

Um curso de programação para calouros com *Scratch*: um relato de licenciandos em computação

Flávia Brenda¹, Cássia Venâncio¹, Cristiana Pedrosa¹, Jean Miguel¹, Waldecir Martins¹, Fabrizio Honda¹, Jeniffer Macena¹, Fernanda Pires¹, Marcela Pessoa¹

¹Escola Superior de Tecnologia – Universidade do Estado do Amazonas (EST/UEA)
ThinkTED Lab - Pesquisa, Desenvolvimento e Inovação em tecnologias emergentes

{fbrdl.lic21, cjvb.lic20, cdnp.lic23, jmcb.lic22, wdsml.lic22, mspessoa, fpires}@uea.edu.br, {fabrizio.honda, jeniffer.souza}@icompu.ufam.edu.br

Abstract. *Considering the role of the professional in Computer Science Education and the challenges found in higher education computing courses, such as high failure and dropout rates, this paper presents the perspective of computing students, through an experience report, on the application of a programming course using the Scratch tool for students from three computing courses (Computer Science Education, Information Systems, and Computer Engineering) at the State University of Amazonas (UEA). Results indicate that the experience allowed participants to glimpse the challenges of computer science graduates, whether as teachers or learning designers.*

Resumo. *Considerando o papel do profissional de Licenciatura em Computação e os desafios encontrados nos cursos superiores de computação, como altas taxas de reprovação e evasão, este trabalho apresenta o ponto de vista de licenciandos em computação, através de um relato de experiência, em relação à aplicação de um curso de programação utilizando a ferramenta Scratch para discentes de três cursos de computação (Licenciatura em Computação, Sistemas de Informação e Engenharia da Computação) da Universidade do Estado do Amazonas (UEA). Resultados indicam que a experiência permitiu aos participantes vislumbrarem os desafios dos licenciados em computação, seja como professores ou designers de aprendizagem.*

1. Introdução

A formação de professores de computação é um dos desafios da educação brasileira, especialmente em um contexto acelerado de avanço tecnológico e exponencial crescimento das Tecnologias da Informação e Comunicação (TICs) [Oliveira and Cambraia 2020]. Espera-se que o profissional licenciado em Computação assuma o papel de mediador entre as tecnologias e o processo de aprendizagem [Freire et al. 2016], trazendo como formação o estímulo da autonomia dos estudantes na resolução de problemas, indo além de instruí-los no uso de ferramentas tecnológicas [Linhares and Santos 2021]. Dessa forma, é papel da Licenciatura em Computação garantir que os egressos possuam uma sólida formação em Ciência da Computação, Matemática e Educação [Brasil 2016], dentre outras requisições, para que se tornem profissionais capacitados para disseminar os conceitos de computação nas escolas. Isso é fundamental, como preconizam as Diretrizes para

Ensino de Computação na Educação Básica [Ribeiro et al. 2019], para o desenvolvimento de competências e habilidades como o Pensamento Computacional [Wing 2006].

Estudos realizados pelo Brasscom [2021] e pelo *Google for Startups* [2023] apontam um déficit de profissionais de TIC até o ano de 2025: entre 2021 e 2025, 53 mil profissionais irão se graduar anualmente, enquanto a demanda projetada é de 800 mil novos talentos, resultando em um *déficit* de 530 mil profissionais nesse período. O que corrobora com esses dados são os índices de evasão e reprovação em cursos superiores de computação; por exemplo, em Tecnologia da Informação (TI) no Brasil, as taxas de evasão superam a média geral com 38,5% em cursos presenciais e 38% em cursos de Educação a Distância (EAD), comparadas a 30,7% e 36,5% em outras áreas [Semesp 2023]. Tais taxas revelam que a quantidade de estudantes que completam suas formações está bem aquém da necessidade do mercado.

Um dos grupos de disciplinas com maior índice de reprovação é composto por disciplinas de programação [Junior et al. 2021, Viana et al. 2019], tais dificuldades podem estar relacionadas à falta de utilização de elementos lúdicos e práticos, ao uso de uma linguagem de difícil aprendizado e à falta de aulas com teor mais práticas. Mais pontualmente, a dificuldade de absorção de conceitos como definição e uso de variáveis, estruturas condicionais e estruturas de repetição é apontada como fator que dificulta a aprendizagem [Silva et al. 2018].

Por outro lado, o Scratch, uma ferramenta desenvolvida pelo Massachusetts Institute of Technology (MIT), tem se mostrado uma estratégia alternativa no processo de aprendizagem de programação, caracterizando-se como uma linguagem de programação que utiliza blocos para construir programas, atrelados a elementos visuais [Ferreira et al. 2021]. Através de sua interface intuitiva e sua mecânica de arrastar blocos, a ferramenta pode ser uma alternativa para auxiliar na compreensão de conteúdos de programação, tido como complexos pelos estudantes. Assim, este trabalho apresenta um relato de experiência a partir da produção e aplicação de um curso introdutório de programação com a ferramenta Scratch, realizado na Universidade do Estado do Amazonas (UEA), visando promover uma abordagem de aprendizagem para a lógica de programação, destinado aos cursos de computação da instituição. O artigo está estruturado da seguinte forma: Na Seção 2 são discutidos conceitos e os trabalhos relacionados, na Seção 3 é descrita a metodologia, na Seção 4 são apresentados os resultados e na 5 as considerações finais.

2. Referencial Teórico e Trabalhos Relacionados

Pesquisadores têm utilizado estratégias para contornar as altas taxas de evasão e reprovação nos cursos superiores de computação, dentre elas, o uso linguagens visuais como o Scratch – uma linguagem de programação em blocos para transformar a codificação em uma experiência dinâmica e interativa para os usuários [Resnick et al. 2009]. A ferramenta inicialmente foi projetada para um público mais jovem, entretanto sua eficácia no aprendizado de conceitos iniciais de programação permite a utilização por pessoas de todas as idades. Por ser uma linguagem baseada na conexão de blocos, ajuda no entendimento da lógica, minimizando erros de sintaxe e oferecendo uma abordagem eficaz ao aprendizado da programação através da criação de jogos [Marji 2014]. Uma das características mais atrativas do Scratch é a sua abordagem

visual e intuitiva: ao invés de digitar linhas de código, os usuários montam programas conectando blocos de comandos. Essa interface simplificada auxilia estudantes iniciantes que enfrentam dificuldades quando começam a estudar programação, possibilitando-os focar na lógica e estrutura do código.

Nesse intuito, observa-se uma diversidade de trabalhos na literatura que utilizam a linguagem de programação Scratch. Por exemplo, o estudo de Monteiro e Holanda [2023] descreve a aplicação de cursos de Scratch em escolas públicas de Ensino Médio no Distrito Federal. Os resultados evidenciaram o aumento do nível de conhecimento de informática dos estudantes, boa recepção do curso, engajamento dos participantes e que o uso do Scratch para exercitar conteúdos introdutórios de programação mostrou-se eficaz, pois os estudantes puderam focar na lógica de programação e na resolução de problemas, diminuindo, principalmente, os erros de sintaxe.

De modo similar, Jesus et al. [2019], os autores descrevem a realização de oficinas de programação introdutórias com estudantes iniciantes do curso de Engenharia de Computação. A metodologia utilizada envolveu uma abordagem lúdica por meio da realização de desafios e construção de jogos em Scratch e Python. Durante as oficinas, os participantes realizavam atividades práticas que visavam facilitar o processo de aprendizagem, incluindo a transição entre linguagens de programação. Os resultados das aplicações apontam para a alta participação dos estudantes e também para a simplificação do processo de aprendizagem através do Scratch.

Em Cárdenas-Cobo et al. [2021], os autores descreveram um quase-experimento realizado com 74 estudantes de graduação durante o primeiro semestre da Universidade Estadual de Milagro (UNEMI). A metodologia aplicada focou em dividir os estudantes em dois grupos: i) 38 estudantes, que teriam aulas de lógica de programação em Python e; ii) 36 estudantes, que teriam aulas semelhantes, porém com o Scratch. Como resultados, constatou-se que o grupo de estudantes que utilizou o Scratch obteve níveis maiores de entendimento dos conteúdos ministrados, avaliados por meio de testes cognitivos.

Da mesma maneira que nos trabalhos antecedentes, este relata a experiência da criação e aplicação de um curso de programação que utiliza a linguagem Scratch para auxiliar estudantes de ensino superior. Entretanto, ao contrário das outras pesquisas, este tem o foco na experiência de licenciandos em computação, destacando os desafios e as contribuições que a criação e aplicação do curso tiveram na experiência profissional de cada facilitador.

3. Metodologia

Esta seção trata das etapas metodológicas para o registro da experiência vivenciada por licenciandos em computação, descrevendo os objetivos da experiência e do curso, a definição do público-alvo, instrumentos e procedimentos, incluindo o planejamento das aulas, a aplicação do curso e a análise dos dados.

3.1. Objetivo da experiência

O objetivo da experiência foi oferecer aos bolsistas discentes de Licenciatura em Computação a possibilidade de aplicar seus conhecimentos teóricos em um contexto prático através da aplicação de um curso de programação, analisando os desafios que os facilitadores enfrentariam ao lecionar para estudantes de Ensino Superior e as lições

aprendidas ao atuar como instrutores em sala de aula. Para isso, houve necessidade da elaboração do curso, que teria como principal foco a lógica inicial de programação utilizando a linguagem Scratch. Além disso, analisar o impacto que o curso causaria no desenvolvimento nas habilidades de resolução de problemas nos estudantes, observando as dificuldades enfrentadas ao aprender a lógica inicial de programação.

3.2. Objetivo do curso

Uma das etapas da experiência foi a definição do objetivo do curso, para isso, levou-se em consideração as dificuldades relacionadas à aprendizagem de conceitos introdutórios de programação, especialmente aqueles elencados na literatura como conteúdos que os estudantes enfrentam maior dificuldade [Morais et al. 2020]. Dessa forma, o curso de Introdução à Programação com Scratch teve como objetivo auxiliar estudantes de Ensino Superior que apresentam dificuldades na aprendizagem em conteúdos de lógica inicial de programação. Para isto, considerando a abordagem prática e lúdica que o Scratch proporciona, o curso foi estruturado com foco em apresentar conteúdos iniciais de lógica de programação, através da criação de projetos como jogos e animações, para motivar/engajar os estudantes.

3.3. Definição do público-alvo do curso

Para atingir o objetivo do curso, o público-alvo definido foram estudantes iniciantes de cursos de computação (Licenciatura em Computação, Sistemas de Informação e Engenharia de Computação) da Universidade do Estado do Amazonas - Escola Superior de Tecnologia (UEA/EST).

A seleção dos participantes se deu de forma voluntária, por meio de chamada nas redes sociais e grupos de mensagens eletrônicas. Os estudantes interessados responderam um formulário de inscrição, com questões relacionadas a experiências anteriores com programação, período letivo do estudante e dados pessoais. Inicialmente foram disponibilizadas 30 vagas para o curso. Ao todo, houve 27 inscrições, sendo que 15 estudantes atendiam ao critério de ser discente iniciante no curso. Como o número de inscritos não ultrapassou o limite de vagas, todos os estudantes interessados puderam participar do curso. Entretanto, apenas 7 estudantes considerados iniciantes participaram das aulas. Para o contexto desta pesquisa, considerou-se iniciante todos os estudantes que se encontravam até o segundo período do curso. Como o curso aconteceu em um semestre letivo par, os discentes ingressantes estavam cursando o segundo período, com isso, todos já haviam cursado a disciplina de Introdução à Programação de Computadores.

3.4. Instrumentos e Procedimentos

Esta subseção apresenta os instrumentos utilizados e os procedimentos executados para a planejamento, aplicação e análise dos dados do curso de Scratch.

3.4.1. Planejamento e organização do curso

Para o planejamento e organização das aulas do curso, realizou-se as seguintes etapas:

- **Reunião entre os bolsistas:** Um momento de *brainstorm* (tempestade de ideias), junto a um dos coorientadores do projeto, em que se definiu quais aulas seriam contempladas no curso e sua sequência de aplicação, disponíveis na Tabela 1;

Aula	Tema	Objetivo
1	Apresentação e pré-teste	Introduzir o curso e os monitores, além da realização de um diagnóstico para mensurar o nível de conhecimento dos estudantes
2	<i>Game Design</i>	Possibilitar uma introdução ao universo de jogos, através de um momento de explicação de conceitos fundamentais e prototipação de jogo
3 a 10	Scratch	Introduzir a linguagem de programação Scratch, abordando seus blocos fundamentais (Evento, Movimento, Aparência, Som, Sensores, Variáveis, Operadores, Controle, Listas e Meus Blocos) e a extensão de caneta, com exemplos e exercícios respectivos
11	Maratona	Propor um momento para os estudantes conceberem um jogo, a partir de uma lista de requisitos e um tempo delimitado
12	Vetorização de Imagem	Explicar e demonstrar a realização de ilustração vetorial, utilizando o <i>Adobe Illustrator</i>
13	Tratamento de Imagem	Explicar e demonstrar edição de imagem através do <i>Adobe Photoshop</i>
14 e 15	Robótica	Introduzir e demonstrar conceitos de robótica, com o uso dos <i>softwares Tinkercad e Pictoblox</i>
16	Pós-teste	Analisar as soluções de programação dos estudantes através das linguagens <i>Python</i> e <i>Scratch</i>
17	Projeto Final	Orientar os participantes sobre a elaboração do projeto para finalizar o curso e acompanhar as apresentações

Tabela 1. Estrutura das aulas do curso.

- **Definição da metodologia das aulas:** Estabeleceu-se que todas as aulas de Scratch deveriam conter: (i) uma atividade lúdica associada ao conteúdo abordado; (ii) descrição explicativa sobre o conteúdo; (iii) exemplos; (iv) desafio final aos alunos; e (v) atividade de reforço para ser resolvida em casa.
- **Concepção das atividades lúdicas:** Com o objetivo de motivar os estudantes no início das aulas, definiu-se que o curso iria contar com a aplicação de atividades lúdicas, também denominadas ludificações. Foram planejadas para serem momentos dinâmicos de atividades práticas, referentes ao conteúdo a ser abordado na aula. Para isso, a equipe se reuniu e propôs a criação de uma atividade para cada aula que utilizaria a ferramenta Scratch, totalizando oito ludificações. Dentre as atividades lúdicas propostas, algumas consistem em computação desplugada – exercitar o conteúdo sem o uso do computador, outras não possibilitando praticar exatamente o conteúdo mas em contextos relacionados. A Figura 1 ilustra uma dessas atividades, que consiste em dividir os estudantes em equipes de até 5 integrantes, cujo tema refere-se à aula de Caneta e Sons. Nesta atividade, intitulada “Adivinhação Desenho e Som”, cada equipe recebe aleatoriamente um cartão indicando um objeto ou som. Caso seja um cartão de objeto, um dos integrantes da equipe deverá ir ao quadro e desenhá-lo, para que as demais equipes possam adivinhar. Se for um cartão de som, a dinâmica é similar, mas o objetivo é imitar o som da figura ilustrada no cartão.
- **Construção dos planos de aula:** Todas as aulas foram planejadas de forma individual. Para que os membros da equipe tivessem acesso e pudesse haver construção coletiva, foi utilizada como ferramenta a Planilha do Google. Cada plano de aula especificava a definição do tempo de aplicação de cada atividade, conteúdo, objetivo, metodologia, recursos e referências.
- **Elaboração das apresentações:** *Slides* de cada aula, produzido através do Google Apresentações, cujo material didático foi baseado nos livros de Varela [2017] e Marji [2014].

Objetivo: Estimular a interação dos estudantes com uma breve dinâmica de mímica e adivinhação, que consiste em fazer com que os participantes adivinhem performances de colegas eleitos.

Descrição: Dividir os participantes em equipes de até 5 pessoas, cada equipe irá receber um cartão que pode ter a categoria de objeto ou som, ao receber o cartão de objeto o membro escolhido terá que fazer este desenho no quadro para que outras equipes adivinhem, ao receber o cartão de som o membro escolhido terá que emitir o som para que outras equipes adivinhem, as demais equipes irão ter um tempo para anotar o que acham ser e depois revelar a resposta, a equipe que adivinhar receberá pontos e a equipe que foi adivinhada também.



Figura 1. Planejamento de atividade lúdica.

3.4.2. Aplicação do curso

Após o término do período de inscrição, iniciou-se a aplicação do curso, que ocorreu entre novembro de 2023 e março de 2024 na Universidade do Estado do Amazonas - Escola Superior de Tecnologia (UEA/EST). Um total de 21 aulas¹ foram ministradas, que ocorreram nas segundas e sextas-feiras, com duração entre 2 a 3 horas. Dos 27 estudantes que haviam realizado a inscrição no curso, 20 participaram de pelo menos uma aula. No total, 11 desses estudantes prosseguiram até o final do curso (40%), dos quais: 6 de Licenciatura em Computação (55%), 3 de Engenharia da Computação (27%) e 2 de Sistemas de Informação (18%). A Figura 2 ilustra alguns momentos das aulas do curso.



(a) Atividade lúdica



(b) Aula sobre funções no scratch com Meus Blocos



(c) Aula de Robótica

Figura 2. Registros da Aplicação.

¹ Apesar do escopo inicial possuir 17 aulas, 4 aulas extras foram realizadas, das quais: uma de reposição, duas para acompanhamento e uma adicional para apresentação de projeto final

3.4.3. Análise dos Dados

Com a aplicação do Curso de Scratch foi possível obter dados dos estudantes participantes do curso, bem como dos estudantes que planejaram e executaram o curso, que são os autores deste relato de experiência. Os dados relacionados aos estudantes que participaram do curso incluem atividades em sala, exercícios para casa que eram propostos em cada aula, desafios, tela e face dos estudantes gravadas com consentimento e *feedbacks* dos participantes.

Em relação à avaliação de aprendizagem desses participantes, no início do curso aplicou-se pré-testes para analisar o nível de conhecimento dos estudantes em lógica de programação. Em seguida, ao longo do curso, as atividades dos estudantes foram coletadas e analisadas a partir da ferramenta *Dr Scratch*. Além disso, realizou-se anotações durante as aulas, a partir da resolução de atividades pelos participantes e conversas informais. De forma geral, com os *feedbacks* analisados a partir dessas estratégias, pôde-se notar um avanço dos estudantes, principalmente a respeito à codificação. Isso pois, a medida que os conteúdos avançavam, os projetos elaborados pelos estudantes evoluíam, incorporando os novos conhecimentos abordados nas aulas. Ademais, por meio das anotações e comentários dos estudantes, pôde-se perceber que estavam compreendendo o assunto e tirando dúvidas pertinentes. Em aplicações posteriores, pretende-se incluir mais instrumentos de avaliação, para gerar indícios sobre a relevância ou não da ferramenta Scratch como estratégia de aprendizado.

Já os dados relacionados à equipe de elaboração e aplicação do curso, têm-se o *feedback* dos participantes do curso, que impactou nas habilidades pedagógicas dos instrutores do curso, como a comunicação, adaptação a imprevistos e crescimento pessoal. Também há atividades lúdicas autorais e planos de aulas que podem ser adaptados e melhorados para uma nova aplicação do curso posteriormente. Em momento futuro, pretende-se apresentar a análise desses dados e, por meio deles, identificar pontos positivos e negativos da aplicação da experiência.

4. Resultados e Discussão

Essa seção apresenta os resultados referentes ao planejamento e aplicação do curso de Scratch, que revelou não apenas desafios na produção dos conteúdos, mas também lições para o desenvolvimento profissional dos instrutores/monitores (*soft skills*), como destacam os tópicos a seguir, que foram definidos a partir de uma reunião realizada pelos instrutores do curso, com o intuito de pontuar os principais desafios e contribuições obtidos na realização das aulas, desde seu planejamento até a aplicação.

4.1. Lições aprendidas

Para fins de organização, as principais lições aprendidas foram agrupadas em cinco classes, como descritas a seguir:

Comunicação e interação com os estudantes: com a realização do curso houve contribuições que surgiram a partir da interação entre a equipe e com os estudantes. Essas interações permitiram melhoria na comunicação e nas habilidades de resolução de problemas, bem como o aprimoramento da didática ao lecionar as aulas. Esse fator contribuiu para que os monitores pudessem concluir o curso de acordo com os objetivos definidos

e organização. O tempo de curso proporcionou momentos de tomada de decisão onde paciência, interação com os diferentes estudantes e os *feedbacks* fossem essenciais para que a assistência prestada pelos monitores melhorasse gradativamente. Isso possibilitou que os monitores evoluíssem não somente na comunicação, mas também na assistência prestada aos estudantes dentro do curso.

Aprendizado e prática de programação: uma das principais contribuições para os instrutores do curso de Scratch, foi a prática profissional e atualização constante dos conceitos iniciais de programação. Ao lecionar para estudantes que enfrentam dificuldades na aprendizagem dos conteúdos introdutórios de programação, foi necessário tanto a compreensão dos respectivos assuntos, quanto uma constante revisão para entender onde estavam os maiores obstáculos que os estudantes enfrentaram ao aprender a lógica inicial de programação. Ao se deparar com desafios práticos junto com os participantes, os facilitadores melhoraram seu entendimento destes conceitos, tornando sua orientação pedagógica mais eficaz.

Adaptação a Imprevistos: tendo em vista a quantidade de eventos não planejados que aconteceram na aplicação das aulas, os instrutores puderam adaptar-se a imprevistos ao lecionar em um curso de programação. Por exemplo, problemas técnicos referentes à configuração de algumas máquinas no laboratório, causando dificuldade em alguns participantes ao utilizar o Scratch. Outro acontecimento inesperado estava relacionado ao tempo oferecido para o término de atividades, já que muitos estudantes tiveram dificuldades em finalizar os exercícios propostos pelos facilitadores no tempo previsto. Além disso, as imprevisibilidades relacionadas ao gerenciamento de tempo também aconteciam quando as discussões em sala de aula se estendiam além do esperado devido a dúvidas sobre tópicos específicos das aulas. Portanto, tais acontecimentos permitiram aos instrutores compreender a importância de realizar planejamentos flexíveis, que serão importantes na aplicação de cursos futuros.

Crescimento pessoal: a concepção do curso proporcionou melhorias na autoconfiança dos instrutores, devido ao *feedback* positivo dos participantes após o término de cada aula. A princípio, havia um certo receio unânime de que algo poderia falhar por conta da primeira experiência, mas com o passar das aulas, os facilitadores conseguiram ministrar o conteúdo proposto, segundo os estudantes, de forma clara e objetiva. Com isso, foi possível identificar pontos fortes e fracos de cada monitor, permitindo uma melhora significativa de dificuldades como timidez, insegurança e baixa autoestima.

Melhoria nas habilidades de trabalho em equipe: ser instrutor em um curso de programação possibilitou o desenvolvimento de habilidades como a colaboração na divisão de tarefas, troca de experiências e a facilitação do conteúdo proposto. Durante o curso, a troca de experiências entre os facilitadores enriqueceu o processo de aprendizagem, oferecendo diversas perspectivas e abordagens para o conteúdo apresentado. Ao longo das aulas, o desenvolvimento do material usado exigiu comunicação e coordenação entre os bolsistas da equipe, garantindo que os estudantes assimilassem os conceitos de forma eficaz. Essa colaboração fortaleceu os laços entre os instrutores, promovendo um ambiente de trabalho colaborativo e produtivo. Portanto, o curso de Scratch contribuiu para o crescimento pessoal e profissional dos instrutores envolvidos.

4.2. Desafios

Gestão de Tempo: a gestão de tempo representou um desafio para os monitores e instrutores ao desenvolver e conduzir as aulas planejadas. Com o intuito de promover exemplos e exercícios aos participantes, houve dificuldade em cumprir o tempo determinado para a realização das atividades, tendo em vista que alguns estudantes precisaram de mais tempo para concluir os exercícios propostos. As aulas tiveram que ser adaptadas pelos facilitadores para que os participantes pudessem compreender e executar os exemplos/exercícios. Esse desafio serviu como aprendizado e experiência para os próximos cursos, contribuindo para a definição de prazos realistas, flexibilidade durante as aulas e melhora da comunicação, tornando mais clara e acessível a todos os participantes.

Planejamento das Aulas: O curso de Scratch representou uma das primeiras experiências de cada monitor na elaboração das aulas. Assim, por se tratar de uma das primeiras práticas, os facilitadores ainda não possuíam uma base sólida no desenvolvimento de planos de aula ou atividades para os discentes praticarem o conteúdo. Esse fato evidencia-se no desenvolvimento das atividades lúdicas, que consistiam em momentos práticos, utilizando materiais como canetas, papéis, tinta, etc., devendo estar relacionados ao conteúdo do Scratch a ser abordado na aula. A construção dessas atividades tornou-se mais complexa a medida em que o tema da aula era mais avançado, como os blocos de “Sensores e variáveis”, exigindo um nível maior de abstração e criatividade. Além disso, outra dificuldade foi determinar a melhor maneira de organizar as aulas, seguindo uma sequência didática adequada, conforme estabelecido na Tabela 1. Os monitores precisavam compreender a importância de uma sequência para o aprendizado. Por fim, outro obstáculo no desenvolvimento das aulas foi a criação das atividades e exemplos para serem realizados na plataforma Scratch, pois estes eram de autoria própria e precisavam seguir uma ordem lógica. Para esta elaboração, além de ser necessário usar a criatividade, era preciso compreender a lógica por trás de cada exemplo e exercício, conectando aos blocos estudados e conseqüentemente, visando assim auxiliar o processo de aprendizagem dos participantes de forma gradual.

Evasão dos Estudantes: a aplicação das aulas do curso de Scratch apresentou um desafio inesperado: uma alta taxa de evasão. Realizado em época de intenso calendário acadêmico, o cronograma do curso coincidiu com o fim de período, uma fase trabalhosa para estudantes de graduação. Ademais, para alguns estudantes, o horário de 18 até as 20 horas pode não ter sido muito adequado, por questões de transporte e segurança. Outro fator que pode ter contribuído para essa alta taxa foi o fato das aulas acontecerem nos dias de segunda e sexta-feira, onde o intervalo de tempo entre cada aula era extenso, podendo resultar em desinteresse por parte dos participantes. Esses pontos evidenciaram a necessidade de medidas para manter os participantes engajados com o curso, como as atividades lúdicas no início de cada aula, o adiamento dos prazos para as entregas de exercícios e o contato contínuo para tirar as dúvidas dos estudantes. Portanto, a partir dessa experiência, foi possível observar pontos que podem ser melhor ajustados na aplicação de um novo curso, visando contornar a evasão dos estudantes, como realizar as aulas em horários menos tardios e em um intervalo menor de dias (segundas e quartas, por exemplo).

Experiência em Lecionar: Ao assumir o papel de facilitador em um curso, enfrentou-se uma série de desafios. Dentre eles, a necessidade de desenvolver habilidades de comunicação e gestão em sala de aula. Essa experiência demandou uma compreensão

mais profunda das necessidades e dificuldades individuais dos participantes, pois explicar conteúdos complexos de forma compreensível exigiu domínio do assunto, bem como a habilidade de identificar e abordar as dúvidas específicas de cada discente. A dinâmica de aprendizagem na área de computação é particularmente desafiadora devido à sua estrutura técnica, que muitas vezes pode ser um obstáculo para os estudantes. Portanto, tornou-se um desafio ao educador criar um ambiente de aprendizado inclusivo e adaptativo.

Dessa forma, a aplicação do curso possibilitou diversos achados, estruturados em desafios e contribuições. Esses achados estão em concordância com a Literatura, por exemplo: (i) nos trabalhos de Souza et al. [2023] e Galvão et al. [2019] há semelhanças nos desafios enfrentados durante a aplicação do curso, como o obstáculo na gestão de tempo necessária para conclusão das atividades propostas em sala de aula, a adaptação dos facilitadores a imprevistos e demanda de dúvidas e a necessidade constante de revisar conteúdos para sanar as dúvidas dos participantes. Além disso, ressalta-se que o o horário definido para aplicação do curso é um fator importante considerando a alta taxa de evasão; e (ii) em Figuerêdo et al. [2021] e Bastos [2020], há contribuições similares, como: o aprofundamento do conhecimento dos facilitadores durante a aplicação do curso, a melhora na comunicação e trabalho em equipe e a obtenção de experiência ao lecionar.

Entretanto, apesar das diversas contribuições deste estudo, é importante destacar uma ameaça à validade: os instrutores do curso – cujos desafios e contribuições foram apontados por eles – são autores do trabalho. Esse fator pode ter causado um viés que compromete a objetividade e imparcialidade da avaliação. Isso pois se trata da perspectiva e opinião dos próprios instrutores que, sem uma avaliação externa neutra, pode ter afetado a validade do estudo. Contudo, ressalta-se que dois especialistas, que já aplicaram o curso anteriormente, validaram os tópicos apontados pelos instrutores – minimizando o viés.

5. Considerações Finais

Tendo em vista a falta de profissionais no mercado de trabalho, as dificuldades enfrentadas durante a formação de licenciados em computação e a utilização do Scratch como alternativa lúdica para contornar os obstáculos ao aprender programação, este trabalho tratou de um relato de experiência sobre a aplicação de um curso de Scratch na Universidade do Estado do Amazonas (UEA) para estudantes de ensino superior que apresentavam dificuldades em disciplinas de programação, destacando desafios e contribuições que os facilitadores enfrentaram e adquiriram durante a aplicação das aulas.

Além disso, o trabalho destacou como a junção de conteúdos de programação associados a criação de projetos criativos é eficaz, tendo em vista os feedbacks dos participantes. Os resultados revelam descobertas importantes, destacando a contribuição dessa experiência prática para a formação acadêmica dos licenciandos em Computação.

Por fim, essa experiência reforçou a necessidade de adaptar as estratégias educacionais em Computação, destacando a importância de abordagens dinâmicas para promover a aprendizagem efetiva dos conceitos de programação, dando ênfase não apenas aos desafios enfrentados, mas também as contribuições e aprendizados proporcionados pelo curso de Scratch. Os próximos passos incluem o desenvolvimento de mais cursos/oficinas de Scratch e Robótica, visando não apenas o público universitário, mas também a comunidade externa, além de dar continuidade aos estudos e pesquisas.

Referências

- Bastos, V. d. C. (2020). Formação continuada de professores: Contribuições da oficina pedagógica para a aprendizagem do software scratch. *Revista Docência e Cibercultura*, 4(3):291–307.
- Brasil (2016). Resolução nº 5 de 16 de novembro de 2016.
- Brasscom (2021). Demanda de talentos em tic e estratégia tcem. <https://brasscom.org.br/pdfs/demanda-de-talentos-em-tic-e-estrategia-tcem/>. Accessed: 2024-03-04.
- Cárdenas-Cobo, J., Puris, A., Novoa-Hernández, P., Parra-Jiménez, Á., Moreno-León, J., and Benavides, D. (2021). Using scratch to improve learning programming in college students: A positive experience from a non-weird country. *Electronics*, 10(10):1180.
- Ferreira, L., Santos, S., and Bonfim, C. (2021). Pensamento computacional e programação scratch: uma revisão de literatura do sbie. In *Anais do VIII Encontro Nacional de Computação dos Institutos Federais*, pages 5–8, Porto Alegre, RS, Brasil. SBC.
- Figuerêdo, J. S. L., Machado, J. G., Lima, S. V., da Silva Cerqueira, C. S., and Pereira, C. P. (2021). Percepção do ensino-aprendizagem da monitoria de algoritmos e programação em cursos de engenharia na perspectiva de estudantes, monitores e professores. *Revista Brasileira de Informática na Educação*, 29:1433–1462.
- Freire, A. et al. (2016). 2.: Formação profissional do licenciado em computação: uma análise dicotômica. In *Anais do XIV congresso internacional de tecnologia na educação*, page 36.
- Galvão, A. P., da Silva, J. B., and Pereira, P. M. P. (2019). A construção de conhecimento no ensino de lógica de programação usando o scratch: um relato de experiência.
- Google for Startups (2024). Gap de talentos. Acessado em: 04 de março de 2024.
- Jesus, C., Santana, B., and Bittencourt, R. (2019). Oficinas de aprendizagem de programação com scratch e python em um curso de engenharia de computação. In *Anais do xxvii workshop sobre educação em computação*, pages 31–40. SBC.
- Junior, L., de Souza, J., et al. (2021). Uma análise das dificuldades de aprendizagem da lógica de programação no curso técnico integrado em informática do ifba-campus jacobina.
- Linhares, A. C. O. and Santos, K. S. (2021). A licenciatura em computação no brasil: histórica e contexto atual. *Revista Brasileira de Informática na Educação*, 29:188–208.
- Marji, M. (2014). *Learn to program with Scratch: A visual introduction to programming with games, art, science, and math*. No Starch Press.
- Monteiro, V. A. and Holanda, M. (2023). Pensamento computacional e scratch: Um relato de experiências com estudantes do ensino médio público no distrito federal. In *Anais do III Simpósio Brasileiro de Educação em Computação*, pages 254–261. SBC.
- 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 supe-

- rior: Uma revisão sistemática de literatura. *Research, Society and Development*, 9(10):e9429109287–e9429109287.
- Oliveira, W. and Cambraia, A. C. (2020). Desafios na formação de professores de computação: Reflexões e ações em construção. In *Anais do XXVI Workshop de Informática na Escola*, pages 319–328. SBC.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., and Kafai, Y. (2009). Scratch: Programming for all. *Communications of the ACM*, 52(11):60–67.
- Ribeiro, L., Castro, A., Fröhlich, A. A., Ferraz, C. A. G., Ferreira, C. E., Serey, D., Cordeiro, D. d. A., Aires, J., Bigolin, N., and Cavalheiro, S. (2019). Diretrizes da sbc para ensino de computação na educação básica. Relatório Técnico 001/2019, Sociedade Brasileira de Computação.
- Semesp, I. (2023). *Mapa do Ensino Superior - 13ª edição / 2023 – Instituto Semesp*, chapter Capítulo Especial: Cenários e Perspectivas da Área de TI no Ensino Superior.
- Silva, T. S. C. d., de Melo, J. C. B., and Tedesco, P. C. d. A. R. (2018). Um modelo para promover o engajamento estudantil no aprendizado de programação utilizando gamification. *Revista Brasileira de Informática na Educação*, 26(03):120.
- Souza, A. L. d., Neto, N. R., Eliote, Y. C. D., de Oliveira, E. G., de Classe, T. M., de Castro, R. M., Lima, A. A., and de Alcantara Gimenez, P. J. (2023). Desenvolvendo o pensamento computacional utilizando scratch: Um relato de experiência da formação de professores da educação básica. In *Anais do XXIX Workshop de Informática na Escola*, pages 918–929. SBC.
- Varela, H. (2017). *Scratch: Um jeito divertido de aprender programação*. Editora Casa do Código.
- Viana, G., Lopes, A., Portela, C., and Oliveira, S. (2019). Um survey sobre a aprendizagem de programação no curso de sistemas de informação. In *Anais do XXVII Workshop sobre Educação em Computação*, pages 161–175, Porto Alegre, RS, Brasil. SBC.
- Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3):33–35.