimento, e os projetos têm relativa liberdade criativa, com temas geralmente livres, o que favorece o senso de autoria e o engajamento.

As atividades são predominantemente individuais, sobretudo nas etapas iniciais, favorecendo a autonomia, embora ocorram discussões coletivas em momentos específicos. O professor alterna entre momentos expositivos e a atuação como mediador, estimulando os estudantes a refletir sobre suas soluções. As dificuldades mais comuns envolvem abstração lógica, controle de fluxo e compreensão de estruturas de repetição e de ponteiros; nesse contexto, os jogos atuam como elementos motivadores, tornando os problemas mais concretos e reduzindo a resistência inicial. Entre os projetos, destacam-se *quiz* interativo com pontuação, simuladores de combate em turno, jogos de labirinto textual e RPGs simplificados em terminal, incluindo um jogo de aventura textual com decisões que alteravam o fluxo narrativo e um sistema de *ranking* persistente em arquivo de texto.

4.5. Disponibilidade dos Materiais

Com o intuito de favorecer a replicabilidade e a adaptação da proposta em outros contextos, os artefatos utilizados na disciplina, incluindo os enunciados das atividades, os critérios de avaliação, o cronograma de conteúdos e jogos e exemplos de código de referência, encontram-se disponíveis na turma virtual da disciplina no Google Classroom¹. O questionário aplicado também é disponibilizado no mesmo espaço, permitindo sua reutilização por outros docentes e pesquisadores.

5. Resultados e Discussão

Ao longo das diferentes ofertas da disciplina, foi possível observar mudanças no comportamento e no envolvimento dos estudantes com a adoção da abordagem baseada no desenvolvimento de jogos. Em comparação com experiências mais tradicionais, centradas predominantemente em exercícios abstratos e em listas de programação descontextualizadas, percebeu-se um aumento considerável no engajamento dos alunos nas atividades práticas. Isto sugere que a contextualização dos conteúdos por meio de jogos contribuiu para reduzir a abstração e favorecer a participação ativa dos estudantes.

5.1. Engajamento e Motivação dos Estudantes

Uma das principais mudanças observadas foi o aumento da participação dos alunos nas aulas práticas. Os estudantes passaram a demonstrar maior interesse em testar diferentes soluções, explorar novas funcionalidades e discutir estratégias de implementação entre si. Além disso, percebeu-se uma evolução gradual da autonomia ao longo do semestre. Nas primeiras semanas, muitos alunos apresentavam insegurança até mesmo para estruturar pequenos algoritmos, com alta dependência do professor e baixa resolução autônoma. Com a introdução de jogos simples, essa dependência passou a média-alta e a resolução autônoma a média; nos projetos intermediários, a dependência reduziu-se a média e a autonomia elevou-se a média-alta; e, nos projetos finais, tornou-se comum observar estudantes propondo melhorias espontaneamente, adicionando funcionalidades não solicitadas e buscando soluções além do escopo mínimo esperado, com baixa dependência do professor e alta resolução autônoma, o que está alinhado à proposta construtivista.

Os *feedbacks* informais dos estudantes também reforçam essa percepção. Muitos relataram que os jogos ajudaram a tornar os conteúdos menos abstratos e mais acessíveis, especialmente para quem não tem experiência prévia. Comentários recorrentes indicavam que “ver o programa funcionando” aumentava a sensação de progresso e reduzia a frustração normalmente associada às disciplinas introdutórias de programação.

5.2. Resultados do Questionário de Avaliação

Com o objetivo de analisar de forma mais estruturada a percepção dos estudantes sobre a abordagem adotada na disciplina, foi aplicado um questionário de avaliação nas ofertas realizadas entre 2023 e 2025. O instrumento utilizou escala Likert de 1 a 5, variando entre “Discordo totalmente” e “Concordo totalmente”, contemplando aspectos relacionados à motivação, aprendizagem, engajamento e percepção prática da programação. Participaram da avaliação 131 estudantes, sendo 41 em 2023, 46 em 2024 e 44 em 2025.

Quanto ao perfil dos participantes, a maior parte não possuía experiência prévia em programação (73% em 2023, 69% em 2024 e 71% em 2025) e situava-se predominantemente na faixa etária de 18 a 24 anos (92%, 89% e 91%, respectivamente), enquanto

¹<https://classroom.google.com/u/1/c/MTQ4MDEzMDQzNDIy>

o interesse inicial alto ou muito alto pela disciplina cresceu de 38% em 2023 para 41% em 2024 e 44% em 2025. Os resultados quantitativos obtidos (Tabela 2) indicam elevada aceitação da abordagem baseada em jogos, especialmente nos aspectos relacionados à motivação, à criatividade e ao desenvolvimento da lógica de programação, reforçando o papel do desenvolvimento de jogos como estratégia de aprendizagem ativa.

Tabela 2. Resultados quantitativos do questionário

Questão	2023	2024	2025
Jogos tornaram a disciplina mais interessante	4.5	4.7	4.8
Jogos aumentaram a motivação	4.4	4.6	4.7
Projetos reduziram dificuldade inicial	4.1	4.3	4.5
Melhor compreensão de condicionais	4.3	4.5	4.6
Melhor compreensão de repetição	4.4	4.6	4.7
Melhor compreensão de matrizes	4.0	4.3	4.5
Desenvolvimento da lógica	4.5	4.6	4.8
Desenvolvimento de resolução de problemas	4.4	4.6	4.7
Estímulo à criatividade	4.6	4.7	4.8
Confiança ao final da disciplina	4.0	4.2	4.5
Recomendaria a abordagem	4.7	4.8	4.9

Os resultados sugerem um aumento na percepção positiva da disciplina ao longo das ofertas, indicando que a abordagem se mantém eficaz mesmo com diferentes turmas e perfis de estudantes.

5.3. Impactos na Aprendizagem

Observou-se um impacto positivo na compreensão prática dos conceitos. Em vez de memorizar estruturas sintáticas isoladas, os estudantes passaram a associá-las a problemas concretos nos jogos desenvolvidos, em consonância com abordagens construtivistas que defendem a aprendizagem pela construção de artefatos significativos. Estruturas condicionais passaram a representar regras do jogo e controle de estados, estruturas de repetição ganharam sentido em mecânicas contínuas de interação e vetores e matrizes passaram a ser compreendidos como elementos para representar tabuleiros, mapas e estados do jogo. A contextualização por meio de jogos como dama, memória e velha mostrou-se particularmente relevante para reduzir as dificuldades de abstração associadas a estruturas bidimensionais, que em experiências anteriores geravam altos índices de erro.

A evolução da complexidade dos projetos reflete esse amadurecimento: partindo de um nível inicial baixo, cresceu progressivamente ao longo das etapas básica e intermediária, atingindo níveis elevados na etapa final, quando os estudantes já articulavam de forma autônoma diferentes estruturas e mecânicas. Além da evolução técnica, observou-se melhora na resolução de problemas, com os estudantes decompondo problemas em etapas menores, identificando padrões de solução e reutilizando estratégias, o que caracteriza o desenvolvimento do Pensamento Computacional. Essa progressão foi acompanhada por uma elevação consistente da satisfação geral, de 4.2 em 2023 para 4.5 em 2024 e 4.8 em 2025. A Tabela 3 apresenta a percepção sobre a evolução da aprendizagem nesse período.

Entre os projetos desenvolvidos, destacam-se *quiz* interativo com pontuação, simuladores de combate em turno, jogos de labirinto textual e versões simplificadas de RPGs em terminal, alguns com sistemas de *ranking* persistente em arquivos de texto,

Tabela 3. Percepção da evolução da aprendizagem

Aspecto Avaliado	2023	2024	2025
Engajamento dos estudantes	4.3	4.6	4.8
Participação nas aulas práticas	4.4	4.6	4.7
Autonomia na resolução de problemas	3.9	4.2	4.5
Qualidade dos projetos finais	4.0	4.3	4.6
Integração de múltiplos conceitos	4.1	4.4	4.6

integrando múltiplos conteúdos em um único projeto. De maneira geral, os resultados indicam que o desenvolvimento de jogos aumenta o engajamento, a motivação e a compreensão prática dos conceitos de algoritmos, reforçando o potencial de abordagens construtivistas e baseadas em problemas para reduzir a abstração e estimular a criatividade, a autonomia e a aprendizagem ativa.

5.4. Análise Cruzada entre Ofertas, Aprendizagem e Percepções

Para além da leitura item a item, procurou-se examinar a relação entre as ofertas, o desempenho das turmas e as percepções registradas, articulando os dados quantitativos do questionário às observações dos professores. A Tabela 4 organiza, por ano, a média de desempenho das turmas, as percepções de engajamento e de autonomia (extraídas da Tabela 3) e a satisfação geral, permitindo uma leitura bivariada da evolução da abordagem.

Tabela 4. Análise cruzada por oferta: desempenho, percepções e satisfação

Indicador	2023	2024	2025
Nota média da turma (0–10)	8.2	8.4	8.8
Aprovação na disciplina (%)	91	95	95
Percepção de engajamento (1–5)	4.3	4.6	4.8
Percepção de autonomia (1–5)	3.9	4.2	4.5
Satisfação geral (1–5)	4.2	4.5	4.8

A leitura conjunta desses indicadores sugere melhora consistente ao longo das ofertas. A nota média evoluiu de 8.2 em 2023 para 8.4 em 2024 e 8.8 em 2025, e a taxa de aprovação subiu de 91% para 95%, acompanhando a elevação das percepções de engajamento, autonomia e satisfação geral. Associada às observações docentes de maior participação e de projetos finais mais sofisticados, essa convergência reforça a interpretação de que a maturação da abordagem favoreceu tanto a experiência quanto a aprendizagem. Ressalva-se, contudo, que se trata de dados agregados de três ofertas, o que confere caráter exploratório à leitura bivariada e não permite isolar o efeito da abordagem de outros fatores.

5.5. Feedbacks qualitativos dos estudantes

Os comentários positivos recorrentes indicam que a abordagem contribuiu para tornar os conteúdos mais acessíveis, práticos e motivadores para os estudantes. Vários relataram que “os jogos fizeram programação parecer menos abstrata” e que foi “a primeira vez que consegui visualizar utilidade prática dos algoritmos”. Destacaram ainda que “criar jogos tornou as atividades mais divertidas e motivadoras” e que “mesmo sem experiência prévia consegui acompanhar a disciplina”.

Por outro lado, as principais dificuldades relatadas pelos estudantes estão associadas à complexidade técnica inicial e ao desenvolvimento dos projetos, com destaque para a complexidade inicial da linguagem C, a dificuldade na depuração de erros, o gerenciamento do escopo dos projetos, a compreensão de ponteiros e manipulação de memória e o tempo necessário para finalizar projetos mais complexos.

6. Limitações da Abordagem

Apesar dos resultados positivos, a abordagem também apresentou alguns pontos de atenção. Como os projetos têm maior complexidade e liberdade criativa, parte dos estudantes se beneficia de acompanhamento mais próximo, sobretudo nas etapas iniciais do semestre, e a mediação docente ajuda a equilibrar a criatividade com o nível técnico adquirido, evitando escopos ambiciosos demais. A depuração também pode ser mais desafiadora, já que os jogos envolvem múltiplas estruturas e estados de execução, o que abre espaço para estratégias específicas de apoio. De forma geral, o desenvolvimento de jogos como eixo estruturante traz benefícios significativos e se potencializa quando acompanhado de suporte docente e planejamento pedagógico adequados, especialmente em turmas iniciais.

7. Considerações Finais

Este trabalho apresentou uma experiência de ensino de Introdução à Programação I baseada no desenvolvimento de jogos em linguagem C, na qual os jogos deixam de ser recurso pontual para se tornarem o eixo estruturante da disciplina, organizada em torno de sua construção progressiva. Ao longo das diferentes ofertas, observaram-se impactos positivos no engajamento, na motivação e na participação, sobretudo entre alunos sem experiência prévia, além de melhora na compreensão prática dos conteúdos, na resolução de problemas e na autonomia no desenvolvimento de projetos. Os questionários aplicados entre 2023 e 2025 confirmam essas percepções, com altos índices de satisfação quanto à aprendizagem, à criatividade e à utilidade prática dos conteúdos. A análise conjunta dos dados quantitativos e qualitativos sugere que o desenvolvimento de jogos, articulado a uma abordagem construtivista e baseada em problemas, favorece uma aprendizagem mais ativa, contextualizada e significativa, mostrando-se promissor para reduzir a abstração no ensino de programação nas etapas iniciais da formação em Computação. Como principal contribuição, o trabalho sistematiza uma abordagem pedagógica aplicada de forma contínua em diferentes turmas, evidenciando como o desenvolvimento progressivo de jogos pode ser integrado ao ensino de algoritmos como elemento estruturante, e não apenas complementar.

Reconhece-se que as conclusões apoiam-se sobretudo na percepção dos estudantes e na observação docente, limitação inerente a relatos de experiência. Como trabalhos futuros, pretende-se incorporar indicadores objetivos de aprendizagem (notas, aprovação, pré e pós-testes e evolução da complexidade do código), realizar comparações mais controladas com abordagens tradicionais e aplicar análises estatísticas mais rigorosas apoiadas por instrumentos validados, direções que fortalecem a base empírica da proposta e ampliam sua generalização a outros contextos de ensino de algoritmos.

Declaração sobre uso de Inteligência Artificial

Ferramentas de Inteligência Artificial foram utilizadas exclusivamente para apoio editorial. O *Claude* auxiliou na revisão editorial. Todas as decisões de pesquisa, métodos, análises e interpretações foram tomadas integralmente pelos autores, sem influência dessas ferramentas sobre o conteúdo científico ou as conclusões do trabalho.

Referências

- Córdova-Esparza, D.-M., Romero-González, J.-A., Córdova-Esparza, K.-E., Terven, J., and López-Martínez, R.-E. (2024). Active learning strategies in computer science education: A systematic review. *Multimodal Technologies and Interaction*, 8(6):50.
- Denner, J., Werner, L., and Ortiz, E. (2012). Computer games created by middle school girls: Can they be used to measure understanding of computer science concepts? *Computers & Education*, 58(1):240–249.
- Díaz-Lauzurica, B. and Moreno-Salinas, D. (2025). Active learning methodologies for increasing the interest and engagement in computer science subjects in vocational education and training. *Education Sciences*, 15(8):1017.
- DiSalvo, B., Guzdial, M., Bruckman, A., and McKlin, T. (2014). Saving face while geeking out: Video game testing as a justification for learning computer science. *Journal of the Learning Sciences*, 23(3):272–315.
- Elegbeleye, F. and Isong, B. (2025). A systematic review of challenges in teaching and learning computer programming modules. *The Indonesian Journal of Computer Science*, 14(1).
- Games, A. and Kane, L. (2011). Exploring adolescent’s stem learning through scaffolded game design. In *Proceedings of the 6th international conference on foundations of digital games*, pages 1–8.
- Gundersen, S. W. and Lampropoulos, G. (2025). Using serious games and digital games to improve students’ computational thinking and programming skills in k-12 education: A systematic literature review. *Technologies*, 13(3):113.
- Hoover, A. K., Barnes, J., Fatehi, B., Moreno-León, J., Puttick, G., Tucker-Raymond, E., and Hartevel, C. (2016). Assessing computational thinking in students’ game designs. In *Proceedings of the 2016 annual symposium on computer-human interaction in play companion extended abstracts*, pages 173–179.
- Kafai, Y. B. (2006). Playing and making games for learning: Instructionist and constructionist perspectives for game studies. *Games and culture*, 1(1):36–40.
- Kafai, Y. B. and Burke, Q. (2015). Constructionist gaming: Understanding the benefits of making games for learning. *Educational psychologist*, 50(4):313–334.
- Morais, C. G. B., Neto, F. M. M., and Osório, A. J. M. (2020). Dificuldades e desafios do processo de aprendizagem de algoritmos e programação no ensino superior: uma revisão sistemática de literatura. *Research, Society and Development*, 9(10):e9429109287–e9429109287.
- Prensky, M. (2003). Digital game-based learning. *Computers in entertainment (CIE)*, 1(1):21–21.
- Silva, A. L., Nascimento, G. S., Pereira, J. G., Neto, L. M., Araújo, M. E., and de Oliveira, V. A. (2022). Codeland: um jogo para o ensino de lógica de programação com Python. In *Anais do XXI Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames)*, pages 643–652. SBC.
- Silva, M. A. and Dantas, A. (2014). KLouro: Um jogo educacional para motivar alunos iniciantes em programação. In *Anais do XXV Simpósio Brasileiro de Informática na Educação (SBIE)*, pages 702–711. SBC.

- Silva, R. R., Rivero, L., and dos Santos, R. P. (2021). Programse: Um jogo para aprendizagem de conceitos de lógica de programação. *Revista Brasileira de Informática na Educação*, 29:301–330.
- Videnovik, M., Vold, T., Kiønig, L., Madevska Bogdanova, A., and Trajkovik, V. (2023). Game-based learning in computer science education: a scoping literature review. *International Journal of STEM Education*, 10(1):54.
- Vilarinho, L. L. A., Quintela, B. d. M., Chaves, L. J., Valle, P. H. D., and Oliveira, A. M. d. (2026). A digital game to motivate and empower women in learning algorithms for computer science. *Journal on Interactive Systems*, 17(1):126–142.
- Wang, L.-H., Chen, B., Hwang, G.-J., Guan, J.-Q., and Wang, Y.-Q. (2022). Effects of digital game-based stem education on students’ learning achievement: A meta-analysis. *International Journal of STEM Education*, 9(1):26.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.
- Wood, D. F. (2003). Problem based learning. *Bmj*, 326(7384):328–330.