

Aprendizagem por Projetos no Ensino Técnico Integrado: relato do desenvolvimento colaborativo de um *Snake Game*

Ianne Lima Nogueira¹, Anderson Veiga da Silva¹, Carlos Eduardo de Oliveira Barbosa¹, Ian Motta Ferreira¹, Ray Gomes Pereira¹, Lucas Menezes Fasolo¹, Fellipe Augusto Oliveira Jardim¹, João Victor Figueiredo Moura Gomes¹, Ana Mara de Oliveira Figueiredo¹

¹Instituto Federal de Educação, Ciência e Tecnologia Fluminense
Bom Jesus do Itabapoana – RJ – Brasil

{ianne.nogueira, anderson.silva, ana.figueiredo}@iff.edu.br,
{cadubars007, ianmfpro, raygomespereira022, fasololucas18,
felliipea542, joaovictorfigueredomouragomes}@gmail.com,

Abstract. *Teaching programming in basic education fosters the development of computational thinking and creativity. This paper reports the experience of six first-year students from the Integrated Technical Course in Informatics who collaboratively developed a Snake Game using Python, Pygame, Flask, and SQLite. The project, based on project-based learning, integrated programming and database concepts into a practical and interdisciplinary proposal. The game was presented at an academic event and evaluated by the audience, who showed high engagement and interest in programming. The results confirm the potential of active learning methodologies to promote student authorship and effective learning.*

Resumo. *O ensino de programação na educação básica favorece o desenvolvimento do pensamento computacional e da criatividade. Este artigo relata a experiência de seis alunos do primeiro ano do Curso Técnico em Informática Integrado ao Ensino Médio no desenvolvimento de um Snake Game com Python, Pygame, Flask e SQLite. A atividade, baseada na aprendizagem por projetos, integrou programação e banco de dados em uma proposta prática e interdisciplinar. O jogo foi apresentado em evento acadêmico e avaliado pelo público que demonstrou alto engajamento e interesse em programação. Os resultados confirmam o potencial das metodologias ativas para promover autoria discente e aprendizagem efetiva.*

1. Introdução

O ensino de programação na educação básica tem ganhado relevância como estratégia para o desenvolvimento de competências essenciais no século XXI, como o pensamento computacional, a resolução de problemas, a criatividade e o trabalho colaborativo [Oliveira et al. 2023; Wing 2006]. Essas habilidades, cada vez mais demandadas em uma sociedade mediada por tecnologias, são reconhecidas pela literatura como fundamentais para a formação de cidadãos críticos e criadores de soluções tecnológicas [Demo 2008; Brasil 2018]. Nesse contexto, Resnick [2017] defende que a aprendizagem deve se apoiar em quatro pilares (projetos, paixão, pares e brincadeiras), enfatizando a

importância de experiências criativas e colaborativas na formação de indivíduos capazes de pensar e agir de forma inovadora.

De acordo com a Base Nacional Comum Curricular (BNCC) [BRASIL 2018], o pensamento computacional deve ser estimulado desde os anos iniciais da educação básica, promovendo a autonomia, o raciocínio lógico e a compreensão crítica das tecnologias digitais. O pensamento computacional, entretanto, não se limita a um conjunto de habilidades técnicas, mas constitui uma forma de abordar problemas de maneira estruturada, envolvendo abstração, decomposição, reconhecimento de padrões e criação de algoritmos [Brackmann et al. 2017; Shute, Sun e Asbell-Clarke 2017]. Essa perspectiva reforça a necessidade de práticas pedagógicas que permitam aos estudantes aplicarem esses conceitos em situações concretas, integrando teoria e prática em contextos significativos.

Apesar da crescente valorização dessas competências, a inserção da programação no currículo escolar ainda enfrenta desafios, como a escassez de recursos didáticos adequados, a formação docente insuficiente e a dificuldade em manter o interesse dos alunos em conteúdos abstratos. Nesse cenário, metodologias ativas de ensino, como a Aprendizagem Baseada em Projetos (ABP) e a abordagem *maker*, têm se mostrado alternativas eficazes para tornar o processo mais envolvente, criativo e centrado no estudante. Martinez e Stager [2013] defendem que experiências “mão na massa” favorecem a autoria, a experimentação e o protagonismo discente, aspectos fundamentais para a aprendizagem significativa. Evidências recentes corroboram essa perspectiva: Alves et al. [2025] demonstraram que a ABP aplicada ao desenvolvimento de jogos digitais promoveu crescimento médio de 61% nas habilidades de pensamento computacional de estudantes do ensino fundamental, reforçando o potencial dessa abordagem para diferentes contextos da educação básica.

A criação de jogos digitais insere-se nesse contexto como uma estratégia potente de ensino, capaz de aliar o aprendizado de lógica de programação à ludicidade e à resolução criativa de problemas. Papert [1980] já afirmava que programar é, em si, uma forma poderosa de aprender a pensar, especialmente quando o aprendiz se torna criador de tecnologia, e não mero consumidor. Desenvolver jogos possibilita que os estudantes explorem ciclos de planejamento, teste, erro e iteração, integrando saberes técnicos e expressivos em um mesmo processo de criação.

Este artigo relata a experiência de seis estudantes do primeiro ano do Curso Técnico em Informática Integrado ao Ensino Médio, com idades entre 15 e 16 anos, que desenvolveram colaborativamente um jogo da cobrinha utilizando a linguagem Python e as tecnologias Pygame, SQLite e Flask. O projeto foi realizado no âmbito das disciplinas de Introdução à Programação de Computadores e Banco de Dados, sob orientação docente, integrando conceitos de programação, design e persistência de dados em um produto interativo com identidade visual inspirada no Instituto Federal Fluminense.

A atividade, conduzida ao longo de oito semanas, adotou a metodologia de aprendizagem baseada em projetos, com uso da plataforma Replit para programação simultânea e colaboração em tempo real. O processo incluiu etapas como criação do menu, sistema de login, placar de pontuação, efeitos visuais e sonoros, além da integração com banco de dados e interface web. A experiência permitiu aos estudantes vivenciarem um ciclo completo de desenvolvimento de software e consolidar

competências técnicas e socioemocionais, como trabalho em equipe, comunicação e autonomia.

Nesse cenário, o ensino de programação ganha ainda mais força quando articulado a propostas interdisciplinares que unam diferentes componentes curriculares e permitam a aplicação prática dos conhecimentos em projetos concretos. Essa perspectiva dialoga com o currículo da educação profissional técnica de nível médio [BRASIL 2021], que enfatiza a articulação entre teoria e prática e valoriza a aprendizagem contextualizada. Assim, o desenvolvimento colaborativo do jogo digital mostrou-se uma estratégia eficaz para promover engajamento, protagonismo e aprendizagem significativa, ao mesmo tempo em que despertou o interesse dos participantes, e posteriormente, do público que jogou o protótipo durante um evento acadêmico, pelo universo da programação e da tecnologia.

2. Trabalhos Relacionados

A utilização de jogos digitais como ferramenta pedagógica para o ensino de lógica de programação tem sido amplamente explorada em contextos educacionais que valorizam a autonomia e o protagonismo discente. Tais iniciativas alinham-se à concepção de pensamento computacional como competência transversal, essencial para a resolução criativa de problemas e a formação de sujeitos capazes de compreender e transformar o mundo mediado por tecnologias [Wing 2006; Brennan e Resnick 2012]. Nesse cenário, metodologias ativas que colocam o estudante no centro do processo de aprendizagem, como a aprendizagem baseada em projetos e o construcionismo, têm demonstrado efetividade ao promover engajamento, autoria e aprendizagem significativa [Papert 1980; Resnick 2017; Bacich e Moran 2018].

Diversos estudos destacam abordagens inovadoras que integram o ensino de programação ao desenvolvimento de jogos digitais, estimulando a criatividade, a resolução de problemas e o trabalho colaborativo. Um exemplo é o trabalho de Silva et al. [2024], que apresenta a metodologia G-Thinking, voltada à promoção da autonomia discente por meio da cultura da criação prática. A proposta valoriza a realização de projetos tecnológicos idealizados e construídos pelos próprios estudantes, em um ambiente de experimentação contínua, alinhando-se às premissas da aprendizagem ativa e colaborativa.

De modo complementar, Petri et al. [2015] investigam o uso de Game Jams em ambientes escolares a partir da abordagem construcionista. Nessa proposta, os alunos são desafiados a desenvolver jogos digitais em tempo limitado, o que estimula a cooperação, o pensamento criativo e o aprendizado baseado na experimentação. Os autores destacam que a participação ativa no processo de criação favorece o protagonismo e o aprofundamento conceitual em lógica de programação e design de jogos.

No mesmo sentido, Abreu et al. [2023] descrevem o Poké Prog, um jogo educacional voltado ao ensino de programação de forma lúdica e contextualizada. Utilizando referências narrativas e visuais familiares aos estudantes, o trabalho mostra como o uso de elementos do universo dos jogos pode facilitar a compreensão de estruturas de controle e lógica, tornando o conteúdo mais acessível e atrativo, especialmente para iniciantes. Em direção semelhante, Pinheiro et al. [2025] apresentam o Show da Programação, jogo quiz desenvolvido em um instituto federal para o ensino

de lógica de programação com Python, avaliado positivamente por estudantes por meio do modelo MEEGA+. Diferentemente dessas propostas, em que o jogo é o meio pelo qual o aluno aprende, o presente trabalho adota uma perspectiva construcionista: são os próprios estudantes que desenvolvem o jogo, tornando-se autores da tecnologia e não apenas usuários.

Além disso, Brackmann et al. [2017] propõem um currículo estruturado para o ensino de pensamento computacional na educação básica brasileira, defendendo que a aprendizagem deve ocorrer por meio de projetos contextualizados que integrem aspectos técnicos e socioemocionais. Os autores enfatizam que o ensino de programação deve estimular habilidades como trabalho em equipe, comunicação, criatividade e autonomia, articulando-se a objetivos mais amplos da formação integral do estudante.

Apesar do avanço das pesquisas nessa área, observa-se que iniciativas que articulam múltiplas tecnologias, como o uso conjunto de bibliotecas Pygame, frameworks como Flask e bancos de dados SQLite, ainda são pouco relatadas na literatura educacional, sobretudo quando os próprios alunos assumem o protagonismo no desenvolvimento técnico e criativo do projeto. Genesio et al. [2024], em mapeamento sistemático sobre jogos educacionais digitais em Educação em Computação, confirmam que as áreas de Lógica de Programação e Engenharia de Software concentram a maior parte das iniciativas, mas que experiências envolvendo múltiplas tecnologias integradas permanecem sub-representadas. Essa lacuna é relevante, pois evidencia a necessidade de relatos que abordem não apenas o produto final, mas o processo formativo e colaborativo envolvido em sua construção.

Nesse sentido, o presente trabalho busca contribuir com a literatura ao apresentar uma experiência pedagógica baseada na construção colaborativa de um jogo digital desenvolvida por estudantes do ensino técnico integrado, utilizando ferramentas acessíveis e de código aberto. A proposta reforça o potencial transformador de práticas que integram aprendizagem técnica, autoria estudantil e metodologias ativas, demonstrando como o desenvolvimento de jogos digitais pode funcionar como ponte entre o ensino de programação, a criatividade e o engajamento em contextos educacionais reais.

3. Metodologia

A proposta pedagógica foi desenvolvida com seis estudantes do primeiro ano do Curso Técnico em Informática Integrado ao Ensino Médio, com idades entre 15 e 16 anos, no contexto das disciplinas de Introdução à Programação de Computadores e Banco de Dados. O projeto buscou integrar conteúdos dessas duas áreas por meio de uma atividade prática, colaborativa e centrada em metodologias ativas de aprendizagem, nas quais os próprios alunos protagonizam a construção do conhecimento. A tarefa principal consistiu no desenvolvimento do clássico jogo da cobrinha (Snake Game), utilizando a linguagem Python e ferramentas gratuitas e amplamente disponíveis, com o objetivo de promover uma aprendizagem significativa, contextualizada e alinhada à realidade dos estudantes.

As aulas ocorreram ao longo de oito semanas, com encontros semanais de quatro aulas de 50 minutos, organizados em dois momentos complementares: a introdução teórica de conceitos fundamentais e a prática supervisionada em grupo. A carga horária da disciplina foi compartilhada entre diferentes equipes da turma, cada uma

desenvolvendo um projeto distinto, sendo o jogo da cobrinha elaborado por um grupo de seis estudantes. Os professores atuaram como mediadores do processo, propondo desafios intermediários, incentivando a tomada de decisões técnicas e conduzindo momentos de socialização e revisão de código em turma. A avaliação da atividade foi multidimensional, contemplando tanto aspectos técnicos (funcionalidade do jogo, clareza e organização do código) quanto formativos (colaboração, participação e reflexão sobre o processo). Foram ainda utilizados instrumentos complementares, como autoavaliações e avaliações entre pares, que favoreceram a análise crítica e a corresponsabilidade entre os integrantes do grupo.

A adoção da ABP teve como objetivo promover uma aprendizagem contextualizada, em que os conhecimentos de programação e banco de dados fossem mobilizados na resolução de um problema autêntico de desenvolvimento de software. Esperava-se que os estudantes assumissem papel protagonista durante todas as etapas do projeto, desde o planejamento até a implementação e os testes, favorecendo a integração entre teoria e prática, o desenvolvimento do pensamento computacional e a construção colaborativa do conhecimento.

O desenvolvimento ocorreu de forma incremental. Inicialmente, os estudantes definiram a mecânica do jogo e implementaram sua lógica básica em Python. Em seguida, foram incorporadas gradualmente funcionalidades relacionadas à interface gráfica, à persistência de dados e à integração entre as tecnologias utilizadas. A etapa final compreendeu testes, refinamentos e preparação da versão apresentada durante o evento institucional. Ao final de cada etapa, eram realizadas revisões coletivas do código e redefinidas as tarefas para a etapa seguinte.

Durante o desenvolvimento, foram abordados conteúdos como lógica de programação, estruturas condicionais, funções, controle de fluxo, fundamentos de banco de dados e desenvolvimento de interfaces web. O processo foi dividido em etapas sequenciais, com os alunos se organizando em equipes responsáveis por diferentes frentes de trabalho, como lógica do jogo, design gráfico, sonorização, interface web e banco de dados. Logo no início, o grupo definiu coletivamente as características desejadas para o produto final, estabelecendo critérios como jogabilidade simples, desafio progressivo, estética atrativa e recompensa clara. A identidade visual do jogo foi inspirada nas cores e símbolos do Instituto Federal Fluminense, o que reforçou o sentimento de pertencimento e autoria do grupo. Embora cada estudante tivesse uma frente principal de trabalho, a colaboração contínua entre os membros permitiu que todos compreendessem as diferentes camadas do projeto, favorecendo uma visão integrada do desenvolvimento.

Uma das primeiras decisões técnicas envolveu a escolha da plataforma de desenvolvimento. Foram avaliados o Visual Studio Code e o Replit, considerando critérios como facilidade de acesso, desenvolvimento colaborativo e compatibilidade com as tecnologias empregadas no projeto. Optou-se pelo Replit por possibilitar edição simultânea e colaboração em tempo real. Apesar das limitações da versão gratuita, como o tempo reduzido de execução das sessões e a limitação de até três estudantes editando simultaneamente o projeto, o grupo reorganizou o fluxo de desenvolvimento, alternando os responsáveis pela edição do código em um mesmo projeto compartilhado. Essa estratégia permitiu integrar continuamente as contribuições dos diferentes integrantes e contornar as restrições da plataforma, favorecendo o trabalho colaborativo. Embora

ferramentas de Inteligência Artificial Generativa estivessem disponíveis como recurso de apoio, o desenvolvimento do jogo foi conduzido de forma predominantemente autoral, com os estudantes sendo encorajados a buscar soluções por meio da experimentação, da consulta à documentação das tecnologias utilizadas e da colaboração entre pares.

A fase de prototipagem teve início com o uso da biblioteca Pygame, selecionada por sua interface acessível e documentação robusta, adequada a iniciantes. Nessa etapa, os alunos implementaram a lógica principal do jogo: movimentação da cobrinha, detecção de colisões, geração aleatória de itens e pontuação progressiva. Em seguida, foram incorporadas funcionalidades adicionais, como menu inicial, botão de pausa, *sprites* personalizadas, efeitos visuais e sonoros, estes últimos criados pelos próprios estudantes, com exceção da trilha sonora principal, proveniente de um repositório livre. O termo *sprite*, nesse contexto, refere-se a imagens bidimensionais que representam os elementos gráficos do jogo, como personagens e objetos animados, e que são manipuladas em tempo real pela biblioteca Pygame.

Na etapa seguinte, foi desenvolvido um sistema de autenticação de usuários e registro de pontuações. Inicialmente, os dados eram armazenados em arquivos .txt, mas à medida que os alunos avançaram tecnicamente, migraram para um sistema baseado no banco de dados SQLite, integrado a uma interface web construída com o microframework Flask. As páginas de cadastro, login e exibição de ranking foram adaptadas a partir de templates disponibilizados na documentação do Flask, permitindo que os alunos concentrassem esforços na lógica da aplicação e na manipulação dos dados. O uso combinado dessas tecnologias possibilitou uma compreensão integrada do funcionamento de diferentes camadas de uma aplicação real.

O produto final reuniu todos esses componentes em uma aplicação funcional, com interface gráfica, persistência de dados e interatividade via navegador local (127.0.0.1:5000). Essa arquitetura representou um avanço significativo no repertório técnico dos estudantes, que puderam vivenciar o ciclo completo de desenvolvimento de software, ou seja, da concepção à entrega do produto. Além disso, o projeto consolidou competências socioemocionais como comunicação, cooperação, planejamento e autonomia, características centrais das metodologias ativas de ensino [Martinez e Stager 2013; Resnick 2017].

Como etapa complementar, o jogo foi apresentado ao público durante um evento acadêmico de divulgação científica, ocasião em que estudantes de outros cursos (n=13) puderam jogar e responder a um breve questionário sobre usabilidade, jogabilidade e interesse em programação. Essa ação ampliou o alcance pedagógico da experiência, possibilitando aos autores receber feedback externo sobre o produto e refletir sobre melhorias e desdobramentos futuros.

Mais do que a construção de um artefato digital, a experiência representou um processo formativo completo, no qual a autoria e o protagonismo dos estudantes foram fortalecidos em todas as etapas. A vivência prática, o trabalho colaborativo e a exposição pública do resultado final configuraram um percurso de aprendizagem ativo, reflexivo e significativo, aspectos que serão aprofundados na seção seguinte, dedicada aos resultados e à discussão da experiência.

4. Resultados e Discussão

A atividade resultou na criação de um jogo funcional e interativo, que integra interface gráfica, lógica de jogo e persistência de dados. Ao longo do projeto, os seis estudantes do primeiro ano do curso participaram de todas as etapas do ciclo de desenvolvimento de software, desde a concepção da ideia até a entrega do produto final. Esse processo possibilitou a aplicação integrada de conhecimentos de programação em Python, persistência de dados com SQLite e desenvolvimento web com Flask, além de proporcionar experiências de resolução colaborativa de problemas, tomada de decisão em grupo e organização do trabalho durante o desenvolvimento do projeto.

Esses resultados evidenciam o potencial das metodologias ativas de aprendizagem, especialmente da aprendizagem baseada em projetos, no ensino de programação e banco de dados. Ao serem desafiados a conceber e implementar um produto real, os estudantes passaram a compreender a interdependência entre conceitos teóricos e práticos, mobilizando conhecimentos prévios e novos em situações autênticas. Esse tipo de abordagem aproxima o ensino técnico da prática profissional e contribui para a formação de sujeitos autônomos, criativos e colaborativos [Martinez e Stager 2013; Resnick 2017].

O produto final foi apresentado e disponibilizado para testes durante a Mostra do Conhecimento, evento acadêmico anual do Instituto Federal Fluminense – *campus* Bom Jesus, atraindo estudantes de outros cursos. A inclusão de um placar com ranking dos melhores desempenhos despertou o interesse do público e gerou um envolvimento lúdico, transformando a atividade em uma experiência interativa e imersiva. Esse aspecto competitivo ampliou a visibilidade do projeto e fortaleceu o reconhecimento da autoria discente, valorizando o trabalho desenvolvido pelos estudantes e incentivando sua participação durante a apresentação do projeto. Mais do que uma vitrine do produto, a exposição pública constituiu uma etapa formativa do processo: ao apresentar e defender seu trabalho perante outros estudantes, os desenvolvedores vivenciaram de forma autêntica o ciclo completo de criação tecnológica, incluindo a recepção crítica do que produziram.

Durante a exposição, foi aplicado um questionário de percepção do público ($n = 13$), com seis afirmações avaliadas em escala Likert de cinco pontos, instrumento amplamente utilizado na avaliação de jogos educacionais em Computação [Petri et al. 2019], além de duas questões abertas sobre pontos fortes e sugestões de melhoria. As respostas, apresentadas na Figura 1, indicam alta aceitação em todos os aspectos avaliados. Os participantes relataram que o jogo foi fácil de entender ($M = 4,31$), divertido e desafiador ($M = 4,69$) e possuía uma interface visual clara e atrativa ($M = 4,46$). O interesse em continuar jogando também obteve média elevada ($M = 4,46$), reforçando o caráter envolvente da proposta. Por fim, a afirmação “o jogo contribuiu para meu interesse em programação” apresentou média $M = 4,00$, sugerindo que a experiência despertou curiosidade e motivação em relação à computação, mesmo entre participantes externos ao curso.

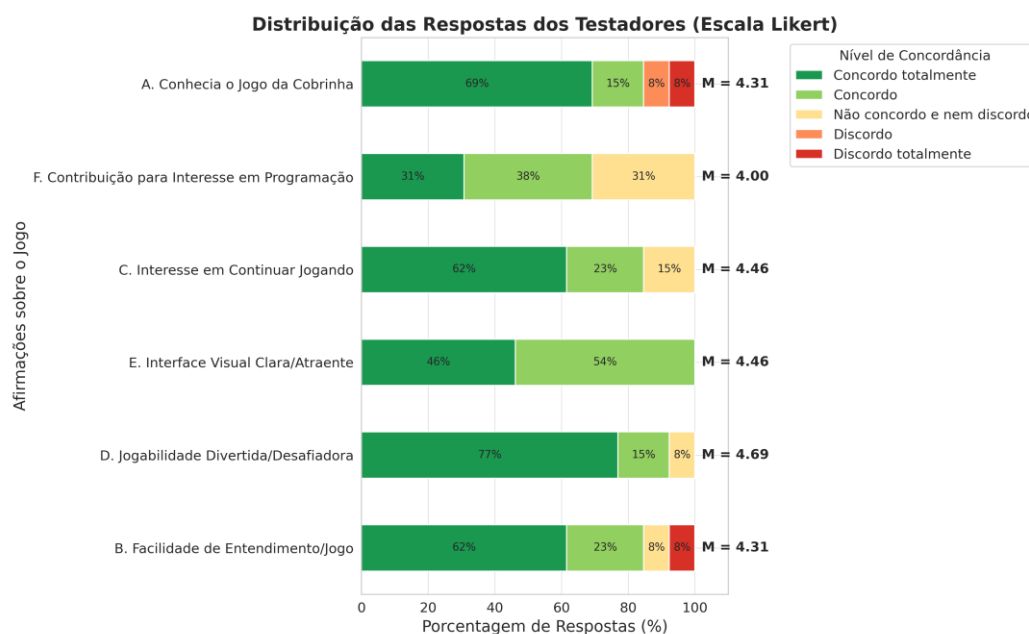


Figura 1 – Distribuição das respostas dos participantes em escala Likert. Observa-se predominância de concordância nas seis afirmações, com médias acima de 4,0, destacando-se a jogabilidade (M = 4,69) e a clareza visual (M = 4,46).

As respostas abertas reforçaram essa percepção positiva. Entre os aspectos mais elogiados, destacaram-se a trilha sonora, a dinâmica dos poderes e o placar com ranking, além de comentários nostálgicos que associaram o jogo à infância dos participantes. As principais sugestões de melhoria incluíram ajustes na velocidade da cobrinha, aumento do número de itens colecionáveis e aprimoramentos visuais. Essas observações serviram como base para uma nova iteração do jogo, em que os próprios alunos implementaram uma mecânica revisada de colisões e bordas, tornando a jogabilidade mais complexa e estimulando a experimentação criativa.

Do ponto de vista pedagógico, a avaliação externa representou uma etapa de consolidação da aprendizagem, permitindo que os alunos percebessem o impacto real de seu trabalho. O feedback do público gerou motivação adicional e senso de pertencimento, reforçando a importância de momentos de socialização do conhecimento. Essa vivência extrapolou o espaço da sala de aula, transformando-se em uma experiência de aprendizagem expandida, que conectou a comunidade escolar à produção tecnológica dos estudantes.

No plano técnico, foi possível observar uma progressiva apropriação de conceitos inicialmente pouco dominados, como modularização, estruturas de repetição e controle de exceções. Esse avanço foi acompanhado qualitativamente pelos professores, por meio de observações diretas e revisões de código realizadas ao longo do desenvolvimento do projeto. A colaboração contínua também favoreceu a adoção de boas práticas de programação, como indentação adequada, padronização e comentários explicativos, frequentemente negligenciadas em etapas iniciais do ensino.

Do ponto de vista dos próprios desenvolvedores, o processo evidenciou avanços perceptíveis na compreensão desses conceitos antes abstratos. A necessidade de integrar partes do código desenvolvidas por diferentes colegas reforçou a importância de

práticas como comentários explicativos e padronização, aspectos que os estudantes passaram a valorizar mais ao longo do projeto. A exposição pública do jogo também foi destacada pelos alunos como um momento significativo, por permitir que percebessem o impacto real do que haviam construído.

Além dos resultados positivos, algumas limitações foram identificadas. A versão gratuita da plataforma Replit impôs restrições de tempo de execução e número de usuários simultâneos, exigindo estratégias de contorno que, embora formativas, impactaram a fluidez do trabalho colaborativo. O tempo reduzido de oito semanas, compartilhado entre diferentes grupos da turma, também limitou a exploração de recursos mais avançados, como design responsivo e aprimoramento da interface web. Ainda assim, essas limitações tornaram-se oportunidades pedagógicas, uma vez que desafiaram os alunos a propor soluções criativas e colaborativas. Reconhecem-se, ainda, desafios próprios das metodologias ativas, como a avaliação individual em projetos colaborativos, a variação de engajamento entre participantes e a adaptação a contextos curriculares mais rígidos, pontos que serão contemplados nas próximas edições da proposta.

De modo geral, a experiência demonstrou que o desenvolvimento de um jogo digital pode atuar como catalisador de aprendizagens integradas, combinando conteúdos técnicos, competências socioemocionais e protagonismo estudantil. Além disso, evidenciou que a integração entre disciplinas, o desenvolvimento de um produto autêntico e a socialização pública dos resultados contribuíram para ampliar o engajamento dos estudantes ao longo do projeto. O processo reforça a visão de que o ensino de programação pode ser simultaneamente rigoroso e prazeroso quando associado à autoria, à experimentação e à resolução de problemas reais.

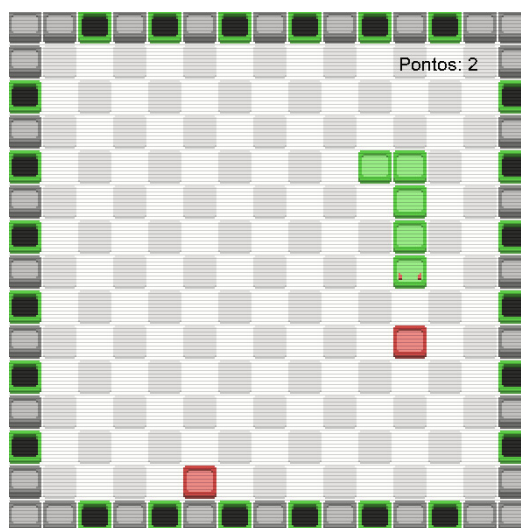


Figura 2- Tela do jogo com a cobrinha em movimento, aproximando-se do item a ser coletado.

5. Conclusão

O desenvolvimento do jogo da cobrinha por estudantes do primeiro ano do Curso Técnico em Informática Integrado ao Ensino Médio evidenciou o potencial de projetos práticos e colaborativos para o ensino de programação e banco de dados. A proposta integrou conteúdos curriculares de forma dinâmica e contextualizada, favorecendo o

protagonismo estudantil, a autonomia e o desenvolvimento do pensamento computacional por meio de metodologias ativas de aprendizagem.

Durante a atividade, os alunos enfrentaram desafios reais relacionados ao ciclo completo de desenvolvimento de software: concepção, tomada de decisões técnicas, prototipagem, testes e aprimoramentos iterativos do produto final. O uso de ferramentas como Replit, Pygame, Flask e SQLite mostrou-se adequado ao contexto educacional, permitindo a aplicação de conceitos teóricos em situações autênticas, mesmo diante das limitações técnicas e do conhecimento prévio reduzido.

A apresentação pública do jogo em um evento acadêmico ampliou os impactos pedagógicos do projeto, possibilitando que os alunos compartilhassem suas criações com a comunidade escolar e recebessem feedbacks significativos. Esse momento reforçou a motivação dos participantes, valorizou o reconhecimento público e proporcionou uma aprendizagem expandida, que extrapolou os limites da sala de aula.

Os resultados obtidos nesta experiência indicam que o desenvolvimento de jogos digitais no ensino técnico pode funcionar como um catalisador de aprendizagens integradas, combinando aspectos técnicos, criativos e socioemocionais. Além disso, evidenciou o potencial das metodologias ativas para estimular a autoria, o pensamento crítico e o engajamento dos estudantes com o conhecimento.

Considerando o sucesso da experiência, o projeto abre caminho para novas edições e desdobramentos pedagógicos. Pretende-se, em versões futuras, ampliar o escopo técnico (incluindo elementos avançados de HTML e CSS, testes automatizados e novas mecânicas de jogo), incorporar temas transversais como sustentabilidade, cidadania e inclusão digital, e aplicar instrumentos de avaliação diretamente aos estudantes desenvolvedores, como questionários de percepção e entrevistas reflexivas, aprofundando a compreensão dos impactos formativos do processo.

Como continuidade institucional, os professores envolvidos planejam estruturar um ciclo de projetos integradores, com desafios progressivos ao longo da formação, e criar um repositório digital aberto para divulgar os jogos desenvolvidos. Essa iniciativa visa consolidar uma cultura de compartilhamento, código aberto e valorização da autoria discente, fortalecendo a educação profissional e tecnológica como espaço de criação, colaboração e inovação.

Cuidados Éticos

Esta pesquisa configura uma atividade normal de ensino, nos termos do art. 1º, §1º, inciso I da Resolução CNS nº 510/2016, que dispensa o registro e avaliação pelo sistema CEP/CONEP pesquisas que envolvam práticas pedagógicas corriqueiras em ambientes de ensino regular. A participação dos estudantes e do público avaliador foi voluntária, com esclarecimento prévio dos objetivos acadêmicos da atividade, garantindo o anonimato e o uso exclusivo dos dados para fins de pesquisa. Não foram coletadas informações pessoais sensíveis, e os participantes podiam se retirar a qualquer momento sem qualquer prejuízo. O questionário aplicado ao público (n=13) foi anônimo e de baixíssimo risco, sem qualquer intervenção que pudesse causar dano físico, psicológico ou social.

Disponibilidade de Artefatos

O código-fonte do jogo desenvolvido e os instrumentos de coleta de dados (questionário Likert) estão disponíveis em: https://github.com/jvfmg-if/combra_do_IFF_2.0.

Declaração sobre uso de Inteligência Artificial

Este trabalho contou com o apoio de ferramentas de Inteligência Artificial generativa exclusivamente para fins de revisão textual e aprimoramento da clareza da redação. As tecnologias utilizadas não participaram da concepção metodológica, da coleta ou análise de dados, nem da interpretação dos resultados científicos apresentados. Todas as decisões acadêmicas, análises e conclusões são de responsabilidade dos autores.

Referências

- Abreu, J. M. de, Castro, H. R. G. de, Reis, C. D. G. e Sousa, P. M. de. (2023) “Poké Prog: Jogo educacional para o ensino dos conceitos básicos de programação”, In: Trilha de Educação – Artigos Completos - Simpósio Brasileiro de Jogos e Entretenimento Digital (SBGames), 22., 2023, Rio Grande/RS, Porto Alegre: Sociedade Brasileira de Computação, p. 881–891.
- Alves, E. J. S., Teixeira, D. C. S., Silva Junior, M. G., Araújo, D. A. S., Nipo, D. T., and Rodrigues, R. L. (2025). "Desenvolvimento de Jogos para Ensino do Pensamento Computacional: Uma Abordagem Baseada em Projetos". RENOTE, Porto Alegre, v. 22, n. 3, p. 334–343. DOI: 10.22456/1679-1916.144998.
- Bacich, L. and Moran, J. (Eds). (2018). Metodologias ativas para uma educação inovadora: uma abordagem teórico-prática. Porto Alegre: Penso.
- Brasil. Ministério da Educação. (2018), Base Nacional Comum Curricular. Brasília, DF: MEC. Disponível em: <https://www.gov.br/mec/pt-br/cne/base-nacional-comum-curricular-bncc>, acesso em 01 maio 2026.
- Brasil. (2021). Resolução CNE/CP n.º 1, de 5 de janeiro de 2021. Define as Diretrizes Curriculares Nacionais Gerais para a Educação Profissional e Tecnológica (EPT). Diário Oficial da União: seção 1, Brasília, DF, p. 77, 6 jan.
- Brackmann, C. P. et al. (2017) “Pensamento computacional: uma proposta de ensino para os anos iniciais do ensino fundamental”, RENOTE – Revista Novas Tecnologias na Educação, v. 15, n. 1, p. 1–10, <https://seer.ufrgs.br/renote/article/view/99894>, acesso em 02 maio 2026.
- Brennan, K. and Resnick, M. (2012). “Using artifact-based interviews to study the development of computational thinking in interactive media design”. Paper presented at the AERA Annual Meeting, Vancouver, BC, Canada.
- Demo, P. (2008). “Habilidades do Século XXI”. Boletim Técnico do Senac, v. 34, n. 2, p. 4–15.
- Genesio, N. O. S., Oliveira, A. M., Oliveira, E. W., and Valle, P. H. D. (2024). "Panorama de Estudos sobre Jogos Educacionais Digitais em Educação em Computação". In: Workshop sobre Educação em Computação (WEI), 32., 2024, Brasília/DF. Porto Alegre: SBC.

- Martinez, S. and Stager, G. (2013), *Invent to Learn: Making, Tinkering, and Engineering in the Classroom*. Constructing Modern Knowledge Press.
- Oliveira, K. K. de S., Marcolino, A. da S., Deus, W. S. de, Falcão, T. P. and Barbosa, E. F. (2023). “Pensamento Computacional na Programação Introdutória e Habilidades do Século XXI: Um Mapeamento Sistemático da Literatura”. *RENOTE*, Porto Alegre, v. 21, n. 2, p. 519–531. DOI: 10.22456/1679-1916.137787. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/137787>. Acesso em: 03 maio 2026.
- Papert, S. (1980), *Mindstorms: Children, Computers, and Powerful Ideas*. Basic Books.
- Petri, A., Schindler, C., Slanly, W., Spieler, B. and Smith, J. (2015). “Pocket Game Jams: a Constructionist Approach at Schools”, In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct*, pages 1207–1211.
- 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*, v. 27, n. 3, p. 52–81. DOI: 10.5753/rbie.2019.27.03.52.
- Pinheiro, F. S., Neves, V. G. da S., Santos, M. B. dos, Mendonça Neto, V. dos S., and Cardoso, C. C. (2025). “Show da Programação: Quiz Educativo para a Aprendizagem de Lógica de Programação e Python”. *RE@D – Revista de Educação a Distância e eLearning*, v. 8, n. 1, e202504. DOI: 10.34627/redvol8iss1e202504.
- Resnick, M. (2017), *Lifelong Kindergarten: Cultivating Creativity through Projects, Passion, Peers, and Play*. Cambridge, MA: The MIT Press.
- Shute, V. J., Sun, C. and Asbell-Clarke, J. (2017). “Demystifying computational thinking”. *Educational Research Review*, v. 22, p. 142–158.
- Silva, K. S., Lima, A. C., Junior, A. A. C. and Odakura, V. V. V. A. (2024) “Desenvolvimento e Aplicação da Metodologia G-Thinking em Escola da Autoria: Promovendo Protagonismo Juvenil através da Cultura Maker”, In: *Proceedings of the 32nd Workshop sobre Educação em Computação (WEI)*, Brasília/DF. Porto Alegre: Sociedade Brasileira de Computação, pp. 172–182.
- Wing, J. M. (2006). “Computational Thinking”. *Communications of the ACM*, v. 49, n. 3, p. 33–35.