

Pensamento Computacional e Scratch: Um relato de Experiências com Estudantes do Ensino Médio Público no Distrito Federal

Vinicius Aguiar Monteiro, Maristela Holanda
viniciusaguiarmonteiro@hotmail.com, mholanda@unb.br
Universidade de Brasilia (UnB)

RESUMO

O ensino do Pensamento Computacional (PC) na Educação Básica Brasileira (EBB) é importante para o atendimento das competências e habilidades recomendadas pela Base Nacional Comum Curricular (BNCC) no Brasil para o ensino médio. Neste contexto, este artigo apresenta um relato de experiências da aplicação dos conceitos do PC, em duas etapas e três escolas públicas de ensino médio no Distrito Federal, utilizando a Linguagem de Programação Visual Scratch. O artigo descreve a metodologia proposta, o acompanhamento dos aprendizados dos estudantes e as lições aprendidas. Os resultados indicam que a metodologia teve boa aceitação e o curso contribuiu para o aumento dos conhecimentos gerais de informática dos estudantes. Dentre as lições aprendidas têm-se: o curso com um projeto único ao longo das aulas melhorou o engajamento dos estudantes; a necessidade de investimentos nos laboratórios de informática das escolas públicas; e a dificuldade do engajamento dos estudantes no ensino remoto.

CCS CONCEPTS

• **Social and professional topics** → Computing education.

PALAVRAS-CHAVE

Pensamento Computacional, Computação na Educação, Scratch, Base Nacional Comum Curricular

1 INTRODUÇÃO

A Base Nacional Comum Curricular (BNCC) [8] faz parte da política nacional da Educação Básica Brasileira (EBB) e contribui para a elaboração de políticas e ações em todos os níveis (Federal, Estadual e Municipal), além de auxiliar na elaboração dos conteúdos educacionais e critérios de oferta de infraestrutura adequada para o pleno desenvolvimento da educação.

Conforme as Normas sobre a Computação na EBB, complemento da BNCC, aprovadas em 17 de fevereiro de 2022, no parecer n.º 2/2022 da Câmara de Educação Básica (CEB) do Conselho Nacional da Educação (CNE), o domínio da construção de algoritmos e noções de decomposição para resolução de problemas deve ser alcançado entre o ensino fundamental (anos finais) e ensino médio [7].

Segundo Wing, em [23], o Pensamento Computacional (PC) é “um conjunto de habilidades, universalmente aplicável, que todos,

não apenas cientistas da computação, podem aprender e devem usar, pois é um processo para resolver problemas, projetar sistemas e compreender comportamentos a partir de ferramentas e técnicas sólidas, baseadas nos conceitos que fundamentam a Ciência da Computação” (tradução nossa).

As linguagens de programação Visual (LPV) são mais adequadas para inserir o PC na EBB, visto que diminuem a carga cognitiva exigida pela sintaxe das linguagens de programação tradicionais, permitindo aos estudantes se concentrar na lógica necessária para a resolução do problema [16]. Scratch é uma LPV que torna o aprendizado de programação mais fácil e interativo e, apesar de ter sido projetado especialmente para jovens (8 e 16 anos), tem sido utilizado por pessoas de todas as idades. No Scratch basta, conectar blocos de maneira lógica e interativa, formando a programação do ambiente para criar histórias, animações, jogos, simuladores, músicas e arte [17].

Nesse contexto, este artigo tem o objetivo de relatar as experiências de aplicação de uma metodologia de inserção do PC no ensino médio público no Distrito Federal, assim como o detalhamento da metodologia e discussões dos resultados e lições aprendidas.

Este artigo está dividido da seguinte forma. Na Seção 2, a fundamentação teórica é apresentada. Na Seção 3, a metodologia utilizada é apresentada. Na Seção 3, são relatadas as aplicações da metodologia, bem como detalhados os resultados, as discussões e as lições aprendidas. Na Seção 4, são apresentadas as conclusões e perspectivas de trabalhos futuros.

2 FUNDAMENTAÇÃO TEÓRICA

Wing [23] enfatiza que “pensar como um cientista da computação significa mais do que ser capaz de programar um computador. Requer pensar em vários níveis de abstração” (tradução nossa).

Para o desenvolvimento do PC, são consideradas as seguintes habilidades: coleta de dados, análise de dados, representação de dados, decomposição de problemas, abstração, algoritmo e procedimentos, automação, simulação e paralelismo. Essas habilidades foram definidas por *International Society for Technology in Education* (ISTE) e *Computer Science Teachers Association* (CSTA) [11].

O PC está inserido nas competências e habilidades da BNCC [8] para o ensino médio, trazendo um tópico específico sobre “As tecnologias digitais e a computação”. André [2] afirma que “o PC nas escolas permite que os alunos resolvam problemas, dividam-nos em problemas menores e criem algoritmos para solucioná-los”.

As LPVs levam os estudantes a uma forma de pensar computacionalmente baseada em estruturas, como laços de repetição, condicionais e outras bases algorítmicas herdadas das linguagens de programação tradicionais [6].

Bombsar *et al.* [6] mostra que os dados estatísticos entre 2006 e o início de 2015 confirmam o grande interesse da comunidade

Fica permitido ao(s) autor(es) ou a terceiros a reprodução ou distribuição, em parte ou no todo, do material extraído dessa obra, de forma verbatim, adaptada ou remixada, bem como a criação ou produção a partir do conteúdo dessa obra, para fins não comerciais, desde que sejam atribuídos os devidos créditos à criação original, sob os termos da licença CC BY-NC 4.0.

EduComp'23, Abril 24-29, 2023, Recife, Pernambuco, Brasil (On-line)

© 2023 Copyright mantido pelo(s) autor(es). Direitos de publicação licenciados à Sociedade Brasileira de Computação (SBC).

científica pelo ensino do PC na educação básica (K-12), e mostram que o uso das LPVs foi o mais utilizado nesse período.

O mapeamento sistemático, de metodologias de ensino e aprendizagem de programação para o ensino médio, feito por Khouri, Santos e Barbosa [15] chegou à conclusão de que 85% (12) dos estudos realizam atividades com o uso de computadores, sendo o Scratch a ferramenta mais utilizada.

Apesar do desafio de mensurar o nível do PC, autores defendem ser possível pela análise da complexidade dos projetos [9]. Para mensurar o PC nos projetos Scratch, o principal critério era a ausência ou presença de estruturas de programação para determinar o nível de competência do estudante em determinada habilidade [4].

O Dr. Scratch é uma ferramenta de análise automática de projetos Scratch que avalia as seguintes habilidades do PC: abstração e decomposição de problemas, raciocínio lógico, sincronização, paralelismo, controle de fluxo, interatividade do usuário e representação de dados. Cada categoria é atribuída uma pontuação de 1 a 3 em função da complexidade e frequência das estruturas e estratégias de programação utilizadas; se o projeto não atender ao requisito em nenhum nível, não receberá pontuação (nulo). Ao final, somam-se as pontuações parciais de cada categoria e classifica o desenvolvedor com a pontuação ≤ 7 como “básico”, > 8 e ≤ 14 como “em desenvolvimento” e como “proficientes” aqueles que têm pontuação > 15 [20].

Apesar das nove habilidades do PC, definidas por ISTE e CSTA, não corresponderem nominalmente às avaliadas pelo Dr. Scratch, é possível relacioná-las. A habilidade de: “Abstração e decomposição de problemas” é relacionado diretamente com “Abstração” e “Decomposição de problemas”; “Raciocínio lógico” pode se relacionar com todas as habilidades; “Sincronização” e a de “interatividade do usuário” não tem uma relação direta com as habilidades, visto que é mais relacionada a interação entre os personagens e cenários no Scratch, porém, pode-se associar, principalmente, a “Paralelismo”; “Paralelismo” tem o mesmo nome; “Controle de fluxo” se relaciona com as habilidades “Algoritmos e procedimentos” e “Automação”; “Representação de dados” se relaciona com “Coleta de dados”, “Análise de dados” e “Representação de dados”.

3 PROPOSTA E METODOLOGIA APLICADA

Curso e/ou oficina de programação é a metodologia mais utilizada para ensinar os conceitos da computação para o desenvolvimento do PC na EBB [5].

Foi aplicado o curso intitulado “Aprenda Pensamento Computacional e Scratch”, tendo como base: os objetivos, habilidades e competências da BNCC para o ensino médio; as Diretrizes para Ensino de Computação na Educação Básica da Sociedade Brasileira de Computação (SBC); e os relatos de experiências de cursos com objetivos, ferramentas, metodologias e público-alvo semelhantes aos propostos por Oliveira *et al.* [21], Schoeffel *et al.* [22], Marinho *et al.* [18] e Costa *et al.* [10].

Os planos de aula e o material didático elaborados estão disponíveis para consulta, respectivamente, em <https://bit.ly/3yTLMrQ> e <https://bit.ly/3Sfpx6n>. Os planos das cinco aulas e o material didático que compõem o curso foram desenvolvidos para o contexto

das aplicações relatadas neste artigo, porém é possível adaptá-las para futuras aplicações do curso para outros estudantes e contextos.

3.1 Objetivos do Curso

O objetivo geral do curso é o ensino dos conceitos do PC (coleta de dados, análise de dados, representação de dados, decomposição de problemas, abstração, algoritmo e procedimentos, automação, simulação e paralelismo) utilizando a LPV Scratch.

Os objetivos específicos são: ensinar os conceitos do PC; contextualizar os conceitos do PC à realidade dos estudantes; ensinar sobre as diferentes formas de representação de algoritmo (pseudocódigo, fluxograma e Scratch); ensinar sobre a utilização da ferramenta Scratch; ensinar os conceitos e ferramentas do Scratch; estimular o aprendizado de linguagens de programação no ensino médio; expandir as competências e habilidades em lógica de programação; e estimular o desenvolvimento criativo.

3.2 Conteúdo Programático e Carga Horária

Os conteúdos programáticos foram estruturados em dois módulos, detalhados a seguir, para alcançar os objetivos do curso:

O primeiro módulo aborda os conceitos do PC e de seus pilares (2 horas). O conteúdo proposto e o material didático foram adaptados a partir do curso “*Computational Thinking for Problem Solving*” da Universidade da Pensilvânia [12].

O segundo módulo aborda os conceitos, a utilização da plataforma Scratch e o desenvolvimento do projeto (8 horas). O conteúdo proposto foi baseado na proposta do artigo intitulado “*Utilização do Scratch como Ferramenta no Ensino de Pensamento Computacional para Crianças*” de Aono *et al.* [3], e o material didático foi baseado no “*Creative Computing Curriculum*”, desenvolvido por pesquisadores do Laboratório de Computação Criativa da Escola de Graduação em Educação de Harvard [14].

Para tentar reduzir as desistências e otimizar a ida dos estudantes às escolas, devido à quantidade de encontros, foi proposta a carga horária total de dez horas, sendo cinco encontros com duas horas cada. A divisão dos conteúdos e da carga horária foi pensada para que estes se adaptassem ao cenário educacional dos estudantes do ensino médio público e estão detalhados na Tabela 1.

3.3 Metodologia de Ensino do Curso

A implementação de metodologias ativas de aprendizagem no contexto do ensino de programação pode trazer diversos benefícios, como a maior aceitação, satisfação, interesse e motivação em programar, além da melhora no desempenho em trabalhos colaborativos e na habilidade de comunicação por parte dos estudantes [13]. O ensino contextualizado estabelece condições para que os estudantes também sejam responsáveis por sua própria aprendizagem [19].

A proposta do curso é que os estudantes desenvolvam as habilidades e competências do PC utilizando o Scratch em metodologia de aprendizagem ativa, pela qual são consideradas e aplicadas as seguintes abordagens e premissas do curso:

- Aprendizagem Ativa: Ao decorrer do curso, são feitos exercícios, perguntas, tarefas, e projetos que motivem os estudantes a procurarem o conhecimento necessário para atingir o objetivo proposto, trazendo o estudante para uma participação ativa na aquisição do conhecimento.

Tabela 1: Conteúdo programático e Carga Horária (CH) do curso. Adaptado de Davidson e Murphy [12] e Aono *et al.* [3].

Conteúdo	Descrição	CH (h)
Módulo I	CH (h) do módulo:	2h
Introdução do curso e ambientação	Apresentar o curso e conhecer os estudantes	2h
Pensamento computacional	Apresentar o significado de Pensamento Computacional	
Conceitos do pensamento computacional	Descrever os conceitos do pensamento computacional	
Fluxograma	Apresentar Conceitos de fluxograma e exemplos	
Pseudocódigo	Apresentar Conceitos de pseudocódigo e exemplos	
Módulo II	CH (h) do módulo:	8h
Introdução à programação e utilização do Scratch	Apresentar a programação para resolução de tarefas cotidianas para os estudantes, o ambiente de programação Scratch e os diferentes tipos de comandos para criação de cenários	2h
Conceitos de algoritmos e criação de figuras e efeitos visuais e sonoros com Scratch	Apresentar os conceitos de algoritmos, descrição narrativas de algoritmos, efeitos sonoros e visuais disponíveis no Scratch, utilização de camadas e criação de cenas com animações	
Procedimentos	Apresentar e Implementar procedimentos em Scratch e técnicas para construção de comandos	2h
Variáveis	Apresentar e Implementar tipos de dados no Scratch, criar e manipular dados	
Estruturas de decisão	Apresentar e Implementar técnicas para resolução de problemas, uso dos comandos de seleção e fluxo de controle	
Estruturas de repetição	Apresentar e Implementar estruturas de repetição para execução de comandos eficientes e variáveis de controle	2h
Processamento de strings	Apresentar e Implementar técnicas para processamento de strings	2h
Finalização do trabalho e apresentação	Finalizar o projeto e apresentar	
	CH (h) TOTAL:	10h

- **Aprendizagem Significativa:** Os exemplos mostrados são pensados para que se identifique a realidade da maioria dos estudantes, considerando o contexto tanto social quanto escolar de cada um deles e buscando seu interesse pelo conteúdo apresentado pela aprendizagem significativa.
- **Aprendizagem Colaborativa:** Na proposta do curso, foi previsto que os estudantes realizariam o projeto e as atividades em grupos de, no mínimo, duas pessoas. Assim, as trocas de ideias refinariam tanto o pensamento lógico quanto as soluções para os problemas. Dessa maneira, a construção do conhecimento também se tornaria muito mais efetiva, colaborativa e compartilhada.
- **Aprendizagem Baseada em Projetos:** Ao longo das aulas, os estudantes desenvolveram um projeto. O desenvolvimento do projeto é incremental em relação aos conteúdos repassados nas partes expositivas e práticas de cada aula.

3.4 Avaliação de Aprendizagem do Curso

A avaliação de aprendizagem do curso é realizada por questionários, elaborados pelos autores, e aplicados antes do curso (Disponível em: <https://bit.ly/3SfQuXI>) para entender o perfil dos estudantes e depois do curso (Disponível em: <https://bit.ly/3VBHFul>) para coletar a percepção e a avaliação dos estudantes sobre o curso. Também é feita a avaliação automática dos códigos produzidos nos projetos Scratch, a avaliação observacional e qualitativa com discussão do desempenho e a análise do que é produzido pelos estudantes [1], [4], [9].

A avaliação das competências do PC é feita pela análise automática dos códigos dos projetos produzidos no Scratch pela ferramenta *online* Dr. Scratch¹, visto que é uma ferramenta traduzida para o português e mais utilizada na avaliação automática das competências do PC em projetos Scratch [1]. A avaliação no Dr. Scratch é feita com o envio do arquivo com extensão “.sb2” ou a inserção do endereço do projeto no site do Dr. Scratch.

3.5 Planos de Aula do Curso

Os planos de aula possuem as informações do curso com: o nome do curso, o tema, a carga horária, os objetivos e os conteúdos previstos para a aula, bem como os conteúdos para cada parte da aula com: o tempo previsto, os materiais didáticos, e o detalhamento do que deve ocorrer, planejadamente e sequencial, se possível e, por fim, o método de avaliação e as referências bibliográficas da aula.

4 APLICAÇÕES DO CURSO

As experiências de aplicação do curso intitulado “Aprenda Pensamento Computacional e Scratch” foram feitas em duas etapas em três turmas e escolas diferentes. A primeira aplicação foi realizada em caráter experimental com uma turma na modalidade de ensino remoto e a outra aplicação foi feita em duas turmas, na modalidade presencial e incorporadas as melhorias da primeira aplicação.

4.1 Primeira Aplicação do Curso

A primeira aplicação do curso teve o objetivo de coletar dados e percepções para poderem ser feitas melhorias para a versão final

¹Dr. Scratch:<http://www.drscratch.org/>.

e as próximas aplicações do curso. O curso foi ministrado para estudantes do ensino médio, em uma escola, na modalidade de ensino remoto. As aulas foram ministradas entre os dias 26/10/2021 e 16/12/2021 com 14 encontros de 1 hora, cada.

Os estudantes registraram o interesse de participação em um formulário virtual, e, após a finalização do período de inscrições, 120 estudantes fizeram o preenchimento do formulário. Em seguida, os professores organizaram o repositório no Google Sala de Aula para armazenar os materiais relevantes para o curso. Após isso, os estudantes foram comunicados do local virtual das aulas.

Ao longo da aplicação do curso, houve dois acontecimentos que impossibilitaram a participação de muitos estudantes que estavam participando do curso até a segunda semana de aula (26/10/2021 a 09/11/2021), conforme ilustrado na Figura 1.

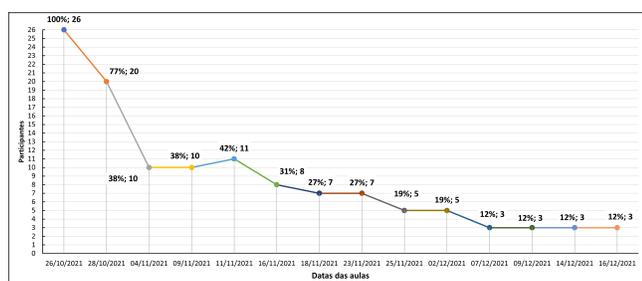


Figura 1: Estudantes presentes na primeira aplicação do curso.

A primeira ocorrência foi que todos os estudantes foram obrigados a retornar às aulas em formato híbrido (presencial e remoto) no dia 02/11/2021, causando assim o retorno confuso, visto que os estudantes nunca haviam vivenciado esse tipo de dinâmica escolar e ocasionando algumas desistências no curso.

A segunda ocorrência foi quando o governo local decretou o retorno 100% presencial, aproximadamente na terceira aula do curso, o que causou grande parte das desistências no curso. No dia 06/11/2021, foi feita uma consulta aos estudantes, por mensagem no WhatsApp, sobre o motivo das faltas nas aulas e os estudantes relataram que com o retorno das aulas presenciais estava muito difícil acompanhar o curso, visto o horário do curso. Considerando esse novo cenário, foi sugerido o novo horário para as aulas. A partir da quarta aula o horário do curso mudou, porém, mesmo com a mudança de horário, a participação caiu de 20 para 10 estudantes da segunda para a terceira aula. Da quarta aula em diante as desistências (de 10 para 3 estudantes) foram gradativas.

As ocorrências podem ter causado a maioria das desistências dos estudantes, outra hipótese que pode ter inviabilizado a participação dos estudantes é que apenas 11 (44%) dos 26 estudantes que participaram da primeira aula possuíam computador para utilizar durante o curso. Além disso, a interação dos estudantes durante as aulas foi baixa, o que pode ter sido uma barreira para continuidade e finalização do curso.

Ao final, 3 estudantes finalizaram o curso e entregam o projeto final proposto. Os projetos foram analisados na ferramenta Dr. Scratch e os estudantes “A”, “B” e “C”, obtiveram, respectivamente, as notas 12, 18 e 18 do total de 21 pontos.

O desempenho dos estudantes foi satisfatório, tanto pela análise automática dos projetos no Dr. Scratch quanto pelo produto apresentado, pelo domínio na apresentação dos códigos e o desempenho durante as aulas.

Na última aula, foi relatado pelos estudantes que o curso foi muito longo e que não gostaram de o desenvolvimento do projeto ter ocorrido somente a partir da décima aula, o que pode ter sido o motivo da desistência de muitos estudantes, além das intercorrências causadas pela COVID-19 que impactaram diretamente o andamento e participação dos estudantes. Relataram também que aprenderam o conteúdo do curso, o que pode ser confirmado com o processo avaliativo do curso.

4.2 Segunda Aplicação do Curso

A segunda aplicação do curso foi realizada em duas escolas, cada uma com uma turma, na modalidade presencial, e teve o objetivo aplicar a versão melhorada e final, além de avaliar o curso, considerando as alterações realizadas após a primeira aplicação.

Algumas questões de organização, planejamento e material didático da primeira aplicação difere da utilizada na segunda, visto que a primeira foi ministrada em caráter de teste para melhorias do curso e a segunda é a proposta final, refinada a partir do que foi observado e aprendido.

As melhorias e ajustes foram motivadas pelo retorno dos estudantes coletados ao final do curso, pela quantidade de desistências no curso, pela natureza de cada modalidade, pela dinâmica das aulas e pela dificuldade de desenvolvimento do projeto individualmente. As diferenças de organização e planejamento entre as aplicações estão detalhadas na Tabela 2.

Tabela 2: Diferenças entre a primeira e segunda aplicação.

Item	1ª aplicação	2ª aplicação
Carga horária total	14 horas	10 horas
Carga horária por aula	1 hora	2 horas
Divisão do conteúdo	3 módulos (Pensamento Computacional, Scratch e desenvolvimento do projeto final)	2 módulos (Pensamento Computacional e Scratch)
Desenvolvimento do Projeto	Individual	Grupo
Modalidade de ensino	Remoto	Presencial
Quantidade de professores	2 professores	1 professor

O público-alvo na escola “A” foram estudantes do segundo e terceiro ano do ensino médio e na escola “B” eram do primeiro ano do ensino médio, possibilitando atingir todo o ensino médio. Como o curso foi ministrado em apenas uma turma em cada escola, houve uma limitação na quantidade de participantes em cada escola, visto que na escola “A” 15 computadores funcionavam e na escola “B”, apenas 5.

4.3 Segunda Aplicação do Curso - Escola "A"

Na Escola "A", o registro de interesse e participação no curso foi realizado em formulário virtual. Apesar de ter tido 23 inscritos (82,6%) eram do terceiro ano e 4 (17,4%) do segundo ano), apenas 15 estudantes compareceram na primeira aula. Ao longo das aulas houve algumas desistências gradativamente, conforme presenças ilustradas na Figura 2. As aulas foram realizadas entre 02/05/2022 e 30/05/2022 com 5 encontros de 2 horas, cada.

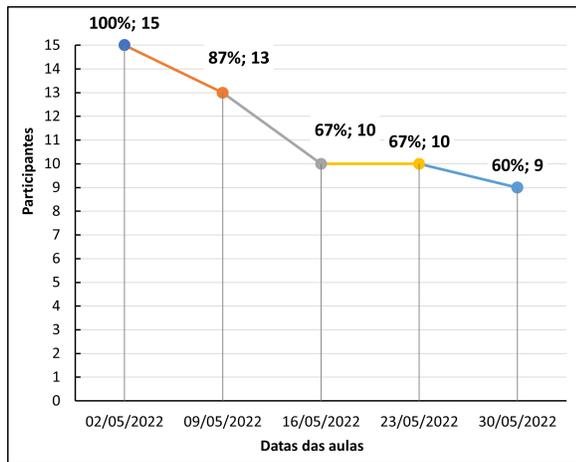


Figura 2: Estudantes presentes na segunda aplicação do curso - Escola "A".

Na primeira aula os estudantes preencheram o formulário de caracterização do perfil dos estudantes para melhor condução das aulas. Alguns resultados do perfil dos estudantes merecem destaque: apenas 22% (2) estudantes possuíam computador de uso exclusivo, ou seja, 78% (7) dos estudantes realizavam atividades apenas na escola no horário do curso; percebeu-se que 89% (8) dos estudantes não possuíam nenhum conhecimento em programação, o que reforça que o Scratch foi uma ótima escolha para introdução à programação nessa turma; e dos 7 (78%) estudantes que declararam serem bons em raciocínio lógico, 5 (71%), informaram que Matemática é a disciplina que eles mais gostam/tem facilidade de aprender, o que pode ter relação com afinidade com a área de Ciências Exatas.

Ao longo das aulas os estudantes foram divididos em grupos para desenvolvimento do projeto final do curso. Na aplicação do curso pôde-se perceber que os estudantes estavam engajados na elaboração do projeto, tirando dúvidas e discutindo as soluções nos grupos. Após o curso, foi realizada a análise dos códigos produzidos pelos 9 estudantes (5 grupos) que efetivamente finalizaram o curso e entregam o projeto final, no Dr. Scratch. Observa-se que os estudantes, em geral, obtiveram notas satisfatórias, conforme detalhado a seguir: grupo 1 - 18 pontos; grupo 2 - 16 pontos; grupo 3 - 16 pontos; grupo 4 - 14 pontos; grupo 5 - 11 pontos.

Em relação ao desempenho dos grupos, o Grupo 1 obteve a maior pontuação dentre os projetos apresentados. O maior desempenho pode ter sido motivado porque eles possuíam computador para uso exclusivo e *internet* em casa, oportunizando mais tempo para estudar e desenvolver o projeto. O estudante do Grupo 4 finalizou o projeto sozinho, visto que sua dupla desistiu.

Após a finalização do curso os 9 estudantes preencheram o questionário que teve o objetivo de coletar a percepção dos estudantes em relação ao curso. Os resultados demonstram que os estudantes gostaram do curso, em seus diversos aspectos, conforme ilustrado na Figura 2.

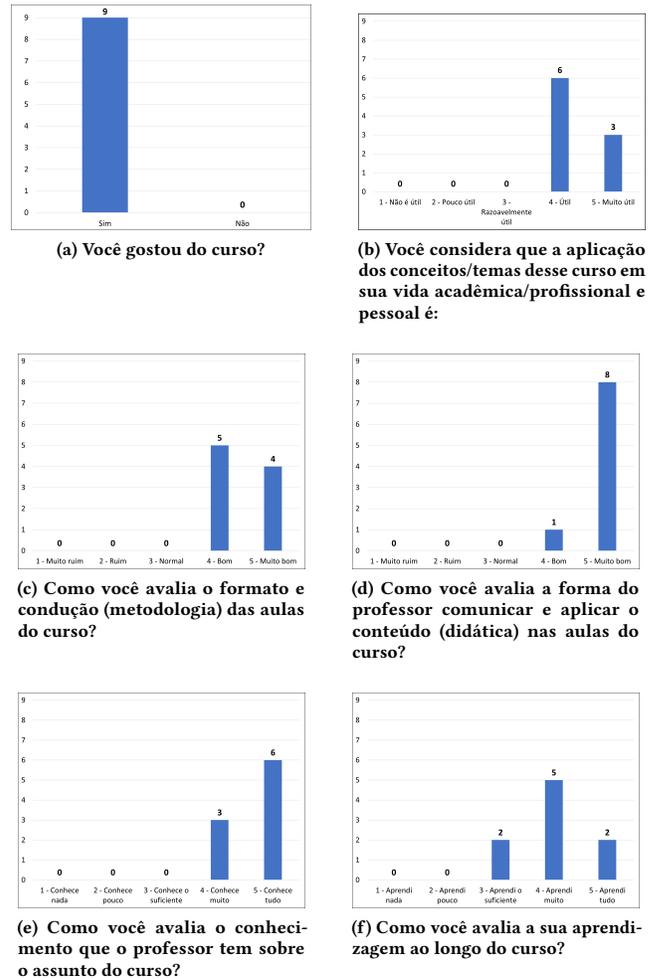


Figura 3: Avaliação final dos estudantes - Escola "A".

A pergunta sobre o nível de conhecimento em informática dos estudantes foi aplicada tanto no questionário anterior quanto no posterior ao curso e 78% (7) dos estudantes consideraram que, após o curso, seu nível de conhecimento em informática aumentou, em média, 2 pontos.

4.4 Segunda Aplicação do Curso - Escola "B"

Na Escola "B", o registro de interesse e participação no curso foi realizado no e-mail da professora responsável pelo laboratório de informática da escola. Ao final das inscrições 7 estudantes registraram interesse, porém 5 compareceram na primeira aula. Ao longo das aulas houve apenas uma desistência, conforme presenças ilustradas

na Figura 4. As aulas na Escola “B” foi realizado entre 21/04/2022 e 09/06/2022 com 5 encontros de 2 horas, cada.

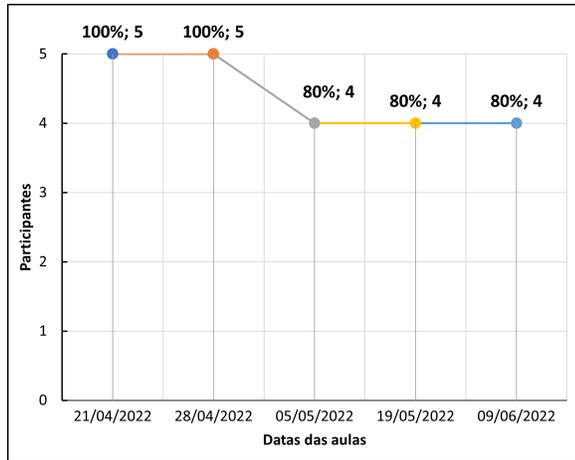


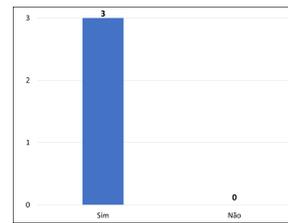
Figura 4: Estudantes presentes na segunda aplicação do curso - Escola “B”.

Apenas 3 (60%) dos 5 estudantes que finalizaram o curso preencheram o questionário de caracterização de perfil dos estudantes. Todos os 3 (100%) estudantes: não possuíam computador em casa, ou seja, todas as atividades foram desenvolvidas em sala no horário do curso; não possuíam conhecimento em programação, reforça que o Scratch foi uma ótima escolha para essa escola; se considera bom em raciocínio lógico; e mais gosta/tem facilidade de aprender na disciplina de Matemática, o que pode ter relação com a afinidade na área de Ciências Exatas.

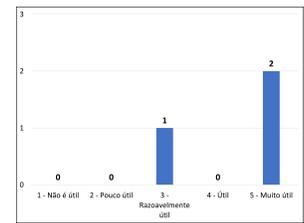
Na aplicação do curso, pôde-se perceber que os estudantes estavam engajados na elaboração do projeto e tiravam dúvidas e discutiam sobre as soluções dos problemas, porém com dificuldades. Após o curso foi realizada a análise dos códigos produzidos pelos 5 estudantes (2 projetos) que efetivamente finalizaram o curso e entregaram o projeto final e observa-se que mesmo com as dificuldades, obtiveram notas satisfatórias. Os projetos dos 2 grupos foram analisados na ferramenta Dr. Scratch e obtiveram as seguintes notas: grupo 1 - 10 pontos; e grupo 2 - 11 pontos.

Após a finalização do curso, 3 dos 5 estudantes que finalizaram o curso preencheram o questionário final de avaliação que teve o objetivo de coletar a percepção dos estudantes em relação ao curso. Os resultados mostram que os estudantes gostaram do curso, em seus diversos aspectos, conforme ilustrado na Figura 5.

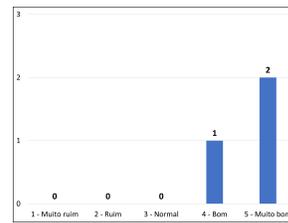
A pergunta sobre o nível de conhecimento em informática dos estudantes foi aplicada nos questionários antes e depois do curso, possibilitando uma comparação da percepção do conhecimento geral em informática pelos estudantes. Observa-se que 100% (3) dos estudantes consideraram que, após o curso, seu nível de conhecimento em informática aumentou em média 2 pontos.



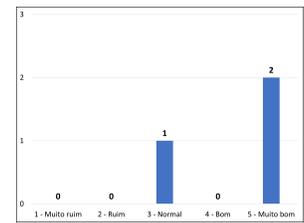
(a) Você gostou do curso?



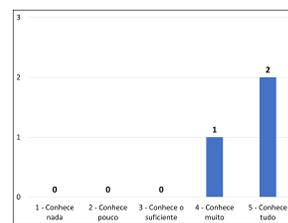
(b) Você considera que a aplicação dos conceitos/temas desse curso em sua vida acadêmica/profissional e pessoal é:



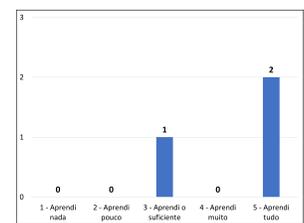
(c) Como você avalia o formato e condução (metodologia) das aulas do curso?



(d) Como você avalia a forma do professor comunicar e aplicar o conteúdo (didática) nas aulas do curso?



(e) Como você avalia o conhecimento que o professor tem sobre o assunto do curso?



(f) Como você avalia a sua aprendizagem ao longo do curso?

Figura 5: Avaliação final dos estudantes - Escola “B”.

4.5 Discussões e Lições Aprendidas das aplicações

As aplicações do curso em duas etapas, três escolas e modalidades diferentes possibilitou diferentes análises, discussões e lições aprendidas. A primeira aplicação foi feita na modalidade de ensino remoto e a segunda aplicação na modalidade presencial. As principais percepções dos autores nas diferenças encontradas conforme a modalidade de ensino estão detalhadas na Tabela 3.

A segunda aplicação do curso possibilitou uma comparação do desempenho no desenvolvimento dos projetos nas duas escolas, visto que os projetos dos estudantes do segundo e terceiro ano (Escola “A”) obtiveram desempenho superior aos do primeiro ano (Escola “B”). Ao analisar as habilidades avaliadas pelo Dr. Scratch na Figura 6, observa-se que “Abstração”, “Lógica” e “Paralelismo” foram as três habilidades que os estudantes mais tiveram dificuldades de implementar nos projetos, levando às hipóteses de investigação de quais as habilidades que o curso não abordou de deveria ou que

Tabela 3: Percepções dos Autores entre as diferenças das modalidades de ensino das experiências.

Ensino Remoto	Ensino Presencial
Menor participação dos estudantes	Maior participação dos estudantes
Dificuldade de desenvolvimento do projeto em grupos	Facilidade de desenvolvimento do projeto em grupos
Dificuldade de identificar se os estudantes possuem dúvidas	Maior facilidade de identificar os estudantes possuem dúvidas
Distanciamento da relação estudante-professor	Proximidade da relação estudante-professor
Maior quantidade de desistências	Menor quantidade de desistências

os estudantes possuem mais dificuldades de aprender, independente da abordagem.

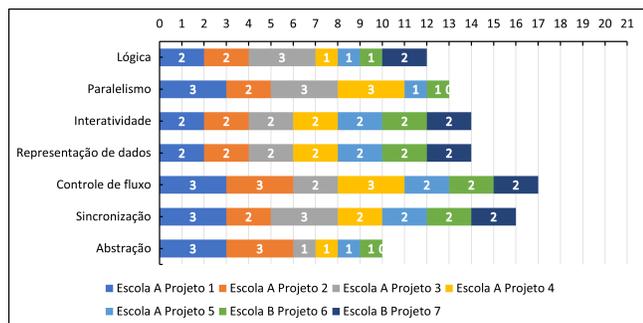


Figura 6: Comparação dos projetos por habilidade analisada no Dr. Scratch.

Outro dado importante, que merece destaque, é que 8 (67%) dos 12 (100%) estudantes relataram que gostam/tem mais facilidade de aprender Matemática, o que pode levar a hipótese da relação de afinidade dos estudantes com Ciências Exatas.

Os estudantes da segunda aplicação relataram que gostaram do curso, em diversos aspectos, e consideraram que a aplicação do curso contribuiu para o aumento do conhecimento em informática, com uma média de mais 2 pontos em suas percepções.

As escolas da segunda aplicação possuíam infraestrutura de computadores e *internet*, porém precárias e não suportava tantas conexões, logo isso foi limitante e uma das principais dificuldades encontradas na aplicação do curso. O investimento em recursos tecnológicos e de conexão trariam possibilidades e ferramentas para os professores da escola desenvolverem as competências e habilidades de todas as disciplinas. Além disso, o investimento proporcionaria a inclusão e o letramento digital, visto que 9 (75%) dos 12 (100%) estudantes da segunda aplicação não possuíam computador ou eram de uso compartilhado com seus familiares.

O curso, em geral, cumpriu com seu objetivo de aprendizagem nessa escola, conforme mostram as pontuações obtidas nos projetos, bem como os dados analisados dos questionários, apesar de ter tido uma baixa adesão nos preenchimentos.

As principais lições aprendidas a partir das experiências de aplicação do curso nas três escolas, nas duas aplicações, são: o professor precisa criar um ambiente propício a tirar dúvidas para que os estudantes se sintam à vontade para perguntar, independente da modalidade de ensino; os estudantes se sentem mais próximos e acolhidos pelo professor ao ter um diálogo aberto, amigável e sendo presencial, favorecendo a aprendizagem e diminuindo as desistências; o professor precisa estar preparado para adaptar sua proposta conforme a realidade e contexto social na qual está inserido, independente da modalidade de ensino; apesar de se ter o curso com material didático padrão, é preciso considerar o contexto da turma e fazer algumas adequações, por isso a importância de entender o perfil dos estudantes; e os estudantes gostam de produzir produtos criativos, logo é preciso dar liberdade, mesmo que guiada, para que os estudantes desenvolvam o projeto que seja representativo para a vivência de ensino-aprendizagem.

5 LIMITAÇÕES

A quantidade e qualidade dos computadores e a estabilidade e velocidade da *internet* nas escolas limitou a quantidade de estudantes que puderam participar das atividades, porém por se tratar de turmas iniciais do curso, serviram como teste para futuras aplicações. Outra limitação que impactou a oferta do curso foi pelo fato de ele ter sido ministrado no turno contrário de estudo dos estudantes, fazendo com que eles fossem para a escola em outro turno.

6 CONSIDERAÇÕES FINAIS

O presente artigo elaborou uma metodologia para ensinar o Pensamento Computacional em escolas públicas de ensino médio, bem como relatou as experiências de sua aplicação. O Pensamento Computacional e o Scratch foram conceituado e contextualizado na Educação Básica Brasileira, especificamente no ensino médio.

O curso “Aprenda Pensamento Computacional e Scratch” foi elaborado, segundo a literatura e contexto local, e aplicado em duas etapas e em três escolas.

Os dados gerados na metodologia de avaliação da aprendizagem dos estudantes mostrou que o curso cumpriu com o seu objetivo, considerando as notas dos projetos desenvolvidos e avaliados no Dr. Scratch. Os dados coletados a partir da percepção dos estudantes sobre o curso mostram que o curso recebeu a média de 4, em uma escala de 1 a 5, em diversas dimensões de análise do curso, o que demonstra que os estudantes gostaram do curso.

Conforme mostrado nas aplicações do curso, mesmo que as escolas possuam infraestrutura de computadores e *internet*, ela pode ser precária e não suportar tantas conexões, logo isso pode ser um limitante e deve ser analisado antes de iniciar uma aplicação do curso em alguma escola.

As perspectivas de trabalhos futuros a partir deste trabalho são: aplicar o curso em outras escolas públicas para serem gerados mais dados para analisá-los; aprimorar o curso com base nas aplicações e coletas e análises de dados futuros; e elaborar uma versão para aplicação do curso em escolas sem infraestrutura adequada para aplicação do curso.

REFERÊNCIAS

- [1] Nathalia da Cruz Alves, Christiane Gresse Von Wangenheim, and Jean C.R. Hauck. 2019. Approaches to Assess Computational Thinking Competences Based on Code Analysis in K-12 Education: A Systematic Mapping Study. *Informatics in Education* 18, 1, 17–39.
- [2] Claudio F André. 2018. O pensamento computacional como estratégia de aprendizagem, autoria digital e construção da cidadania. *teccogs - Revista Digital de Tecnologias Cognitivas* 18, 94–109. <https://revistas.pucsp.br/index.php/teccogs/article/view/48579/32061>
- [3] Alexandre Aono, Hugo Rody, Daniela Musa, Vanessa Pereira, and Jurandy Almeida. 2017. A Utilização do Scratch como Ferramenta no Ensino de Pensamento Computacional para Crianças. In *Anais do XXV Workshop sobre Educação em Computação* (São Paulo). SBC, Porto Alegre, RS, Brasil, 2169–2178.
- [4] Ana Araujo, Wilkerson Andrade, and Dalton Guerrero. 2016. Um Mapeamento Sistemático sobre a Avaliação do Pensamento Computacional no Brasil. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* 5, 1, 1147.
- [5] Christiano Avila, Adriana Bordini, Monica Marques, and Luciana Cavalheiro, Simone e Foss. 2016. Desdobramentos do Pensamento Computacional no Brasil. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)* 27, 1, 200.
- [6] James Bombasar, André Raabe, Elisangela Miranda, and Rafael Santiago. 2015. Ferramentas para o Ensino-Aprendizagem do Pensamento Computacional: onde está Alan Turing? *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)* 26, 1, 81.
- [7] BRASIL. 2022. *Parecer CNE/CEB n.º 2/2022 - Normas sobre Computação na Educação Básica – Complemento à Base Nacional Comum Curricular (BNCC)*. Brasília, DF: MEC, Brasília, Brasil. http://portal.mec.gov.br/index.php?option=com_docman&view=download&alias=235511-pceb002-22&category_slug=fevereiro-2022-pdf&Itemid=30192
- [8] Ministério da Educação BRASIL. 2018. *Base Nacional Comum Curricular*. Brasília, DF, Brasília, Brasil. <http://basenacionalcomum.mec.gov.br/>
- [9] Karen Brennan and Mitchel Resnick. 2012. New frameworks for studying and assessing the development of computational thinking. In *In AERA 2012*. American Educational Research Association meeting, Vancouver, BC, Canada, 25. <http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf>
- [10] Newerlyson Correia Costa, José Ricardo Santos de Lima, Gylmara Kylma Feitosa Carvalhêdo Almeida, Yonara Costa Magalhães, and Will Ribamar Mendes Almeida. 2022. O uso da plataforma Scratch como ferramenta facilitadora durante o ensino de lógica de programação para alunos do ensino médio: The use of the Scratch platform as a facilitating tool during the teaching of programming logic for high school students. *Brazilian Journal of Development* 8, 8, 59279–59293.
- [11] ISTE CSTA. 2011. *Computational Thinking in K–12 Education: teacher resources* (2 ed.). ISTE, CSTA, ISTE. 66 pages. https://cdn.iste.org/www-root/2020-10/ISTE_CT_Teacher_Resources_2ed.pdf
- [12] Susan Davidson and Chris Murphy. 2022. *Computational Thinking for Problem Solving*. Vol. 1. Coursera, Philadelphia, Pennsylvania, United States. <https://www.coursera.org/learn/computational-thinking-problem-solving>
- [13] Scott Freeman, Sarah L. Eddy, Miles McDonough, Michelle K. Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the National Academy of Sciences* 111, 23, 8410–8415. arXiv:<https://www.pnas.org/doi/pdf/10.1073/pnas.1319030111>
- [14] Creative Computing Lab Harvard. 2019. *Creative Computing Curriculum. Desenvolvido por Laboratório de Computação Criativa da Escola de Graduação em Educação de Harvard e Traduzido por Rede Brasileira de Aprendizagem Criativa (aprendizagemcriativa.org) e Universidade Estadual de Feira de Santana (UEFS), Brasil* 1, 1, 192. <https://lcl.media.mit.edu/resources/readings/creative-computing-guide.pt.pdf>
- [15] Cátia Mesquita Brasil Khouri, Gidevaldo Novais dos Santos, and Maria Silva Santos Barbosa. 2020. Mapeamento Sistemático em Metodologias de Ensino-aprendizagem de Programação. *Revista de Ciência da Computação* 2, 1, 13–27.
- [16] Sze Yee Lye and Joyce Hwee Ling Koh. 2014. Review on teaching and learning of computational thinking through programming: What is next for K-12? *Computers in Human Behavior* 41, 51–61.
- [17] Marji Majed. 2014. *Learn to Program with Scratch*. Novatec Editora, São Francisco, Califórnia, EUA.
- [18] Anna Raquel da S. Marinho, Pauleany S. de Moraes, Givanaldo R. de Souza, and Alba S. L. do Nascimento. 2018. Relato de Experiência Vivenciada no PIBID sobre a Utilização da Computação Desplugada, a Hora do Código e do Scratch no Ensino Médio. In *Anais do XXVI Workshop sobre Educação em Computação* (Natal). SBC, Porto Alegre, RS, Brasil, 10.
- [19] Joel Michael. 2006. Where's the evidence that active learning works? *Advances in Physiology Education* 30, 4, 159–167. arXiv:<https://doi.org/10.1152/advan.00053.2006> PMID: 17108243.
- [20] Jesús Moreno-León, Gregorio Robles, and Marcos Román-González. 2015. Dr. Scratch: Automatic Analysis of Scratch Projects to Assess and Foster Computational Thinking. *RED-Revista de Educación a Distancia* 46, 23. https://www.researchgate.net/publication/281714025_Dr_Scratch_Automatic_Analysis_of_Scratch_Projects_to_Assess_and_Foster_Computational_Thinking
- [21] Milena de Oliveira, Anderson de Souza, Aline Ferreira, and Emanuel Barreiros. 2014. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. In *Anais do XXII Workshop sobre Educação em Computação* (Brasília). SBC, Porto Alegre, RS, Brasil, 239–248. <https://sol.sbc.org.br/index.php/wei/article/view/10978>
- [22] Pablo Schoeffel, Paolo Moser, Geraldo Varella, Leticia Durigon, Gustavo Albuquerque, and Matheus Niquelatti. 2015. Uma Experiência no Ensino de Pensamento Computacional para Alunos do Ensino Fundamental. *Anais dos Workshops do Congresso Brasileiro de Informática na Educação* 4, 1, 1474.
- [23] Jeannette M. Wing. 2006. Computational Thinking. *Commun. ACM* 49, 3, 33–35.