

De consumidor a criador: desafios de acesso e permanência no ensino de programação com *Scratch* em contexto de vulnerabilidade social

Túlio Augustus Santos Viana¹, Diego Zabet², Débora Abdalla Santos²

¹Instituto de Humanidades, Artes e Ciências – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

²Instituto de Computação – Universidade Federal da Bahia (UFBA)
Salvador – BA – Brasil

{tulioviana, diego.zabet, abdalla}@ufba.br

Abstract. *This paper presents an experience report of an introductory programming course developed as part of an extension initiative at the Federal University of Bahia, in partnership with the Onda Digital Project and a community center located in a socially vulnerable area in Salvador, Brazil. The course used the Scratch platform to introduce youth to digital authorship and basic programming concepts. Throughout the experience, challenges related to limited access to equipment and participant dropout were observed. The paper discusses strategies for participant mobilization, retention, and pedagogical adaptation in computing education initiatives carried out in socially vulnerable contexts.*

Resumo. *Este artigo apresenta o relato de experiência de um curso de introdução à lógica de programação desenvolvido no âmbito de uma ação extensionista da Universidade Federal da Bahia, em parceria com o Projeto Onda Digital e um centro comunitário em território de vulnerabilidade social em Salvador-BA. O curso utilizou a plataforma Scratch para introduzir jovens à autoria digital e a conceitos básicos de programação. Ao longo da experiência, observaram-se desafios relacionados ao acesso a equipamentos e à evasão de participantes. O trabalho discute estratégias de mobilização, permanência e adaptação pedagógica em iniciativas de ensino de computação em contextos socialmente vulneráveis.*

1. Introdução

O acesso à educação em computação ainda é marcado por desigualdades estruturais, especialmente em contextos de vulnerabilidade social, nos quais a limitação de infraestrutura tecnológica compromete as oportunidades formativas [CGI.br 2023]. Estudos realizados em realidades brasileiras semelhantes indicam que a ausência de computador no ambiente doméstico e o acesso restrito a dispositivos reduzem a continuidade e a profundidade das experiências de aprendizagem em programação, impactando inclusive a permanência dos estudantes em cursos da área [Santos et al. 2021].

Nesse cenário, ações extensionistas universitárias podem atuar como mediadoras entre a produção acadêmica e as demandas da comunidade, ampliando o acesso à programação e à autoria digital, ao mesmo tempo em que consideram as especificidades locais

no planejamento pedagógico [FORPROEX 2012]. Iniciativas desse tipo têm sido apontadas como relevantes para ampliar o acesso à Educação em Computação em territórios com acesso limitado a recursos tecnológicos.

Este artigo apresenta o relato de experiência de um curso de iniciação à programação desenvolvido no âmbito de uma ação extensionista da Universidade Federal da Bahia, em parceria com o Projeto Onda Digital e com o Centro de Educação Livre Vida Plena (CELVIP), localizado em Salvador-BA.

O público atendido pelo CELVIP é composto por jovens residentes em área de vulnerabilidade social, com acesso limitado a recursos tecnológicos e a oportunidades formativas na área da computação. Em contextos como esse, a Educação em Computação tende a ocupar um espaço secundário na trajetória escolar, sendo muitas vezes restrita ao uso instrumental de tecnologias digitais. Tal cenário reforça a condição de consumo de tecnologias, sem necessariamente promover a compreensão de seus mecanismos de funcionamento ou a autoria digital.

Nesse sentido, propor a programação como prática de criação — e não apenas de uso — pode contribuir para ampliar a percepção dos estudantes sobre seu próprio potencial formativo e sobre as possibilidades de intervenção tecnológica em seu contexto social. A utilização do Scratch, em virtude de sua interface em blocos e de sua abordagem visual de programação, configura-se como ferramenta potencialmente adequada para a iniciação à programação, ao facilitar a construção de algoritmos e a experimentação criativa, consideradas habilidades essenciais para a vida no século XXI [Resnick 2007].

Adota-se neste trabalho a noção de linguagem computacional proposta por [Costa et al. 2024], compreendida como um conjunto de habilidades relacionadas à organização lógica, construção de algoritmos e compreensão do funcionamento das tecnologias digitais nas práticas pedagógicas. Essa perspectiva dialoga com formulações clássicas sobre o desenvolvimento de habilidades associadas à resolução de problemas na computação [Wing 2006] e com o framework de conceitos, práticas e perspectivas no Scratch, que enfatiza a autoria, a experimentação e a construção de artefatos digitais como parte do processo formativo [Brennan and Resnick 2012].

O curso foi estruturado considerando as condições locais dos participantes, o que exigiu que o desenvolvimento das atividades ocorresse majoritariamente no laboratório da instituição parceira. Ao longo da oferta, registraram-se desafios relacionados à evasão e à permanência, demandando adaptações metodológicas baseadas em acompanhamento individual e escuta ativa. A partir dessa experiência, o trabalho analisa desafios pedagógicos e organizacionais na oferta de cursos de programação em contextos socialmente vulneráveis, discutindo estratégias de mobilização inicial, permanência em atividades extracurriculares e adaptação metodológica diante de cenários de evasão.

2. Trabalhos relacionados

No contexto brasileiro, diferentes estudos têm investigado o uso do *Scratch* como ferramenta pedagógica para o ensino de programação. [Melo et al. 2022] propõem a utilização de roteiros práticos para introdução de conceitos fundamentais, como variáveis, estruturas condicionais e repetição, evidenciando que a abordagem contribui para tornar o processo de ensino-aprendizagem mais dinâmico e acessível para iniciantes.

De forma complementar, [Aono et al. 2017] apresentam uma experiência com estudantes do ensino fundamental na qual a construção de um jogo digital é utilizada como elemento central do processo de aprendizagem, indicando aumento do engajamento e da motivação dos alunos, além da compreensão dos conceitos de programação.

Além disso, abordagens baseadas em projetos e na autoria digital têm sido apontadas como estratégias relevantes para o desenvolvimento de habilidades computacionais. [Madureira and Schneider 2024] destacam que a criação de jogos no Scratch favorece a articulação entre lógica de programação e expressão criativa, promovendo maior autonomia dos estudantes e ampliando as possibilidades de aprendizagem significativa.

Paralelamente, outras abordagens investigam estratégias voltadas ao engajamento e à continuidade dos estudantes na área. [Santos et al. 2025] analisam programas de treinamento em programação no ensino básico, evidenciando que metodologias estruturadas e baseadas em resolução de problemas podem aumentar o interesse dos alunos pela computação, além de estimular o desenvolvimento de habilidades cognitivas e computacionais.

Em cenários com acesso limitado a recursos tecnológicos, estudos recentes têm explorado a oferta de cursos de programação como estratégia de inclusão e aproximação com a área de Computação. [Mandowski et al. 2024] apresentam uma ação extensionista voltada ao ensino de lógica de programação para estudantes do ensino médio, na qual foram observadas melhorias no desempenho dos participantes. No entanto, os autores também destacam desafios relacionados à permanência dos alunos ao longo do curso, incluindo taxas de evasão, evidenciando a necessidade de estratégias pedagógicas que considerem o engajamento contínuo.

Apesar dos avanços observados, ainda são limitados os estudos que analisam, de forma integrada, o ensino de programação com *Scratch* considerando aspectos relacionados à continuidade e envolvimento dos participantes ao longo do processo formativo. Nesse sentido, o presente trabalho contribui ao apresentar e analisar uma experiência que evidencia estratégias pedagógicas adotadas diante de desafios de continuidade e participação.

3. Metodologia

Trata-se de um estudo qualitativo exploratório, fundamentado na análise pedagógica de uma experiência de ensino de programação em contexto comunitário com restrições de acesso a recursos digitais.

3.1. Contexto institucional e público participante

O curso foi ofertado presencialmente no laboratório de informática do CELVIP, localizado no bairro de Pau da Lima, em Salvador-BA, no período de agosto a dezembro de 2025, totalizando 17 encontros semanais com duração de três horas cada. A divulgação do curso ocorreu por meio da coordenação do centro comunitário, com convite direto aos jovens participantes das atividades do espaço. O público inicial foi composto por três jovens do sexo masculino, com idades entre 11 e 15 anos, residentes na comunidade local. Observou-se heterogeneidade quanto ao nível de familiaridade com computadores, sendo que apenas um dos participantes possuía computador em casa. Essa limitação de acesso

impactou diretamente o ritmo de desenvolvimento das atividades e exigiu planejamento voltado à realização das tarefas predominantemente durante os encontros presenciais.

3.2. Estrutura pedagógica do curso

O curso foi estruturado com foco na introdução à lógica de programação por meio da criação de projetos no *Scratch*. A proposta articulou momentos expositivos breves com atividades práticas orientadas, priorizando a construção progressiva de pequenos artefatos digitais. O primeiro encontro foi dedicado ao diagnóstico inicial, por meio de atividades individuais e em duplas, incluindo a “Dinâmica do Robô”, na qual um participante assumia o papel de programador e outro de executor das instruções.

A atividade foi realizada após uma explicação introdutória sobre lógica de programação, sistema binário e a importância da precisão na comunicação com computadores. Foi discutido com os alunos que máquinas executam comandos exatamente como são instruídas, sem interpretar intenções ou corrigir ambiguidades, destacando a necessidade de instruções claras, organizadas e sequenciais. Na atividade prática, os alunos foram organizados em duplas rotativas: o estudante E1 trabalhou com o estudante E2, o estudante E2 com o estudante E3 e, posteriormente, o estudante E3 com o estudante E1. Cada estudante assumiu alternadamente o papel de programador e de “robô humano”. Inicialmente, os alunos desenharam em seus cadernos percursos representando caminhos dentro da sala, utilizando linhas, curvas e pontos de chegada, inserido também instruções detalhadas para que conseguissem dar comandos mais claros para o trajeto, como “andar 5 passos”, “virar à esquerda”, “virar à direita” e “andar para trás”. O colega que assumia o papel de robô deveria seguir exatamente os comandos escritos, sem interpretações adicionais. Durante a execução, os alunos perceberam na prática a importância da precisão, da sequência lógica e da clareza das instruções, especialmente quando surgiam erros de percurso causados por comandos incompletos ou ambíguos, ou até mesmo quando eles tomavam decisões sozinhos.

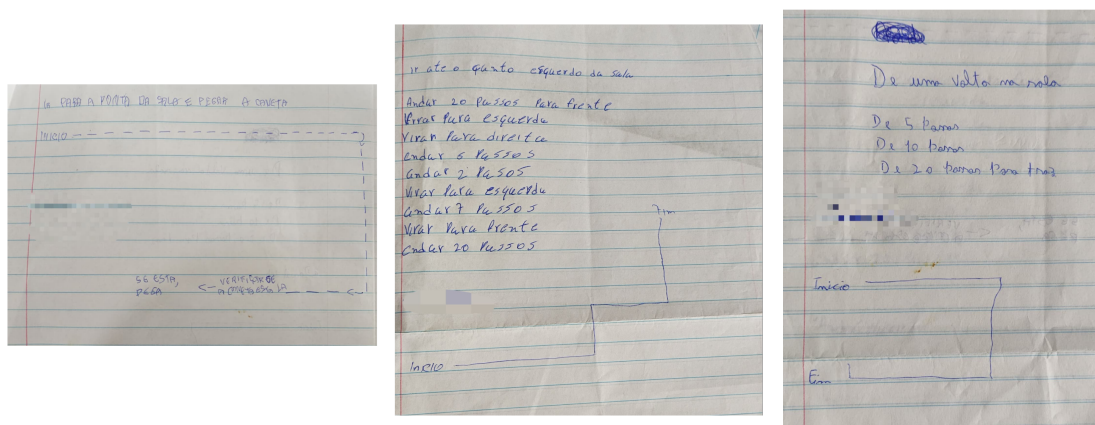


Figura 1. Registros das anotações produzidas pelos estudantes durante a dinâmica do robô, evidenciando diferentes formas de representação dos comandos por meio de desenhos e textos.

Como primeira atividade avaliativa, os estudantes produziram um *teaser* no *Scratch*: uma animação curta e não interativa que apresentava a temática do jogo final.

A atividade teve como objetivo familiarizar os participantes com o ambiente e introduzir conceitos básicos de sequenciamento lógico e planejamento de algoritmos.

Ao longo do curso, os conteúdos foram organizados em três eixos principais:

- fundamentos de lógica de programação;
- exploração dos blocos e estruturas do *Scratch*;
- desenvolvimento de um projeto autoral (jogo final).

A Tabela 1 apresenta a organização do curso.

Tabela 1. Estrutura do curso e progressão dos conteúdos

Período / Encontros	Objetivo principal	Conteúdos abordados	Atividades principais
Encontros 1–4	Nivelamento, introdução à lógica e ao Scratch	Sequência, precisão de comandos	Dinâmica do Robô; exercícios iniciais; teaser
Encontros 5–7	Introdução ao Scratch	Eventos, movimentos, blocos básicos	Mini-projetos orientados; mini desafios
Encontros 8–10	Criação do design do jogo	Design, narrativa	Construção de narrativa; elaboração de personagens e layout; tarefas para casa
Encontros 11–12	Estruturas lógicas	Condicionais, laços, variáveis	Desafios progressivos
Encontros 13–17	Projeto final	Integração de conceitos	Desenvolvimento do jogo

As ferramentas de apoio incluíram o *Moodle* (organização de materiais e registros), *Dontpad* (compartilhamento de links e textos de forma rápida entre envolvidos), *Padlet* (dinâmica inicial de expectativas), e grupo de *WhatsApp* (comunicação e incentivo à permanência).

3.3. Estratégias de acompanhamento e adaptação

Durante as primeiras semanas, registrou-se evasão parcial da turma, reduzindo o grupo a um participante concluinte. Diante desse cenário, a metodologia foi adaptada para um formato de acompanhamento individualizado. As principais estratégias adotadas foram:

- divisão das aulas em blocos curtos de exposição e prática;
- desafios progressivos com metas claras por encontro;
- apoio constante durante a implementação dos projetos;
- flexibilização do planejamento inicial;
- incentivo ao uso do laboratório em horários alternativos.

O momento inicial de cada aula passou a incluir espaço para diálogo sobre dificuldades e expectativas, buscando fortalecer o vínculo e favorecer a permanência. Buscou-se incentivar o estudante a avançar progressivamente em etapas específicas do projeto, iniciando pela construção do cenário, seguida pela criação dos *sprites* e, posteriormente, pela elaboração do roteiro e da lógica de funcionamento das fases.

Durante esse processo, o professor atuava como mediador, propondo questionamentos que estimulavam o pensamento crítico, a organização lógica e a tomada de decisões no desenvolvimento do jogo, tais como:

- como será a coleta de pontuação?
- como ocorrerá a perda de pontos?
- de que forma o jogador avançará de fase: por tempo, pontuação, superação de desafios ou outros critérios?

Essas reflexões eram desenvolvidas pelo estudante também fora da sala de aula e retomadas nos encontros presenciais, nos quais eram discutidas as possibilidades elaboradas e suas formas de implementação. Quando surgiam dificuldades técnicas, a mediação ocorria por meio de perguntas orientadoras e problematizações, evitando respostas prontas e promovendo a análise do próprio código e o desenvolvimento de autonomia na resolução dos problemas encontrados.

Embora o contexto tenha se configurado como um acompanhamento individual, as estratégias adotadas evidenciam possibilidades de adaptação pedagógica em cenários de baixa participação, podendo ser consideradas em ações similares de ensino de programação.

3.4. Produção do artefato final

Na segunda metade do curso (a partir do encontro 8), a metodologia adotada concentrou-se na aprendizagem baseada em projetos, propondo como desafio final o desenvolvimento de um jogo digital autoral utilizando a plataforma *Scratch*. Para garantir a aplicação e a avaliação dos conceitos da linguagem computacional abordados nas aulas teóricas, foram estabelecidos requisitos técnicos obrigatórios para a concepção do artefato. O projeto final deveria complementar: (i) a estruturação do jogo em múltiplas fases, exigindo lógicas de transição de cenários; (ii) a implementação de um sistema de variáveis para o gerenciamento de dados da partida, como pontuação e controle de status; (iii) o uso de estruturas de controle de fluxo, com ênfase em lógicas condicionais; e (iv) a integração de múltiplos *sprites* (atores). Além dos aspectos de programação, a proposta metodológica previu uma etapa de socialização como critério de conclusão, na qual o estudante deveria realizar uma apresentação pública de seus projetos, visando justificar as escolhas lógicas e exercitar habilidades de comunicação.

4. Resultados e Discussão

A análise da experiência permitiu identificar diferentes aspectos relacionados à implementação do curso, especialmente no que se refere à mobilização e continuidade dos participantes, às adaptações pedagógicas diante da redução da turma e às evidências de desenvolvimento da linguagem computacional observadas no artefato final.

4.1. Perfil da turma e permanência

A turma iniciou com três participantes do sexo masculino (E1, E2, E3), com idades entre 11 e 15 anos. Ao longo das primeiras semanas observou-se redução progressiva da participação, resultando em um único estudante concluinte. A frequência registrada nos 17 encontros foi: E1, 16/17; E2, 3/17; E3, 4/17. A evasão de E2 e E3 ocorreu antes da etapa final de desenvolvimento do projeto autoral. E3 interrompeu a participação durante a fase de elaboração do teaser por motivos pessoais comunicados à coordenação. E2 deixou de frequentar as aulas sem justificativa formal, não sendo possível contato posterior.

Identificou-se heterogeneidade quanto ao acesso a equipamentos. Apenas E2 possuía computador em casa, enquanto E1 e E3 dependiam exclusivamente do laboratório do centro comunitário para a realização das atividades. Essa diferença implicava variação no tempo de prática extraclasses e pode ter influenciado tanto o ritmo de aprendizagem quanto a permanência no curso.

Com a redução da turma, o curso foi reestruturado para acompanhamento individualizado do estudante remanescente (E1). Como este não dispunha de computador em casa, o tempo presencial passou a concentrar integralmente o desenvolvimento das atividades, exigindo planejamento voltado à execução completa das tarefas durante os encontros.

4.2. Evidências de aprendizagem no artefato final

O estudante concluinte desenvolveu um jogo autoral no *Scratch* estruturado em quatro fases, do tipo plataforma, envolvendo movimentação lateral do personagem, presença de inimigos e coleta de objetos. O jogo inclui sistema de pontuação e interações baseadas em condições lógicas.

O artefato final foi composto por 23 *sprites* (personagens e objetos), 11 variáveis de controle e 5 listas, utilizadas para organizar estados do jogo e gerenciar comportamentos específicos. Foram ainda implementadas 5 mensagens de comunicação, evidenciando a compreensão da lógica de sincronização entre eventos e da coordenação entre diferentes *scripts*. A Figura 2 apresenta capturas de tela do artefato desenvolvido. As imagens ilustram a integração entre interface, controle de estado e progressão lógica implementada no projeto.



Figura 2. Capturas de tela do artefato desenvolvido, evidenciando diferentes estados do jogo: (a) a tela inicial de inserção do nome do jogador, (b) o primeiro nível com mecânica ativa e sistema de pontuação visível, e (c) a tela de encerramento da partida.

A análise do projeto permitiu identificar a incorporação de estruturas fundamentais da programação em blocos, como eventos, sequências, condicionais, laços, variáveis e operadores lógicos. A complexidade estrutural do artefato indica progressão na organização do código e maior autonomia na implementação das funcionalidades propostas.

Observou-se que, ao longo do desenvolvimento do projeto, o estudante passou a integrar de forma autônoma recursos multimídia, como áudios e imagens, além de aplicar estruturas de controle de fluxo, especialmente condicionais simples, sem necessidade de intervenção constante do educador.

Uma análise comparativa entre as versões iniciais e a versão final indica progressão na organização lógica do código e na personalização da ambientação visual. Enquanto os primeiros protótipos apresentavam inconsistências de execução e uso restrito de elementos gráficos, a versão final incorporou personagens originais e variações visuais, incluindo uma versão alternativa do avatar principal.

Esse processo sugere ampliação da autonomia técnica e da capacidade de articulação entre lógica computacional e expressão criativa no contexto da autoria digital.

A Tabela 2 sistematiza as principais evidências observadas no artefato final.

Tabela 2. Evidências de aprendizagem no artefato final

Conceito de Programação	Implementação observada no jogo	Função no jogo
Eventos	“quando bandeira verde clicada”; “quando tecla for pressionada”; “quando o cenário mudar para...”; “quando eu receber...”; “transmita”	Inicialização do jogo, controle de entrada do usuário e sincronização entre fases
Sequência	Organização encadeada de blocos	Execução ordenada das ações dos personagens
Condicionais	“se... então...” para colisões e controle de estados	Definição de regras, detecção de eventos e tomada de decisão
Laços	“repita até...”; “para sempre”	Movimentação contínua de personagens e repetição de padrões de ataque
Variáveis	Sistema de pontuação; sistema de dano	Controle de estado e monitoramento do desempenho
Operadores	Comparações numéricas (>, <, =)	Implementação de regras lógicas e critérios de progressão

Resultados semelhantes têm sido observados em pesquisas que associam a autoria de jogos no *Scratch* ao desenvolvimento de habilidades relacionadas à organização lógica e à construção de algoritmos, especialmente quando o processo privilegia a produção de artefatos autorais [Madureira and Schneider 2024].

4.3. Desenvolvimento da linguagem computacional

Considerando a noção de linguagem computacional adotada neste trabalho [Costa et al. 2024], observou-se avanço na capacidade do estudante de organizar sequências lógicas, estruturar algoritmos em blocos e depurar erros durante a execução do programa. Tais aspectos dialogam com habilidades associadas ao pensamento computacional, entendido como a capacidade de formular problemas e expressar soluções passíveis de execução por um agente computacional [Wing 2006].

No contexto do *Scratch*, essas habilidades manifestaram-se na decomposição do jogo em fases distintas, na definição de regras condicionais para interações entre perso-

nagens e na utilização de variáveis para controle de pontuação e estados do sistema. A comparação entre os protótipos iniciais e a versão final indica progressão na organização do raciocínio e maior autonomia na antecipação de comportamentos do jogo.

A “Dinâmica do Robô”, realizada no início do curso, contribuiu para a compreensão da necessidade de instruções precisas e sequenciais. Ao final do processo, o estudante demonstrava maior clareza na formulação de comandos e na identificação de inconsistências lógicas.

Como parte da atividade de encerramento, foi solicitado ao participante um relato escrito reflexivo sobre sua trajetória no curso. O estudante destacou a **aprendizagem dos blocos de programação**, a **criação do primeiro jogo autoral** e a **percepção de ampliação** de sua capacidade de desenvolver soluções consideradas anteriormente difíceis. Esse registro qualitativo sugere reconhecimento do próprio processo de aprendizagem e maior autoconfiança na produção tecnológica.

4.4. Lições aprendidas

A análise da experiência permitiu identificar três dimensões centrais para a implementação de cursos de programação em contextos de vulnerabilidade social: (i) estratégias de mobilização inicial dos participantes; (ii) mecanismos de permanência em atividades formativas extracurriculares; e (iii) adaptações metodológicas diante de cenários de evasão. A partir dessas dimensões, destacam-se os seguintes aspectos observados ao longo da experiência:

- a permanência exige estratégias específicas em contextos de vulnerabilidade;
- a ausência de computador em casa limita o tempo de prática e exige planejamento para conclusão das atividades em sala;
- metas intermediárias (como *teaser*, desafios progressivos e jogo final) favorecem engajamento;
- o acompanhamento individual pode potencializar a autonomia quando a turma é reduzida;
- o *Scratch* mostrou-se adequado para introdução à programação por permitir experimentação sem barreiras sintáticas.

Este relato apresenta como limitação o número reduzido de participantes e a ausência de instrumentos formais de avaliação padronizada, caracterizando-se como estudo exploratório. Ainda assim, experiências desse tipo podem oferecer evidências qualitativas relevantes sobre os desafios e possibilidades de iniciativas de Educação em Computação em territórios com acesso limitado a recursos tecnológicos.

5. Considerações Finais

A ação extensionista, desenvolvida pela UFBA em parceria com o centro comunitário local, não apenas promoveu acesso à programação, mas também aproximou o estudante da universidade como instituição pública, ampliando a percepção de pertencimento e de possibilidade de inserção futura no ensino superior.

Mais do que os resultados de aprendizagem observados, a experiência evidencia a importância de compreender as condições de acesso e permanência em iniciativas de

ensino de computação voltadas a públicos em contextos com restrições de infraestrutura e recursos tecnológicos.

A mobilização inicial revelou-se um desafio significativo. A evasão observada durante a experiência reforça que iniciativas de ensino de programação em territórios socialmente vulneráveis precisam considerar estratégias específicas de mobilização comunitária e permanência dos participantes. A introdução à programação em contexto de vulnerabilidade social demanda não apenas planejamento técnico, mas um desenho pedagógico orientado à permanência e ao diálogo com a realidade dos estudantes. Nessa perspectiva, a mediação adotada aproximou-se da concepção freireana de educação como prática dialógica, na qual o processo formativo se constrói na interação e na escuta ativa [Freire 2011]. Estratégias como metas intermediárias (*teaser* e jogo final), acompanhamento individualizado e abertura para diálogo mostraram-se relevantes para sustentar o engajamento do estudante concluinte. Experiências semelhantes relatadas na literatura indicam que a combinação entre atividades práticas, ludicidade e projeto final autoral contribui para o aumento da confiança e da autonomia dos participantes [Ferreira et al. 2016].

Além disso, a composição exclusivamente masculina da turma reforça evidências apontadas na literatura sobre a persistência de estereótipos que associam a computação ao universo masculino, influenciando o interesse e a autoconfiança de meninas desde a educação básica [Maciel et al. 2021]. Esse dado reforça a necessidade de ações afirmativas e estratégias de sensibilização específicas para ampliar o acesso de meninas à computação em territórios vulneráveis.

Como próximos passos, pretende-se manter a parceria com o centro comunitário e ampliar as estratégias de mobilização local, incentivando a participação de estudantes de diferentes idades e gêneros. Busca-se também estruturar mecanismos mais sistemáticos de acompanhamento da evasão e ampliar o acesso a equipamentos, fortalecendo iniciativas de ensino de computação sensíveis ao contexto social dos participantes. Espera-se, assim, consolidar um modelo de curso capaz de promover tanto o desenvolvimento da linguagem computacional quanto a percepção de pertencimento à área da computação.

Agradecimentos

Os autores agradecem ao Centro Comunitário Vida Plena (CELVIP) pelo apoio financeiro, por proporcionarem o espaço e a infraestrutura necessária para a realização deste projeto. Expressamos também nossos sinceros agradecimentos à coordenadora Ednolia Peixinho por todo suporte, liderança e incentivo contínuo ao longo do processo. Por fim, agradecemos à Universidade Federal da Bahia (UFBA) e ao projeto de pesquisa e extensão Onda Digital pelo fundamental apoio pedagógico e de extensão acadêmica que norteou as práticas realizadas neste trabalho.

Declaração sobre uso de Inteligência Artificial

Declaramos que a ferramenta de Inteligência Artificial Generativa Google Gemini foi utilizada como assistente de revisão gramatical, adequação de estilo acadêmico, tradução do resumo e auxílio na estruturação textual. O conteúdo original, os dados da pesquisa, as reflexões pedagógicas e a autoria intelectual pertencem integralmente aos autores, que assumem total responsabilidade por todo o conteúdo do trabalho.

Referências

- Aono, A. H., Rody, H. V. S., Musa, D. L., Pereira, V. A., and Almeida, J. (2017). A utilização do scratch como ferramenta no ensino de pensamento computacional para crianças. In *Workshop sobre Educação em Computação (WEI)*, pages 2169–2178. SBC.
- Brennan, K. and Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American educational research association, Vancouver, Canada*, volume 1, page 25.
- CGI.br (2023). Comitê Gestor da Internet no Brasil. Pesquisa sobre o uso das tecnologias de informação e comunicação nos domicílios brasileiros: TIC Domicílios 2023. Technical report, CGI.br, São Paulo. Disponível em: <https://cetic.br/pt/pesquisa/domicilios/>. Acesso em: 26 fev. 2026.
- Costa, I. T. d., Nunes, A. P. R., Sousa, A. G. d., Malta, D. P. d. L. N., Trindade, K. C., and Barbosa, R. A. (2024). Narrativas docentes sobre a apropriação da linguagem computacional e o uso da tecnologia como instrumento técnico-semiótico nas práticas pedagógicas. *Revista Científica de Alto Impacto*, 28(136). Disponível em: <https://revistaft.com.br/narrativas-docentes-sobre-a-apropriacao-da-linguagem-computacional-e-o-uso-da-tecnologia-como-instrumento-tecnico-semiotico-nas-praticas-pedagogicas/>. Acesso em: 26 fev. 2026.
- Ferreira, A. C., Santos, J., Silva, R., Oliveira, A. T. R., Zobot, D., Abdalla, D., and Matos, E. (2016). Hello world: relato de experiência de um curso de iniciação à programação. In *Anais dos Workshops do Congresso Brasileiro de Informática na Educação*, volume 5, page 1306.
- FORPROEX (2012). Fórum de Pró-Reitores de Extensão das Universidades Públicas Brasileira. Política Nacional de Extensão Universitária. Technical report, FORPROEX, Manaus. Disponível em: <https://www.ufmg.br/proex/renex/images/documentos/2012-07-13-Politica-Nacional-de-Extensao.pdf>. Acesso em: 26 fev. 2026.
- Freire, P. (2011). *Pedagogia do oprimido*. 50. ed. São paulo: paz e terra.
- Maciel, C., Bim, S. A., and Ribeiro, K. d. S. F. (2021). Meninas digitais: Uma jornada de ciclos enriquecedores. *Computação Brasil*, (44):9–13.
- Madureira, J. S. and Schneider, H. N. (2024). Desenvolvendo o pensamento computacional por meio da estratégia de autoria de jogos associada ao ambiente de programação scratch. *EaD & Tecnologias Digitais na Educação*, 12(16):78–89.
- Mandowski, V. d. M., Rodrigues, W. d. E., Vieira, E. d. V., Konradt, A. C., Martinez, S. D., Mackedanz, T. D., Jurgina, L. Q., and Rosa Júnior, L. S. d. (2024). Curso introdutório de lógica de programação: Uma ação para aproximar jovens em vulnerabilidade social ao aprendizado de computação. In *Workshop sobre Educação em Computação (WEI)*, pages 139–149. SBC.
- Melo, N. R. d., Guimarães, R. A., and Vieira, N. J. (2022). Aplicação do scratch no ensino de programação para iniciantes. *Revista Educação em Debate*.
- Resnick, M. (2007). All i really need to know (about creative thinking) i learned (by studying how children learn) in kindergarten. In *Proceedings of the 6th ACM SIGCHI conference on Creativity & cognition*, pages 1–6.

- Santos, A. S., Almeida, A. M. P., Bezerra, M. A., Bezerra, Â. C. S., da Silva, P. E., and Maia, D. L. (2021). Literacia e inclusão digital: Um estudo piloto com jovens em vulnerabilidade social na cidade de natal-rn. *Research, Society and Development*, 10(6):e27310615859–e27310615859.
- Santos, C. d. C., Santos, S. S. d., Irion, C., de Menezes, G. R., Araújo, R. D., and Pereira, J. H. d. S. (2025). Impacto de treinamento em programação competitiva no ensino médio: Resultados e desafios. *arXiv:2503.04732*.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.