

RECORTE DE UM PROCESSO ENSINO E APRENDIZAGEM DE PROGRAMAÇÃO COM O JOGO SCRATCH EM UMA ESCOLA TÉCNICA DO INTERIOR DO ESTADO DE SÃO PAULO: UMA PESQUISA-AÇÃO

EXCERPT FROM A PROGRAMMING TEACHING AND LEARNING PROCESS WITH THE SCRATCH GAME IN A TECHNICAL SCHOOL IN THE INTERIOR OF THE STATE OF SÃO PAULO: AN ACTION RESEARCH

940

Jonas José Oliveira Duzo¹, Reginaldo Rideaki Kamiya²

1- Especialista em Pedagogia pelo Centro Universitário UNIFIEO, graduado em Tecnologia da Gestão da Tecnologia da Informação e docente da ETEC de Itapira João Maria Stevanatto; 2- Mestre em Ciências da Computação (USP) e docente no Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro (IFRJ).

Contato: jonas10duzo@gmail.com

RESUMO

Este artigo apresenta um estudo de caso sobre a aplicação do jogo *Scratch* em um processo de ensino e aprendizagem de programação em uma escola técnica localizada no interior do estado de São Paulo. Utilizando uma abordagem de pesquisa-ação, o estudo investigou como o *Scratch*, uma ferramenta de programação visual, pode facilitar a compreensão de conceitos de programação entre os alunos. O objetivo foi avaliar o impacto do uso do jogo na aquisição de habilidades de programação e no engajamento dos estudantes com o conteúdo. A pesquisa envolveu a implementação de atividades práticas e a coleta de dados sobre o desempenho dos alunos e suas percepções em relação ao aprendizado. Os resultados indicam que o uso do *Scratch* não apenas melhorou a compreensão dos conceitos de programação, mas também aumentou o interesse dos alunos pela disciplina. O estudo contribui para a discussão sobre a eficácia de ferramentas lúdicas no ensino de programação e fornece insights sobre a aplicação de metodologias ativas em contextos educacionais técnicos.

Palavras-chave: Ensino de Programação. Jogo *Scratch*; Pesquisa-Ação. Ensino. Técnica Aprendizagem Ativa.

ABSTRACT

This paper presents a case study on the application of the Scratch game in a programming teaching and learning process at a technical school located in the interior of the state of São

Paulo. Using an action research approach, the study investigated how Scratch, a visual programming tool, can facilitate the understanding of programming concepts among students. The objective was to evaluate the impact of using the game on the acquisition of programming skills and on students' engagement with the content. The research involved the implementation of practical activities and the collection of data on students' performance and their perceptions regarding learning. The results indicate that the use of Scratch not only improved the understanding of programming concepts, but also increased students' interest in the subject. The study contributes to the discussion on the effectiveness of playful tools in teaching programming and provides insights on the application of active methodologies in technical educational contexts.

Keywords: Programming Teaching. Scratch Game; Action Research. Teaching. Active Learning Technique.

1 INTRODUÇÃO

Esta pesquisa teve início a partir da observação realizada nas aulas de programação com os alunos ingressantes do primeiro ano do ensino médio em uma escola técnica do interior do estado de São Paulo, onde os alunos apresentaram uma extrema dificuldade em aprender a lógica de programação.

Há uma ideia equivocada sobre as habilidades naturais dos nativos digitais no uso de tecnologias. De acordo com as observações dos autores em citação, os jovens passam uma parte significativa de seu tempo consumindo tecnologia, seja através de redes sociais ou jogos. No entanto, esse comportamento não os torna automaticamente proficientes em tecnologia. Embora eles desenvolvam habilidades para consumir informações de forma eficaz, filtrando e assimilando conteúdos tecnológicos, isso não se traduz em uma capacidade de criar conhecimento inovador. Em outras palavras, esses indivíduos não necessariamente desenvolvem a habilidade de explorar plenamente o potencial criativo das tecnologias disponíveis (RESNICK et al., 2012 apud PONTES, 2019, p. 744).

Evidencia-se que o uso do computador favorece a aprendizagem ao combinar a interação entre o aluno, o software ou aplicativo utilizado e o conteúdo apresentado. Recursos diversos, como simulações, jogos educativos, acesso a informações em tempo real e ferramentas de colaboração, estimulam diferentes aspectos cognitivos dos estudantes. A interatividade e a possibilidade de feedback imediato são aspectos

essenciais, permitindo que os alunos experimentem, cometam erros e aprendam com eles de maneira dinâmica. Além disso, essas ferramentas ajudam no desenvolvimento de habilidades como pensamento crítico, resolução de problemas, colaboração e criatividade (DOBGENSKI; PRADO; SILVA, 2024). Nesse contexto, é evidente a importância de incentivar os estudantes a pensar de forma significativa. O Pensamento Computacional é uma excelente alternativa para isso, pois esse tipo de pensamento lógico e racional, característico dos profissionais de Ciência da Computação, busca estratégias eficazes para resolver problemas, aproveitando os recursos computacionais disponíveis (CERATI, 2024).

A instituição escolar precisa adotar uma abordagem diferenciada, mantendo-se vigilante em relação às evoluções tecnológicas, de modo a minimizar o impacto que estas podem exercer. Dentro desse contexto educacional, os educadores se veem diante da necessidade de reavaliar o processo de ensino e aprendizado, desenvolvendo abordagens inovadoras que aproveitem a informática como uma ferramenta de aprendizagem contemporânea (PASQUAL JÚNIOR, 2005).

Grandes organizações têm reconhecido a importância de ensinar programação na educação básica. O Google tem investido muito em programas e projetos que estão lotados na divisão Google na Educação, desenvolvidos para incentivar a introdução dos conceitos da ciência da computação e estimular a prática do pensamento computacional com estudantes do ensino médio (SCAICO et al., 2013).

Nesta perspectiva, Rogério e colaboradores (2024) pontuam que a evolução do uso crescente de máquinas digitais está transformando o ensino e a aprendizagem, oferecendo aos professores novas maneiras de ensinar e aos alunos novas formas de aprender. Tal fato resulta em uma maior exigência sobre os alunos, que se tornam mais responsáveis por seu próprio aprendizado, enquanto os professores deixam de ser apenas transmissores de conhecimento e adotam metodologias menos mecânicas. Esse novo cenário fomenta o desenvolvimento do pensamento crítico, criatividade, flexibilidade e habilidades de pesquisa e resolução de problemas nos alunos. A educação está se adaptando à revolução tecnológica, que inclui a linguagem

computacional, inteligência artificial e a internet das coisas, promovendo uma aprendizagem mais prática e dinâmica. Com essas novas abordagens, os professores assumem o papel de facilitadores e mediadores, promovendo engajamento, criando trilhas de aprendizagem que conectam o ambiente digital ao físico e incentivando a colaboração, a resolução de problemas e a criatividade (VALENTE; SARAIVA; PESSOA, 2024).

Nesse contexto, diversas propostas têm surgido com a intenção de introduzir no currículo do ensino fundamental e médio os conceitos fundamentais da computação. Esses conceitos abrangem a compreensão da lógica de programação, a elaboração de algoritmos bem estruturados e a capacidade de abstração necessária para resolver problemas computacionais, conforme já antes tratado por Fernandes e Menezes (2011).

Winslow (1996) já ressaltava que os alunos que começam a estudar linguagem de programação na escola, podem até apreender a sintaxe (estrutura de um programa escrito em código) e a semântica (análise do significado) das declarações da linguagem, porém a maior dificuldade é combinar cada linha de código de forma a gerar um programa funcional. Uma outra pesquisa realizada por Ambrósio (2011), apontou que os alunos apresentam uma dificuldade de sair de um pensamento global utilizado na realidade, e ir para um pensamento digital, em suma, o aluno precisa aprender a pensar que nem um computador para aprender a programar.

A ideia de incorporar jogos digitais no ensino de programação pode ser direcionada ao nível do ensino médio. Especialistas em tecnologia, educadores e engenheiros estão advogando pela inclusão do ensino de programação nas escolas como uma maneira de compreender a base de todas as tecnologias disponíveis atualmente e de contextualizar o conhecimento adquirido na escola. O desafio reside na disseminação dessa cultura em todo o país, o que implica na necessidade de capacitar os professores e oferecer uma infraestrutura adequada para as aulas (MEDEIROS; SILVA; ARANHA, 2013).

Sendo assim, o presente estudo tem como objetivo introduzir os alunos no mundo da programação por meio da ferramenta *Scratch*, com o intuito de desenvolver o pensamento computacional, que inclui habilidades como a resolução de problemas e a criação de algoritmos. Os objetivos específicos são: demonstrar por meio de aulas em laboratório de informática, como programar com a ferramenta *Scratch* e introduzir a linguagem de programação *Javascript*, avaliar os ganhos no processo de ensino/aprendizagem em pensamento computacional através do uso do *Scratch*.

2 REFERENCIAL TEÓRICO

A ferramenta de estudo *Scratch* tem se destacado como uma solução inovadora para o ensino de programação, especialmente no contexto do "aprender fazendo" (*Learning by Doing*), que enfatiza a prática ativa e a experimentação como métodos de aprendizado. O *Scratch*, com sua interface visual e intuitiva, permite aos alunos desenvolverem habilidades de programação ao criar projetos interativos e resolver problemas de forma prática e lúdica. Esse ambiente de aprendizagem está alinhado com os princípios do Pensamento Computacional, que envolve habilidades como decomposição de problemas, reconhecimento de padrões e elaboração de algoritmos. Ao proporcionar uma plataforma onde os alunos podem aplicar diretamente os conceitos teóricos, o *Scratch* facilita a internalização do Pensamento Computacional, promovendo uma compreensão mais profunda e aplicada dos fundamentos da programação. Assim, o uso do *Scratch* vem a engajar os alunos de maneira criativa, reforçando a importância de métodos práticos para a construção de competências essenciais no campo da informática.

2.1 A Ferramenta de Estudo: *Scratch*

Os jogos educativos têm se mostrado eficazes na aplicação prática dos conceitos de programação, tornando o aprendizado mais envolvente e permitindo que os alunos experimentem os resultados de suas programações, o que reforça o aprendizado e torna o processo mais interessante. A inclusão de projetos práticos no currículo de programação também traz benefícios significativos, pois os alunos não

apenas aprendem a programar, mas também desenvolvem habilidades adicionais, como design e usabilidade, que são úteis em diversos contextos. Trabalhar em projetos com aplicação real ajuda os alunos a entender como a programação pode resolver problemas concretos. Além dos ganhos cognitivos, o ensino de programação contribui para o desenvolvimento de habilidades sociais e colaborativas, uma vez que atividades de programação frequentemente envolvem trabalho em grupo, promovendo a colaboração e o trabalho em equipe (MARTINS; REIS; MARQUES, 2016).

A inclusão da programação no ensino fundamental é impulsionada por vários fatores. Em primeiro lugar, reconhece-se que a programação desenvolve habilidades essenciais para o século XXI, como resolução de problemas, pensamento lógico e criatividade. A utilização de ferramentas como o *Scratch* oferece um ambiente interativo que facilita a compreensão de conceitos básicos de programação pelas crianças, permitindo que elas se familiarizem com a lógica computacional de maneira lúdica e intuitiva (SILVA; SOUZA; SILVA, 2016).

Rogério e colaboradores (2024) também colaboram ao pontuar que o *Scratch* é uma ferramenta desenvolvida pelo *Lifelong Kindergarten Group no MIT Media Lab* e atualmente está disponível em várias versões, com algumas delas oferecidas como *software* gratuito (Disponível em: <http://scratch.mit.edu>). Ele permite que os usuários criem animações e códigos utilizando um sistema baseado em blocos, que funciona de forma semelhante a um quebra-cabeça (**Figura 1**). Esse método facilita o aprendizado na criação de animações, jogos, histórias e interatividade, além de ajudar na resolução de problemas.

Figura 1. Tela tradicional do Scratch.

participantes. Igualmente, o *Scratch* busca promover a equidade na área da computação, tornando-a acessível a um amplo espectro de pessoas (SCRATCH, 2023).

O uso do *Scratch* como plataforma de ensino oferece uma abordagem altamente prática e interativa para os adolescentes no ensino médio. Essa abordagem vai além do tradicional aprendizado passivo, onde os alunos apenas absorvem informações teóricas. A utilização do *Scratch*, permite que os alunos se tornem os próprios protagonistas do processo de aprendizagem, sendo encorajados a criar projetos tangíveis, como jogos, animações e histórias (PASQUAL JÚNIOR, 2018).

Essa experiência prática não apenas os mantém engajados, mas também lhes proporciona uma sensação de realização à medida que veem suas ideias transformando-se em realidade através da programação. A interface de blocos e a capacidade de arrastar e soltar tornam a criação de programas acessível mesmo para iniciantes. Isso permite que os adolescentes experimentem livremente e testem suas ideias, observando instantaneamente os resultados de suas ações (SCRATCH, 2023).

A pesquisa pode contribuir demonstrando como os alunos podem aprender programação de maneira divertida com o *Scratch*, mostrando para o aluno, que por meio da linguagem de blocos (utilização de pequenos blocos de códigos), ele conseguirá assimilar o funcionamento da lógica de programação, ou seja, apreender a pensar como um computador. Tendo este conceito em mente, ficará mais fácil em compreender o funcionamento de futuras linguagens de programação utilizadas em mercado de trabalho.

2.2 Aprender Fazendo (*Learning by Doing*)

A aprendizagem através do fazer (*Learning by Doing*) implica em aprender com as experiências resultantes das próprias ações, em oposição a aprender apenas observando outras pessoas, lendo suas instruções ou ouvindo suas palestras. Embora observar, ler e ouvir sejam ações, eles não se encaixam na categoria de aprendizagem através do fazer, pois proporcionam experiências diretas com

demonstrações ou descrições de ações, em vez de experiências com ações que o aprendiz realiza de fato (REESE, 2011).

Nas práticas educativas tradicionais, primeiro aprende-se e depois faz-se, mas os saberes que a formação transmite, por vezes, não têm muito a ver com aquilo que as práticas exigem, donde a alternativa "aprender fazendo" que não deixa de ser a estratégia mais utilizada na generalidade das aprendizagens (SANCHES, 2011). Assim, qualquer bom processo de orientação vocacional baseado neste método, principalmente na área de design, deverá traduzir as diferentes possibilidades de atuação no mercado de trabalho e os diferentes níveis de conhecimento que o futuro profissional deverá possuir, levando-se em conta, obviamente, o contexto sócio-econômico-cultural envolvido (LIMA, 2002).

Na Educação Pré-Escolar, Cruz e Torres (2024) enfatizam que a robótica tem se destacado como uma ferramenta inovadora que promove o desenvolvimento infantil por meio da interação social. Ela expõe as crianças a situações-problema, ajudando no desenvolvimento de competências como pensamento crítico e resolução de problemas, além de estimular a criatividade e a imaginação. A construção e programação de robôs oferecem uma maneira divertida de introduzir a tecnologia e desenvolver habilidades sociais, como trabalho em equipe e comunicação. Dessa forma, a robótica não só facilita o aprendizado de forma interativa, mas também incentiva o entusiasmo pela aprendizagem.

Azevêdo (2018) relata que a experiência com lógica, programação e robótica educacional abriu novas formas de percepção, argumentação e criticidade. O estudo do autor, com alunos do 6º ano do ensino fundamental II, surgiu do interesse em explorar os benefícios que essas áreas podem oferecer ao aprendizado das crianças, promovendo seu desenvolvimento cognitivo e colocando-as como protagonistas de seus projetos. A pesquisa destacou a importância de tornar a educação mais atraente, incentivando a interação com o objeto de aprendizagem e o aprendizado prático, o que contribuiu para o aprimoramento dos conhecimentos lógicos.

Um modelo amplamente reconhecido no contexto de "aprender fazendo" é o modelo 70/20/10 (**Figura 2**), desenvolvido na metade dos anos 1990 através de pesquisas realizadas por Morgan McCall, Robert W. Eichinger e Michael M. Lombardo, do *Center for Creative Leadership* na Carolina do Norte, EUA (MARIOTTI; ZAHY, 2020).

Figura 2. Modelo de "aprender fazendo".



Fonte: adaptado de Lêus (2020)

Este modelo de aprendizagem integra a prática com a teoria, enfatizando que, além do aprendizado formal, a aplicação prática e o aprendizado social são essenciais. A metodologia 70/20/10 defende que a combinação desses elementos resulta em um processo de aprendizado mais eficaz, destacando que "aprender fazendo" é uma abordagem particularmente poderosa para alcançar melhores resultados educacionais (MARIOTTI; ZAHY, 2020).

A programação, como demonstrado em estudos anteriores, pode transformar a aprendizagem, facilitando a realização de atividades interdisciplinares e revelando habilidades e talentos dos participantes, sendo o *Scratch* uma ferramenta que permitiu o desenvolvimento de atividades interdisciplinares e exteriorizou as habilidades e os talentos. O "Aprender Fazendo" tem como alicerces (OLIVEIRA, 2019; CORRÊA JÚNIOR; VOSGERAU; MARTINS, 2020):

- **Relevância:** Aprender fazendo permite que os estudantes compreendam melhor a relação entre teoria e prática, esclarecendo o propósito e a aplicação dos conceitos

aprendidos. Ao aplicar teorias na prática, o aprendizado se torna mais significativo e as experiências prepararam melhor os jovens para o mundo real. Isso resulta em maior desenvoltura para enfrentar desafios e resolver problemas, pois a prática ajuda a consolidar o conhecimento e a torná-lo mais aplicável em situações concretas.

- **Errar Faz Bem ao Aprender Fazendo:** O método de "aprender fazendo" é uma abordagem inovadora que valoriza os erros como parte essencial do processo de aprendizagem. A experiência mostra que a curva de aprendizagem muitas vezes é mais acentuada quando se aprende com os erros do que com os acertos. Por isso, o papel dos professores como mediadores e orientadores é crucial para guiar os alunos através desses erros e ajudá-los a adquirir conhecimento valioso para evitar falhas futuras. O processo de tentativa e erro promove a criatividade e a busca por alternativas inovadoras, e a escola deve ser um espaço onde os alunos possam testar, errar, aprender e evoluir.
- **Interdisciplinaridade:** A possibilidade de transitar por diferentes áreas do conhecimento é uma grande vantagem do processo pedagógico baseado na prática. Abordar e resolver problemas conectando várias disciplinas resulta em aulas mais dinâmicas e experiências de aprendizado mais estimulantes. A integração de conhecimentos transdisciplinares permite uma absorção e fixação mais eficaz dos conteúdos, proporcionando uma vivência prática que torna o aprendizado mais relevante e aplicado.
- **Cooperação e Colaboração:** O futuro valoriza o trabalho em equipe, tornando as habilidades de cooperação e colaboração essenciais tanto no presente quanto no futuro. O conhecimento é mais valioso quando é compartilhado, e aprender fazendo envolve transformar fracassos em sucessos por meio da atuação conjunta das habilidades de um grupo. A empatia e a capacidade de discutir soluções com os colegas são desafios constantes, e a vivência coletiva contribui para o desenvolvimento da autonomia e independência de cada indivíduo. O equilíbrio entre o desenvolvimento pessoal e o espírito de coletividade, respeito às diferenças e diversidade é fundamental para o sucesso no ambiente educacional e profissional.
- **Autoestima e Confiança:** Desenvolver um projeto do início ao fim tem um impacto significativo na autoestima e confiança dos indivíduos. Tanto docentes quanto educandos experimentam um aumento nessas áreas ao aprender fazendo. A dificuldade em mensurar o aumento das capacidades analítica, crítica, argumentativa, e das habilidades de gestão de recursos e pessoas reflete a complexidade desses processos. A flexibilidade para explorar o desconhecido e a confiança demonstrada em situações de exposição e pressão trazem benefícios duradouros para a vida dos educandos. Em resumo, essas capacidades e atitudes, embora difíceis de medir, são propósitos importantes do processo de ensino baseado na prática.

Na visão dos autores, a metodologia "Aprender Fazendo" tem se mostrado uma abordagem transformadora nas aulas de informática e computação no ensino médio, proporcionando uma experiência educacional rica e envolvente. Ao integrar teoria com prática, essa abordagem facilita a compreensão dos conceitos técnicos e prepara os alunos para enfrentar desafios reais com maior confiança e criatividade.

O envolvimento ativo dos estudantes em projetos práticos promove o desenvolvimento de habilidades críticas, como resolução de problemas e trabalho em equipe, ao mesmo tempo que fomenta a autonomia e a capacidade de inovação. Ao adotar essa metodologia, os educadores conseguem criar um ambiente de aprendizagem dinâmico que valoriza a experimentação e o aprendizado contínuo, preparando os alunos para uma futura carreira no campo da tecnologia e para uma participação mais efetiva em um mundo cada vez mais digital.

2.3 Pensamento Computacional

No Brasil, há uma carência de referenciais teóricos sobre a implementação de projetos educacionais que introduzam novas práticas pedagógicas no ensino de Ciências no Ensino Fundamental das escolas públicas, especialmente aqueles que envolvem programação e o desenvolvimento do Pensamento Computacional. Isso torna a elaboração de estudos sobre esse tema particularmente relevante. Observa-se que, no ambiente escolar, há uma percepção de que esses elementos devem ser abordados exclusivamente nas disciplinas de Exatas (CERATI, 2024).

Os smartphones, computadores e calculadoras não apenas aumentam o uso das tecnologias existentes, mas também impulsionam a necessidade de desenvolver novas. Nesse cenário, enquanto a maioria das pessoas usa a tecnologia, é crucial preparar aqueles que irão criar os recursos tecnológicos do futuro, muitos dos quais ainda são desconhecidos.

Para isso, é fundamental cultivar habilidades de pensamento crítico e computacional, permitindo que as pessoas não apenas utilizem as tecnologias digitais, mas também criem soluções inovadoras além do uso tradicional dessas ferramentas.

O pensamento crítico, conforme discutido por especialistas, é ativado por atividades proporcionadas tanto por computadores quanto por educadores, e envolve um alto nível de engajamento cognitivo. A interação com computadores pode reforçar tanto as habilidades dos alunos quanto as potencialidades da tecnologia, resultando em uma aprendizagem mais significativa do que a que seria possível apenas com a interação isolada entre aluno e tecnologia (GRAÇA; RAMOS; SOLÉ, 2024).

O uso de computadores facilita a aprendizagem por meio de uma combinação de interações entre o aluno, o software ou aplicativo e o conteúdo. Recursos variados, como simulações, jogos educativos, acesso a informações em tempo real e ferramentas de colaboração, estimulam diferentes aspectos cognitivos dos estudantes. A interatividade e o feedback imediato permitem que os alunos experimentem, cometam erros e aprendam de maneira dinâmica, além de desenvolver habilidades importantes como pensamento crítico, resolução de problemas, colaboração e criatividade (DOBGENSKI; PRADO; SILVA, 2024).

Nesse cenário, é evidente a importância de incentivar os estudantes a desenvolver um pensamento significativo. O Pensamento Computacional surge como uma excelente alternativa, pois é um tipo de pensamento lógico e racional, característico dos profissionais da Ciência da Computação, que busca estratégias eficazes para resolver problemas com base nos recursos computacionais disponíveis.

No Brasil, o Pensamento Computacional está oficialmente integrado ao ensino de Matemática, conforme indicado pela Base Nacional Comum Curricular (BNCC, 2018). Esta base define as aprendizagens essenciais que os estudantes devem adquirir ao longo de sua trajetória acadêmica. Levantamentos bibliográficos sobre pesquisas que abordam simultaneamente o Pensamento Computacional e o ensino de Matemática mostram que, apesar de ser uma área ainda recente no país, o número de trabalhos publicados tem aumentado e abrange contextos variados. Esses estudos incluem o desenvolvimento do Pensamento Computacional em estudantes dos Ensinos Fundamental, Médio e Superior, bem como em professores em formação inicial e continuada. As estratégias mais comuns utilizadas incluem o ensino de

programação de computadores, robótica, resolução de problemas e simulação. Entre as ferramentas de programação, o *Scratch* é uma das mais frequentemente empregadas (CERATI, 2024).

Em 17 de fevereiro de 2022, a Câmara da Educação Básica do Conselho Nacional de Educação aprovou as "Normas sobre Computação na Educação Básica – Complemento à BNCC" (Resolução CEB 01/2022). A aprovação foi unânime e o documento foi homologado pelo Ministério da Educação e publicado no Diário Oficial da União em 3 de outubro de 2022. Agora, cabe aos estados, municípios e ao Distrito Federal iniciar a implementação dessas diretrizes. De acordo com as normas, o Ministério da Educação será responsável por definir políticas para a formação de professores, apoiar o desenvolvimento de currículos e recursos didáticos compatíveis com as competências e habilidades necessárias. Além disso, o ministério também deverá estabelecer a política de avaliação para o ensino de computação na educação básica e fornecer orientação aos sistemas e redes de ensino para garantir a implementação e continuidade do ensino (BRASIL, 2022).

O documento enfatiza que a Computação na Educação Básica deve ser vista como um pilar estratégico para a transformação social e cultural no Brasil, visando melhorar a qualidade de vida da população. O objetivo é apoiar o desenvolvimento de cidadãos críticos, criativos, inovadores e empreendedores, que possam resolver problemas e reconhecer o potencial da Computação para beneficiar a sociedade em níveis social, econômico e científico. A elaboração do documento contou com a participação de diversos setores da sociedade, incluindo Institutos Federais, gestores públicos, professores da Educação Básica, especialistas em Computação e membros da SBC.

Cerati (2024) explica que a resolução define as competências e premissas específicas a serem desenvolvidas em cada etapa da Educação Básica e está organizada, assim como o documento de diretrizes desenvolvido pela SBC, pelos eixos de Pensamento Computacional, Cultura Digital e Mundo Digital. A **Figura 3** traz uma representação deste pensamento:

Os anos iniciais do ensino devem introduzir conceitos que gradualmente ajudam os alunos a compreender estruturas abstratas utilizadas para interação e manipulação de dados, informações e resolução de problemas. As práticas recomendadas incluem o uso frequente de artefatos digitais e computadores, atividades lúdicas, computação desplugada e construção de jogos. A expectativa é que o desenvolvimento gradual e consistente permita o entendimento básico de algoritmos e manipulação de dados com diferentes linguagens, incluindo as visuais. A compreensão técnica da construção de algoritmos e a decomposição de problemas são esperadas para ocorrer entre os Anos Finais do Ensino Fundamental e o Ensino Médio. Nos Anos Iniciais, é essencial que experiências concretas ajudem na construção de modelos mentais para as abstrações computacionais que serão formalizadas nos anos seguintes, especialmente com linguagens de programação (OLIVEIRA; CAMBRAIA, HINTERHOLZ, 2021).

Figura 3. Computação na educação básica.



Fonte: adaptado de Cerati (2024).

Nesta pesquisa, adota-se as definições de Pensamento Computacional que consideram este conceito como um processo de resolução de problemas, design de sistemas e compreensão do comportamento humano, baseado em conceitos fundamentais da Ciência da Computação. O Pensamento Computacional é visto como um processo cognitivo voltado para a resolução de problemas, demonstrando habilidades para pensar de forma abstrata e algorítmica, decompor atividades complexas e realizar avaliações e generalizações. Essa definição enfatiza os aspectos cognitivos essenciais para o desenvolvimento do PC e complementa a visão mais ampla do conceito (DOBGENSKI; PRADO; SILVA, 2024).

O Pensamento Computacional é frequentemente descrito em termos de resolução de problemas, que pode ou não envolver programação. Ao considerar a

resolução de problemas com o uso de meios computacionais, é essencial que o estudante compreenda primeiro o problema por meio da abstração, ou seja, entenda profundamente o problema para selecionar os dados essenciais, reduzi-lo ou aproximá-lo. Em seguida, deve ser capaz de programar a solução, o que inclui a coleta e análise de dados, a representação deles, a decomposição do problema, a proposição de algoritmos, e a realização de automação, paralelização e simulação, conforme necessário. Esses processos mentais são cruciais para o desenvolvimento do Pensamento Computacional nos estudantes (VALENTE, 2016).

De acordo com a teoria de Piaget (1969), a inteligência é organizada em estruturas mentais, chamadas esquemas, e é por meio dessas estruturas que as pessoas assimilam o mundo ao seu redor. A interação com o ambiente e a manipulação de objetos permitem ao indivíduo abstrair características das suas ações e expandir suas estruturas mentais, tornando-as mais complexas. Compreender esses processos mentais é fundamental para desenvolver o Pensamento Computacional.

Piaget sugeriu que, em vez de se adaptar imediatamente às novas realidades, o indivíduo deve incorporar os dados de sua própria atividade de forma laboriosa. Isso leva a refletir sobre como “gerar” um banco de dados pessoal e como incorporar novos dados e informações para expandir o conhecimento (PIAGET, 1969). No contexto do Pensamento Computacional, é importante oferecer aos alunos diferentes meios para resolver problemas e apresentar diversas formas de pensar e visualizar esses problemas. Expor os estudantes a uma variedade de situações e problemas contribuirá para que eles desenvolvam estratégias e experiências que serão úteis no futuro.

Ao enfrentar a resolução de um problema específico, podemos nos perguntar: "Quão difícil é resolver este problema?" e "Qual é a melhor maneira de resolvê-lo?" A ciência da computação tem uma base teórica sólida para responder a essas perguntas de forma precisa. Para determinar a dificuldade de um problema, precisamos considerar o poder subjacente da máquina, o dispositivo de computação que

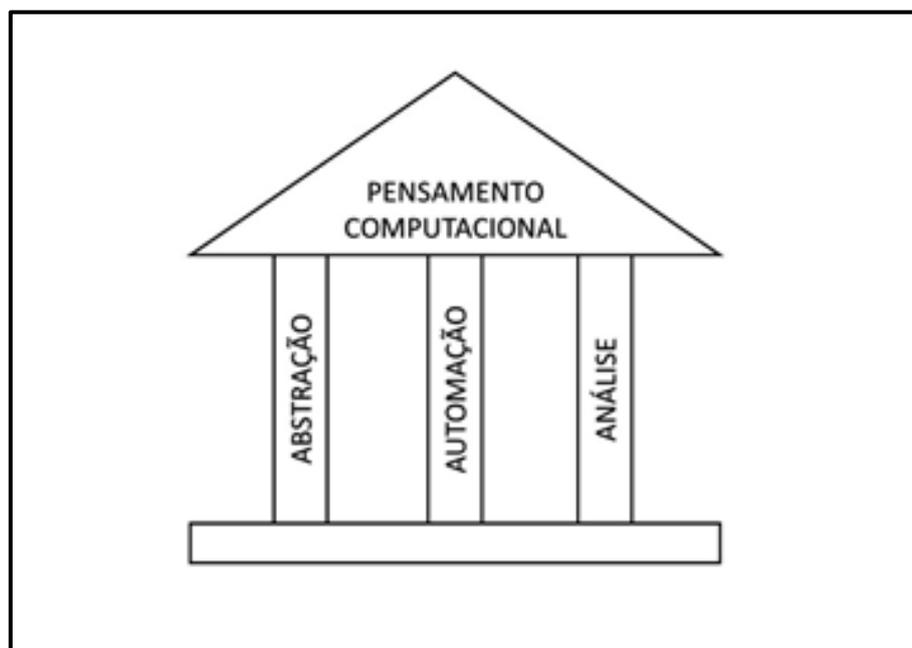
executará a solução. Isso inclui levar em conta o conjunto de instruções da máquina, as limitações de seus recursos e o ambiente no qual ela opera (WING, 2021).

O pensamento computacional é o processo de identificar elementos da computação no mundo ao nosso redor e aplicar ferramentas e técnicas para compreender e raciocinar sobre sistemas e processos naturais, sociais e artificiais. No contexto educacional, o Pensamento Computacional capacita os alunos a resolver problemas por meio da decomposição em partes menores, a abstrair o que não é essencial para a conclusão de uma tarefa, a reconhecer padrões que se repetem em diferentes problemas e a criar algoritmos para resolvê-los (WING, 2006 apud SOUZA, 2021).

Nos últimos anos, o conceito tem recebido crescente atenção por parte de educadores em todo o mundo, que defendem sua inclusão em todos os currículos escolares. No entanto, ainda existe um considerável grau de confusão e incerteza em relação ao que realmente deve ser incorporado e por quê. Em termos simples, o pensamento computacional está relacionado à tentativa de compreender uma das maneiras pelas quais os seres humanos abordam a resolução de problemas, ou seja, o próprio processo de resolução em si.

Ele possui três pilares (**Figura 4**), o pilar de abstração envolve a criação de uma solução para um problema por meio de algoritmos, que são descrições de processos expressas em diferentes formas de linguagem, como oral, textual ou visual, a automação refere-se à mecanização das soluções, permitindo que as máquinas auxiliem na resolução de problemas e a análise envolve a avaliação da viabilidade de encontrar uma solução computacional para um problema ou a avaliação da adequação de uma solução existente (FERREIRA, 2020).

Figura 4. Pilares do Pensamento Computacional



Fonte: adaptado de Ferreira (2020).

Na grande parte de nossas escolas, o que ocorre é uma forma de "treinamento digital", muitas vezes com custos elevados. O que é ainda mais preocupante é que estamos ensinando aos nossos alunos que a tecnologia serve apenas para reorganizar informações já existentes, em vez de capacitá-los a criar conhecimento genuinamente novo. O conhecimento verdadeiramente inovador não se encontra simplesmente na internet, pronto para ser acessado com algumas palavras-chave em um mecanismo de busca (FRANÇA et al., 2014).

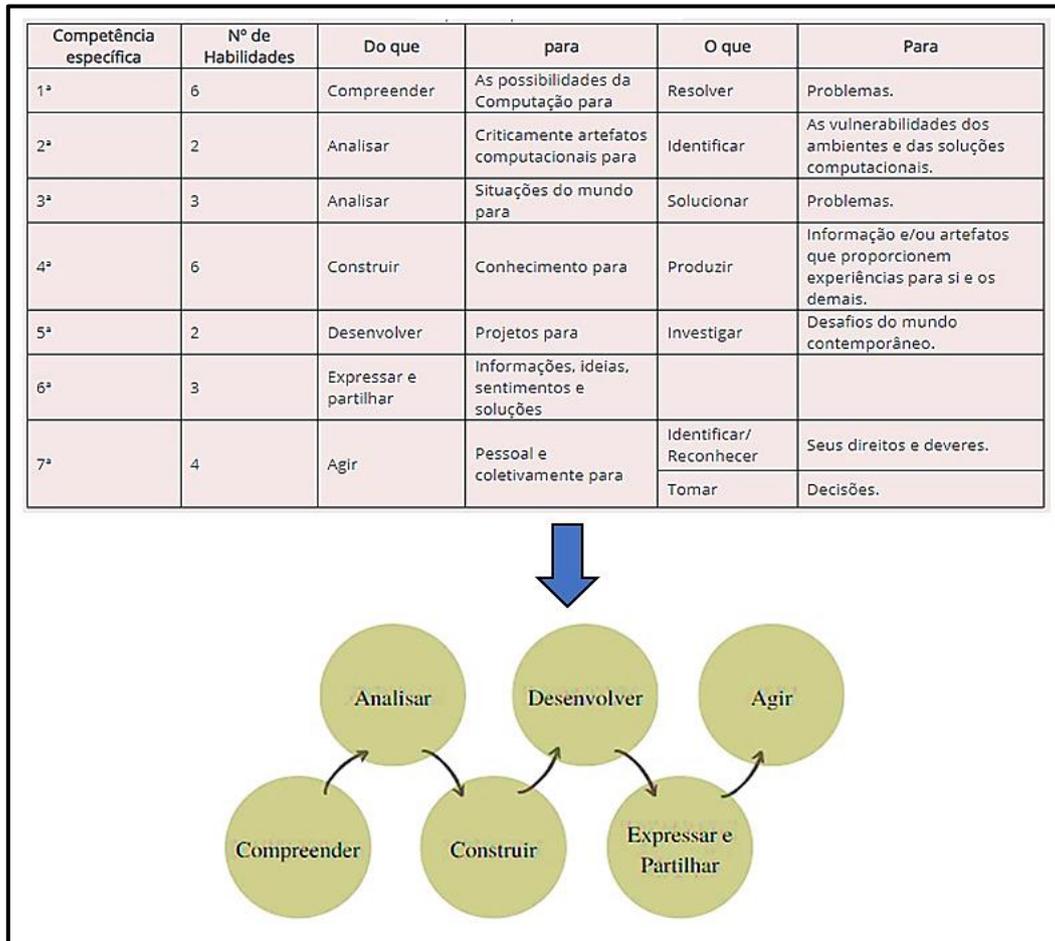
Para que os conceitos fundamentais do pensamento computacional sejam compreendidos de forma eficaz, a SBC (Sociedade Brasileira de Computação) propõe uma abordagem que parte do concreto para o abstrato. Isso significa que, inicialmente, esses conceitos devem ser explorados a partir de situações do dia a dia e usando materiais tangíveis. Posteriormente, as formalizações e abstrações correspondentes são introduzidas. Essa abordagem permite que os estudantes, ao trabalharem com materiais concretos, construam modelos mentais que, em um

estágio posterior, servirão como base para que eles possam abstrair e formalizar seu conhecimento (RIBEIRO; FOSS; CARVALHEIRO, 2019).

O pensamento computacional abrange a habilidade de identificar um problema complexo e dividi-lo em componentes menores, tornando-o mais acessível para resolução. Cada um desses componentes deve ser analisado minuciosamente, priorizando os detalhes cruciais e descartando informações irrelevantes. Busca-se também identificar problemas semelhantes que já foram previamente solucionados, a fim de aproveitar soluções existentes. Por fim, é estabelecida uma sequência de atividades ou etapas para resolver cada um dos componentes identificados do problema (BRACKMANN, 2017 apud FARIAS, 2019).

O processo de aprendizado de programação é desafiador, uma vez que os conceitos abordados e a quantidade de conhecimento envolvido não são simples e de fácil compreensão, especialmente para estudantes iniciantes. Diante disso, é fundamental adotar novas metodologias e estratégias de ensino para proporcionar uma experiência de ensino e aprendizado de programação mais sólida e envolvente para os alunos. Isso inclui a integração da programação na educação básica, com o objetivo de permitir que os alunos tenham acesso aos conteúdos de computação e vejam o computador como uma ferramenta para desenvolver sua criatividade, pensamento lógico e habilidades de resolução de problemas (ARIMOTO; OLIVEIRA, 2019). Na **Figura 5**, estão apresentadas as competências específicas, o número de habilidades associadas a cada uma e os verbos organizados para refletir o que deve ser feito e para onde se deve direcionar:

Figura 5. Competências específicas e seus verbos.



Fonte: adaptado de Brasil (2022).

Deliberalli, Kluber e Boscarioli (2023) retomam que no Ensino Médio, as competências definidas na normativa estão divididas em sete áreas, cada uma abrangendo vinte e seis habilidades com descrições e exemplos correspondentes. Embora essas competências possam ser vistas como limitadas, elas ajudam leitores, professores e formadores a entender claramente os objetivos de cada habilidade. Cada uma das sete competências específicas começa com verbos que indicam as ações a serem realizadas e a direção a ser seguida (BRASIL, 2022). Não significa, necessariamente, que precisem seguir essa ordem, porém esse é ordenamento

explicitado nas competências específicas. Diante disso, deve-se buscar compreender os significados dicionarizados dos verbos apresentados na **Figura 5**, bem como se apresentam na BNCC, no Referencial Curricular para o Ensino Médio e nas normas sobre Computação para a Educação Básica (nessa ordem), para que se avance no pensamento do que se quer com as competências apresentadas em cada um dos documentos. É importante destacar que os documentos servem para alargar a compreensão da temporalidade e da tessitura daquilo que se busca esclarecer neste texto, ou seja, a formação docente. Lógica de programação é a técnica de encadear pensamentos para atingir determinado objetivo. O aprendizado desta técnica é necessário, para quem deseja trabalhar com desenvolvimento de sistemas e programas e algoritmo é uma sequência de passos finitos com o objetivo de solucionar um problema (LOPES; GUTO, 2002).

O ritmo de aprendizagem, é fundamental destacar que o andamento da disciplina nem sempre está alinhado com a velocidade de assimilação de cada aluno. Esse descompasso pode resultar em desafios na compreensão dos conceitos fundamentais, levando ao desinteresse pelo conteúdo ministrado ou, em casos extremos, a uma aversão à disciplina. É comum observar pesquisas que identificam casos de evasão de cursos, um fenômeno intimamente relacionado às dificuldades de aprendizagem (CASTRO, 2003).

A gamificação pode ser empregada para enriquecer o processo de aprendizagem, transformando um conteúdo geralmente apresentado em um ambiente de ensino tradicional, seja presencial ou a distância, ao incorporar elementos de jogos. Isso cria uma oportunidade de aprendizado no formato de um jogo. No entanto, para que a gamificação seja eficaz em adicionar um nível adicional de interesse à educação, é crucial que os elementos do jogo estejam integrados de forma coesa (KAPP, 2012).

Scaico e colaboradores (2013) relatam uma olimpíada de programação com a ferramenta *Scratch*, com o objetivo de motivar os alunos de ensino médio da rede pública a programar, por meio de uma abordagem de ensino que estimula a

criatividade e o poder de exploração dos jovens. Notaram que praticamente todos os alunos alcançaram resultados semelhantes em todas as questões, o que realmente valida a abordagem pedagógica adotada. Esse sucesso pode ser atribuído ao desenvolvimento de sólidas habilidades de programação, incluindo a capacidade de ler e entender código de outros, depurar e corrigir erros, bem como implementar algoritmos simples, que foram enfatizados durante as aulas.

Castro (2017), também relata em seu trabalho sobre a inserção de programação com *Scratch* para crianças dos anos iniciais do ensino fundamental de uma escola municipal. A pesquisa tratou de realizar pequenas oficinas baseadas em duas importantes correntes pedagógicas: o construtivismo de Jean Piaget e o construcionismo de Seymour Papert. Essa abordagem pedagógica foi projetada de forma a estimular a motivação dos alunos, uma vez que os roteiros das aulas foram cuidadosamente elaborados para desafiá-los a criar suas próprias estratégias, isto não apenas promoveu o desenvolvimento do raciocínio lógico dos estudantes, mas também os incentivou a se envolverem ativamente no processo de aprendizagem.

Na experiência de Oliveira; Rodrigues; Queiroga, (2016), os quais realizaram uma pesquisa que abordou o alto índice na quantidade de reprovações e evasões em cursos na área de computação, demonstrou a aplicação de alternativas para facilitar o aprendizado do aluno que visam reduzir estes números por meio do uso do *Scratch*. Quanto às avaliações das turmas em questão, é notável que alcançaram uma taxa de aprovação impressionante de 92% no ensino médio e 85% no ensino superior. Esse aumento de 5% em relação às médias de aprovação dos anos anteriores é um indicativo sólido de melhoria. Essa conquista não apenas reflete um sucesso quantitativo, mas também se traduz em maior satisfação dos alunos, confiança, criatividade e liberdade de expressão.

3 METODOLOGIA

O estudo baseia-se em uma abordagem de pesquisa qualitativa, quanto ao procedimento desenvolveu-se uma pesquisa-ação e bibliográfica, sendo o estudo de

caso um processo de análise da ferramenta *Scratch*. Esta pesquisa caracteriza-se como qualitativa, a qual é compreendida por Carr (2005) como aquela que tem uma disposição para examinar princípios implícitos na prática de pesquisa. Preocupa-se com aspectos da realidade que não podem ser quantificados, centrando-se na compreensão e explicação da dinâmica das relações sociais. A pesquisa qualitativa trabalha com o universo de significados, motivos, aspirações, crenças, valores e atitudes, o que corresponde a um espaço mais profundo das relações, dos processos e dos fenômenos que não podem ser reduzidos à operacionalização de variáveis (MINAYO, 2001).

A perspectiva adotada é da pesquisa-ação, que segundo Baldissera (2001) pressupõe uma relação colaborativa entre os pesquisadores e pessoas envolvidas no estudo da realidade. Para que se alcance o objetivo proposto na pesquisa-ação, no sentido de estabelecer uma relação entre conhecimento e ação, torna-se necessário:

[...] uma ampla e explícita interação entre os pesquisadores e envolvidos na pesquisa e que esta não se limita a uma forma de ação (risco de ativismo), mas pretende aumentar o conhecimento dos pesquisadores e o conhecimento ou nível de consciência das pessoas e grupos que participarem do processo, bem como, contribuir para a discussão ou fazer avançar o debate acerca das questões abordadas (BALDISSERA, 2001, p. 06).

Também se utilizou de uma pesquisa bibliográfica, definida como:

[...] levantamento de referências teóricas já analisadas e publicadas por meios escritos e eletrônicos, como livros, artigos científicos, páginas de websites. Qualquer trabalho científico inicia-se com uma pesquisa bibliográfica, que permite ao pesquisador conhecer o que já se estudou sobre o assunto (Fonseca, 2002, p. 32).

Foram consultadas as bases de dados online livres, tais como Google Acadêmico, Portal Periódicos CAPES/MEC e *Google Books*, seguindo as orientações de Marconi e Lakatos (2003). Assim, inicialmente foi realizada uma busca sobre o ensino de programação com *Scratch*, tendo como objetivo identificar artigos científicos pertinentes sobre o tema. Na busca inicial foram considerados os títulos e os resumos dos artigos para a seleção ampla de prováveis trabalhos de interesse, sendo

destacados os resumos (dos artigos que não tinham texto acessível) e os textos completos dos artigos, utilizando-se como palavras-chave os termos *Scratch*, programação, ensino médio, alunos e programação em blocos. Os textos selecionados foram aqueles que permitiram alinhamento com o objetivo de estudo e sua situação-problema. Foram encontrados diversos artigos referentes aos temas, sendo selecionados aqueles com os conceitos relevantes ao trabalho, descrições e dados que nos permitiram discutir os resultados da ação, contribuindo para a reflexão sobre o projeto.

A respeito do estudo de caso, Yin (2001 apud VENTURA, 2007), retrata como uma pesquisa empírica que abarca um método completo, envolvendo o planejamento, a coleta e a análise de dados. Ele pode englobar tanto estudos de caso individuais quanto múltiplos, bem como diversas abordagens de pesquisa quantitativas e qualitativas.

Na visão da aplicação em uma pesquisa educacional, acaba permitindo a descrição e análise de uma unidade social, levando em consideração suas diversas dimensões e dinâmicas naturais. Quando enquadrados nas abordagens qualitativas e no contexto escolar, os estudos de caso que empregam técnicas para obter uma compreensão profunda, como a observação participante e entrevistas intensivas, possibilitam a reconstrução dos processos e das relações que moldam a experiência cotidiana na escola (ANDRÉ, 2013).

4 RESULTADOS E DISCUSSÃO

A aplicação do projeto não teve o intuito de aprimorar o conhecimento dos alunos na linguagem de programação *Javascript*, ou muito menos saberem o funcionamento de sua sintaxe, e sim, em melhorar o raciocínio lógico em programar. Tanto a avaliação em *Javascript* quanto em *Scratch* foram elencadas aos alunos com o mesmo intuito, no qual é aprimorar o raciocínio lógico. Embora a sintaxe das duas linguagens seja diferente, o propósito implícito era o mesmo e as perguntas foram as mesmas, somente adaptadas para cada ferramenta.

Antes da aplicação das avaliações, ambas as turmas tiveram 5 aulas de *Scratch* e de *Javascript* e a duração de cada aula foram de 1 hora, totalizando 50 horas/aula por turma. Nestas aulas, os alunos foram introduzidos aos conceitos básicos de cada ferramenta e às suas características específicas, juntamente com exercícios práticos, com o objetivo de prepará-los para realizar os testes.

Para evitar qualquer vício de linguagem e garantir que os alunos realmente compreendessem a lógica por trás da programação, a classe foi dividida em dois grupos. Cada grupo fez a avaliação em uma ordem inversa, conforme visto na **Figura 6**, em que a turma 1 realizou a avaliação em *Scratch* e a turma 2 realizou a avaliação em *Javascript* e, em outro dia, as avaliações foram invertidas, de forma, portanto, que a turma 1 ficou com *Javascript* e a turma 2 ficou com *Scratch*. Os alunos que já sabem como resolver o problema usando o *Scratch* não confiariam inteiramente nessas informações para a avaliação do *Javascript*.

Analisando os resultados obtidos na **Figura 6**, podemos concluir que tanto a turma 1 como a turma 2, se saíram muito bem na avaliação em *Scratch*. Apesar da turma 2 ter começado com *Javascript*, a linguagem não ajudou muito na avaliação em *Scratch*, pois a maioria dos alunos foram insatisfatório ou regulares no primeiro exame. A dificuldade dos alunos é com a ferramenta e não com o algoritmo em si. Já a turma 1, apesar de ter começado com *Scratch* e terem ido super bem na avaliação, metade da turma teve notas insatisfatórias ou regulares na avaliação de *Javascript*, mostrando que o *Scratch* é uma ferramenta muito fácil de aprender a utilizar para estimular o raciocínio lógico, pensamento computacional e algoritmos, porém não é capaz de melhorar os conhecimentos em lógica de programação.

Figura 6. Resultado das avaliações em *Scratch* e em *Javascript*

Antes da Avaliação do JavaScript			Após a Avaliação do JavaScript		
TURMA 1 - 17 alunos			TURMA 2 - 18 alunos		
Avaliação em Scratch			Avaliação em Scratch		
Menção	Nota	Qtde. de Alunos	Menção	Nota	Qtde. de Alunos
I (Insatisfatório)	0 até 3	0	I (Insatisfatório)	0 até 3	0
R (Regular)	4 até 6	5	R (Regular)	4 até 6	3
B (Bom)	7 até 9	6	B (Bom)	7 até 9	10
MB (Muito Bom)	10	6	MB (Muito Bom)	10	5
Após a Avaliação do Scratch			Antes da Avaliação do Scratch		
TURMA 1 - 17 alunos			TURMA 2 - 18 alunos		
Avaliação em JavaScript			Avaliação em JavaScript		
Menção	Nota	Qtde. de Alunos	Menção	Nota	Qtde. de Alunos
I (Insatisfatório)	0 até 3	5	I (Insatisfatório)	0 até 3	7
R (Regular)	4 até 6	2	R (Regular)	4 até 6	7
B (Bom)	7 até 9	7	B (Bom)	7 até 9	3
MB (Muito Bom)	10	3	MB (Muito Bom)	10	1

Fonte: obtido pelos autores.

Portanto, embora a turma 2 tenha iniciado sua jornada com *Javascript*, isso não os colocou significativamente à frente na avaliação por meio do *Scratch*, uma vez que a maioria dos alunos teve um desempenho ruim ou quase médio no primeiro exame. Esta falta de uniformidade implica que os desafios enfrentados pelos alunos têm mais a ver com o conhecimento da ferramenta do que com a compreensão dos algoritmos. Por outro lado, para a turma 1, apesar de um desempenho impressionante no *Scratch* no início, metade dos alunos ainda terminou com notas insatisfatórias ou regulares na avaliação de *Javascript*.

5 CONCLUSÃO

O estudo revela que, embora o *Scratch* se destaque por estabelecer conceitos de lógica de programação de maneira significativa, ele não consegue capacitar os alunos o suficiente para aplicar esses conceitos básicos em outras linguagens de programação complexas, como *Javascript*. Há necessidade mais estudos para embasamento destes resultados e de se potencializar mais ações desta natureza no espaço da sala de aula, que é o maior e melhor laboratório do processo ensino-aprendizagem para um docente.

REFERÊNCIAS

AMBRÓSIO, A. P. Programação de computadores: compreender as dificuldades de aprendizagem dos alunos. **Revista Galego-Portuguesa e Psicoloxía e Educación**, v. 19, n. 1, p. 185-197, 2011.

ANDRÉ, M. O que é um estudo de caso qualitativo em educação. **Revista da FAAEBA: Educação e Contemporaneidade**, v. 22, n. 40, p. 95-103, 2013.

ARIMOTO, M.; OLIVEIRA, W. Dificuldades no processo de aprendizagem de programação de computadores: um survey com estudantes de cursos da área de computação. In: **Anais do XXVII Workshop sobre Educação em Computação**. SBC, 2019. p. 244-254.

AZEVÊDO, E. M. S. **A fluência digital e a utilização da robótica educacional mediante a abordagem do aprender fazendo e do brincar com crianças**. 2018. 148 f. Dissertação (Mestrado) - Universidade Federal Rural do Semi-árido, Programa de Pós-graduação em Cognição, Tecnologias e Instituições. Mossoró, RN, 2018.

BALDISSERA, Aderlina. Pesquisa-ação: uma metodologia do “conhecer” e do “agir” coletivo. **Sociedade em Debate**, Pelotas, v. 07, n. 02, p. 05-25, 2001.

BRACKMANN, C. P. **Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica**. 226 f. Tese (Doutorado) - Programa de Pós-graduação em Informática na Educação, Centro de Estudos Interdisciplinares em Novas Tecnologias na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre. 2017.

BRASIL, **Base Nacional Comum Curricular: Ensino Médio**. Brasília, DF: MEC 2018. Disponível em: <http://basenacionalcomum.mec.gov.br/>. Acesso em: 15 de ago de 2022.

CERATI, M. D. P. Codificando saberes: oficinas de *Scratch* como estratégia para fomentar o pensamento computacional no ensino de ciências. **VI Encontro Internacional Sobre Jogos e Mobile Learning**. Universidade de Coimbra, p. 73-84, 2024.

CARR, Wilfred. **Quality in teaching: Arguments for a reflective profession**. Routledge, 2005.

CASTRO, A. de et al. **O uso da programação Scratch para o desenvolvimento de habilidades em crianças do ensino fundamental**. 2017. 124 f. Dissertação

(Mestrado em Ensino de Ciência e Tecnologia) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, PR, 2017.

CASTRO, T H. C. et al. Utilizando programação funcional em disciplinas introdutórias de computação. **Anais do WEI**, p. 1-12, 2003.

CORRÊA JÚNIOR, R.; VOSGERAU, D. S. R.; MARTINS, P. L. O. *Learning by doing* e complexidade. **Revista Teias**, v. 21, n. 63, p. 383-391, 2020.

CRUZ, S.; TORRES, A. Atividades STEAM e robótica na Educação Pré-Escolar. **Sensos-E**, v. 11, n. 2, p. 3–14, 2024.

DELIBERALLI, M; KLÜBER, T. E.; BOSCARIOLI, C. Um olhar para as normas sobre computação na educação básica e suas implicações para a formação de professores. **Revista Espaço do Currículo**, v. 16, n. 3, p. 1-16, 2023.

DOBGENSKI, J.; PRADO, M. E. B. B.; SILVA, A. S. G. Desenvolvimento do Pensamento Computacional com programação *Scratch*: um estudo comparativo entre aprendizado presencial e remoto. **VI Encontro Internacional Sobre Jogos e Mobile Learning**. Universidade de Coimbra, p. 73-84, 2024, p. 141-152.

FARIAS, C. et al. Estimulando o Pensamento Computacional: uma experiência com *ScratchJr*. In: **Anais do XXV Workshop de Informática na Escola**. SBC, 2019. p. 197-206.

FERNANDES, C. S.; MENEZES, P. B. Metodologia do Ensino de Ciência da Computação: Uma proposta para criança. **Anais do Workshop de Informática na Escola**. Fortaleza: CE. 2011.

FERREIRA, M. N. F. et al., 2020. Ensinando design de interface de usuário de aplicativos móveis no ensino fundamental. **Revista Brasileira de Informática na Educação**, [S. l.], v. 28, p. 48–72, 2020.

FONSECA, J. J. Souza. **Metodologia da pesquisa científica**. Fortaleza: UEC, 2002. Apostila.

FRANÇA, R. S. et al. A disseminação do pensamento computacional na educação básica: lições aprendidas com experiências de licenciandos em computação. **XXXIV Congresso da Sociedade Brasileira de Computação – CSBC**, 2014, p. 1472-1482.

GOMES, T. C. S.; MELO, J. C. B. O pensamento computacional no ensino médio: Uma abordagem *blended learning*. **Anais do XXI Workshop sobre Educação em Computação–XXXIII Congresso da Sociedade Brasileira de Computação**. Maceió, AL–Brasil. 2013.

GRAÇA, V.; RAMOS, A.; SOLÉ, G. O uso de plataformas digitais para o ensino e aprendizagem de História: um estudo com alunos do 1.º CEB e 2.º CEB. **VI Encontro Internacional Sobre Jogos e Mobile Learning**. Universidade de Coimbra, p. 73-84, 2024.

KAPP, K. M. **The gamification of learning and instruction: game-based methods and strategies for training and education**. John Wiley & Sons, 2012.

LIMA, J. A. A. **Descobrendo o Design no Ensino Médio** –Aprender Fazendo Discovering Design in High School–Learn by Doing. 2002.

LOPES, A.; GARCIA, G. **Introdução à programação**. Rio de Janeiro: Elsevier, 2002.

MARCONI, M. A.; LAKATOS, E. M. **Fundamentos de Metodologia Científica**. 5ª edição. São Paulo: Editora Atlas, 2003.

MARIOTTI, H.; ZAUHY, C. **A aprendizagem informal e o conceito 70: 20: 10**. 2020. Disponível em: <http://moityca.com/pdfs/APRENDIZAGEM%20INFORMAL.pdf>. Acesso em: 07 set 2024.

MARTINS, R.; REIS, R.; MARQUES, A. B. Inserção da programação no ensino fundamental Uma análise do jogo Labirinto Clássico da Code.org através de um modelo de avaliação de jogos educacionais. In: **Anais do Workshop de Informática na Escola**, 2016.

MEDEIROS, T. J.; SILVA, T. R.; ARANHA, E. H. S. Silva. Ensino de programação utilizando jogos digitais: uma revisão sistemática da literatura. **Renote**, v. 11, n. 3, p. 1-10, 2013.

MINAYO, M. C. S. (Org.). **Pesquisa social: teoria, método e criatividade**. Petrópolis: Vozes, 2001.

MINISTÉRIO DA EDUCAÇÃO. (2018). **Base Nacional Comum Curricular**. Brasília: MEC. Disponível em: <http://basenacionalcomum.mec.gov.br/>

OLIVEIRA, D. D. Disseminando conceitos de ouvidoria utilizando a metodologia “aprender fazendo”. **Revista Científica da Associação Brasileira de Ouvidores/Ombudsman**, v. 2, n. 2, p. 131-138, 2019.

OLIVEIRA, M. V.; RODRIGUES, L. C.; QUEIROGA, A. P. G. Material didático lúdico: uso da ferramenta *Scratch* para auxílio no aprendizado de lógica da programação. In: **Anais do XXII Workshop de Informática na Escola**. SBC, 2016. p. 359-368.

OLIVEIRA, W.; CAMBRAIA, A. C.; HINTERHOLZ, L T. Pensamento Computacional por meio da Computação Desplugada: Desafios e Possibilidades. In: **Workshop**

Sobre Educação em Computação (WEI), 29. , 2021, Evento Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2021.

PASQUAL JÚNIOR, P. A. **Pensamento computacional e formação de professores: Uma análise a partir da plataforma Code**. 2018. Tese de Doutorado. Dissertação (Mestrado em Educação)-Universidade de Caxias do Sul, Caxias do Sul.

PIAGET, J. **Seis estudos de psicologia**. Editora Forense, 1969. 136 p.

PONTES, R. et al. Avaliação de ferramentas para ensino de programação para crianças e adolescentes. **Anais do XXV Workshop de Informática na Escola**. SBC, 2019. p. 744-752.

REESE, Hayne W. The learning-by-doing principle. **Behavioral Development Bulletin**, v. 17, n. 1, p. 1, 2011.

RESNICK, M. et al. *Scratch*: programming for all. **Communications of the ACM**, v. 52, n. 11, p. 60-67, 2009.

RIBEIRO, L.; FOSS, L.; CARVALHEIRO, S. A C. Pensamento computacional: Fundamentos e integração na educação básica. **VIII Congresso Brasileiro de Informática na Educação (CBIE 2019)**; VIII Jornada de Atualização em Informática na Educação (JAIE 2019), p. 25-63, 2019.

RODRIGUES, A. K. M.; SILVA, A. P. M.; CARNEIRO, M. M. Ensino de Pensamento Computacional para alunos do ensino básico usando Computação Desplugada e *Scratch*. **Em Extensão**, Uberlândia, v. 20, n. 2, p. 228-240, 2021.

ROGÉRIO, H. E. et al. Criação de um modelo de avaliação utilizando PDCA e robótica. **Revista Contemporânea**, v. 4, n. 1, 1960–1971, 2024.

SANCHES, I. Do ‘aprender para fazer’ ao ‘aprender fazendo’: as práticas de Educação inclusiva na escola. **Revista Lusófona de Educação**, v. 19, n. 19, p. 135-156, 2011.

SCAICO, P. D. et al. Ensino de programação no ensino médio: Uma abordagem orientada ao design com a linguagem *scratch*. **Revista Brasileira de Informática na Educação**, v. 21, n. 02, p. 92, 2013.

SILVA, G. T.; SOUZA, J. L.; SILVA, L. A. M. Aplicação da ferramenta *Scratch* para o aprendizado de programação no ensino fundamental I. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2016. p. 1285.

SCRATCH. **Scratch for Educators**. Disponível em: <<https://scratch.mit.edu/educators>> . Acesso em: 23 out 2023.

SOUZA, F. A. et al. O ensino de programação na Educação Básica: uma revisão da literatura. **Anais do XXXII Simpósio Brasileiro de Informática na Educação**, p. 1265-1275, 2021.

VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. **Revista e-Curriculum**, v. 14, n. 3, p. 864–897, 2016.

VALENTE, S.; SARAIVA, D.; PESSOA, T. Educ@rtenatureza: relação dialógica entre educação, natureza e tecnologias digitais. **Video Journal of Social and Human Research**, v. 3, n. 1, p. 58-67, 2024.

VENTURA, M. M. O estudo de caso como modalidade de pesquisa. **Revista SoCERJ**, v. 20, n. 5, p. 383-386, 2007.

WING, J. Computational thinking. **Communications of the ACM**, v. 49, n. 3, p. 33-35, 2006.

WING, J. M. **Pensamento computacional**. Educação e Matemática, n. 162, p. 2-4, 2021.

WINSLOW, L. E. Programming pedagogy: a psychological overview. **ACM Sigcse Bulletin**, v. 28, n. 3, p. 17-22, 1996.

YIN, R. **Estudo de caso: planejamento e métodos**. 2a ed. Porto Alegre: Bookman; 2001.

Os autores declararam não haver qualquer potencial conflito de interesses referente a este artigo.