



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO PIAUÍ
CAMPUS TERESINA ZONA SUL
LICENCIATURA EM COMPUTAÇÃO

LUCAS CARDOSO LIMA

**EDUSCRUM COMO FERRAMENTA DO ENSINO DE PENSAMENTO
COMPUTACIONAL PARA O CURSO TÉCNICO DE INFORMÁTICA DO
IFPI/CATZS**

TERESINA
2026

LUCAS CARDOSO LIMA

**EDUSCRUM COMO FERRAMENTA DO ENSINO DE PENSAMENTO
COMPUTACIONAL PARA O CURSO TÉCNICO DE INFORMÁTICA DO
IFPI/CATZS**

Trabalho de Conclusão de Curso (artigo científico) apresentado como exigência parcial para obtenção do diploma do Curso de Licenciatura em Computação do Instituto Federal de Educação, Ciência e Tecnologia do Piauí, Campus Zona Sul.

Professor Orientador: Dr. Stephenson de Sousa Lima Galvão

TERESINA

2026

EDUSCRUM COMO FERRAMENTA DO ENSINO DE PENSAMENTO COMPUTACIONAL PARA O CURSO TÉCNICO DE INFORMÁTICA DO IFPI/CATZS

Lucas Cardoso Lima¹

Stephenson de Sousa Lima Galvão²

RESUMO

Introdução: O Pensamento Computacional é uma competência essencial no contexto educacional contemporâneo, envolvendo habilidades como decomposição, reconhecimento de padrões, abstração e elaboração de algoritmos. Seu desenvolvimento é fundamental para a resolução de problemas em diferentes áreas do conhecimento. Nesse sentido, metodologias ativas, como o EduScrum, têm sido utilizadas para promover aprendizagem colaborativa, organização de tarefas e protagonismo discente, contribuindo para o desenvolvimento dessas habilidades.

Objetivo: Analisar a aplicação do EduScrum no ensino do Pensamento Computacional em alunos do curso Técnico em Informática do Instituto Federal do Piauí – Campus Teresina Zona Sul. **Metodologia:** A pesquisa caracteriza-se como exploratória, de natureza aplicada, com abordagem qualitativa e procedimentos de pesquisa-ação. Participaram 10 estudantes do 1º ano do Ensino Médio. Foi desenvolvido um curso de Arduino com 30 horas, no qual os alunos construíram um semáforo. A coleta de dados ocorreu por meio de questionário em escala Likert, analisado com base em estatística descritiva. **Resultados:** Os resultados indicaram avanços nos quatro pilares do Pensamento Computacional, com destaque para decomposição e algoritmos. O reconhecimento de padrões apresentou desempenho intermediário, enquanto a abstração revelou maior dificuldade. Também se observou percepção positiva em relação ao trabalho colaborativo, apesar de variações na organização dos grupos. **Conclusão:** Conclui-se que o uso do EduScrum contribuiu para o desenvolvimento do Pensamento Computacional, favorecendo a aprendizagem ativa e colaborativa, embora abstração e organização em equipe ainda demandem maior atenção pedagógica.

Palavras-chave: pensamento computacional; eduscrum; metodologias ativas.

¹ Graduando em Licenciatura em Computação. Instituto Federal do Piauí. Email: l.c.limasud@gmail.com

² Professor Orientador. Instituto Federal do Piauí. Email: sgalvao@ifpi.edu.br

ABSTRACT

Abstract: Computational Thinking is an essential competency in contemporary education, involving skills such as decomposition, pattern recognition, abstraction, and algorithm design. Its development is fundamental for problem-solving across different areas of knowledge. In this context, active methodologies such as EduScrum have been used to promote collaborative learning, task organization, and student protagonism, contributing to the development of these skills. **Objective:** To analyze the application of EduScrum in teaching Computational Thinking to students in the Technical Course in Informatics at the Federal Institute of Piauí – Teresina Zona Sul Campus. **Methodology:** This study is characterized as exploratory, applied in nature, with a qualitative approach and action-research procedures. Ten first-year high school students participated. A 30-hour Arduino course was developed, in which students built a traffic light project. Data were collected through a Likert-scale questionnaire and analyzed using descriptive statistics. **Results:** The results indicated advances in the four pillars of Computational Thinking, especially decomposition and algorithms. Pattern recognition showed intermediate performance, while abstraction presented greater difficulty. A positive perception regarding collaborative work was also observed, despite variations in group organization. **Conclusion:** It is concluded that the use of EduScrum contributed to the development of Computational Thinking, fostering active and collaborative learning, although abstraction and team organization still require greater pedagogical attention.

Keywords: computational thinking; eduscrum; technical education.

Data de Aprovação: 07/05/2026

1 INTRODUÇÃO

O avanço tecnológico e o surgimento de novas tecnologias impactam diretamente a sociedade, transformando as formas de comunicação, trabalho e aprendizagem. Nesse contexto, torna-se cada vez mais necessária a inclusão do ensino da computação na educação básica e superior, de modo que as futuras gerações adquiram conhecimentos fundamentais para compreender, utilizar e desenvolver tecnologias voltadas à resolução de problemas.

Nesse cenário, destaca-se o conceito de Pensamento Computacional definido como uma forma de pensar baseada nos princípios da ciência da computação aplicada à resolução de problemas (Wing, 2006, p. 33, tradução nossa). Essa competência envolve habilidades como decomposição, reconhecimento de padrões, abstração e elaboração de algoritmos (Valente, 2017; Bittencourt; Silva, 2017), sendo considerada uma competência essencial no século XXI (Wing, 2006).

No ensino técnico, especialmente no curso Técnico em Informática, o desenvolvimento do Pensamento Computacional é fundamental para a compreensão da lógica e da estrutura das soluções computacionais. No entanto, práticas tradicionais de ensino nem sempre favorecem o engajamento e a autonomia dos estudantes, tornando necessária a adoção de abordagens que estimulem o protagonismo discente e a resolução de problemas. Nesse contexto, tem-se as metodologias ativas, que promovem a participação ativa do aluno na construção do conhecimento. Dentre elas, o EduScrum, adaptação educacional do Scrum (Método de desenvolvimento de Software), propõe um ambiente colaborativo e organizado, favorecendo o trabalho em equipe, a autonomia e o pensamento crítico.

Dessa forma, este trabalho analisou a aplicação do EduScrum como ferramenta para o ensino do Pensamento Computacional no curso Técnico em Informática do Instituto Federal do Piauí – Campus Teresina Zona Sul (IFPI-CATZS). Para isso, foi desenvolvido um curso de Robótica Educacional com alunos do 1º ano do Ensino Médio, no qual os estudantes construíram um semáforo como projeto final. A coleta de dados foi realizada por meio de um questionário em escala Likert. Os resultados indicaram avanços nos quatro pilares do Pensamento Computacional, com destaque para decomposição e algoritmos, além de uma percepção positiva em relação ao trabalho colaborativo. Assim, o estudo contribui para a reflexão sobre o uso do EduScrum como estratégia para tornar o ensino mais dinâmico, participativo e alinhado às demandas contemporâneas.

2 REFERENCIAL TEÓRICO

2.1 PENSAMENTO COMPUTACIONAL

De acordo com Wing (2006), o Pensamento Computacional é definido como a capacidade de um ser humano de entender e resolver problemas por meio da aplicação de princípios da ciência da computação. Esta é uma habilidade importante para todos, não apenas para cientistas da computação, porque permite que as pessoas resolvam problemas de maneira rápida e criativa, sua importância no dia a dia é clara em várias situações, desde a organização de tarefas cotidianas até a resolução de problemas mais complexos. Desta maneira, Huang e Qiao (2022) ressaltam que a educação em inteligência artificial, baseada no Pensamento

Computacional, aprimora as habilidades de resolução de problemas e o pensamento crítico, que são essenciais para as exigências tecnológicas atuais. Essas habilidades são desenvolvidas porque, como indicado por Vicari, Moreira e Menezes (2018), o Pensamento Computacional é baseado em quatro pilares fundamentais: (i) Decomposição — resolver um problema dividindo-o em partes menores, cada uma das quais é mais fácil de entender e analisar; (ii) Reconhecimento de Padrões — encontrar semelhanças ou repetições em um único problema; (iii) Abstração — remover detalhes desnecessários e focar nas questões mais importantes; e (iv) Algoritmo — um conjunto claro de instruções ou etapas que precisam ser seguidas para resolver o problema.

Visando a importância dos pilares do Pensamento Computacional no contexto escolar, ele torna-se um tema que deve ser integrado ao currículo escolar e se torna essencial no ensino técnico de Informática, devido à sua conexão direta com essa área. Um exemplo disso é um estudo realizado por Zanetti e Oliveira (2015), que buscou intervir no processo de aprendizagem de programação de computadores, visando reduzir as principais dificuldades enfrentadas por estudantes iniciantes do ensino médio que estão no curso Técnico em Informática. A metodologia utilizada envolveu a aplicação de uma linguagem de programação visual em conjunto com a robótica pedagógica para desenvolver o Pensamento Computacional. Durante as oficinas, os pesquisadores notaram que os alunos se tornaram participantes ativos na construção do conhecimento, o que também ajudou a aprimorar seu raciocínio lógico.

Todavia, não somente na área da tecnologia, a aplicação do Pensamento Computacional tem se mostrado eficaz na melhoria do aprendizado em outras disciplinas. Por exemplo, a incorporação do Pensamento Computacional no ensino de Matemática tem apresentado resultados positivos, oferecendo uma abordagem abrangente que inclui raciocínio lógico, reconhecimento de padrões e o desenvolvimento de soluções tecnológicas, como robótica e simuladores (Azevedo, 2024). Assim, Vieira e Hai (2022) defendem que a inclusão do Pensamento Computacional nos currículos escolares, em sintonia com os eixos de cultura digital e mundo digital, favorece o desenvolvimento de habilidades tecnológicas e cognitivas em diversas áreas do conhecimento, preparando os alunos para os desafios da sociedade moderna. Além disso, a adoção de metodologias ativas em conjunto com o Pensamento Computacional, incentiva a colaboração, a criatividade e a autonomia dos estudantes, tornando o processo de aprendizado mais dinâmico e significativo.

Contudo, no cenário educacional brasileiro, a Base Nacional Comum Curricular (BNCC) ressalta a relevância do Pensamento Computacional nos currículos escolares. No entanto, a falta de diretrizes específicas gerou a necessidade de normas complementares, como a Portaria nº 1.432 de 2022, que orienta a implementação do Pensamento Computacional nas redes de ensino estaduais (BRASIL, 2022). Pesquisas mostram que estados como Paraná, Minas Gerais e Pernambuco têm avançado na criação de componentes curriculares obrigatórios voltados para o tema, enquanto outras regiões ainda apresentam abordagens limitadas, muitas vezes restritas à Matemática e ao ensino de programação (Araújo; Silva, 2024).

Então nota-se que o Pensamento Computacional é uma competência transversal de grande relevância no mundo atual, contribuindo para a formação de indivíduos capazes de enfrentar desafios complexos de forma criativa e eficiente. Sua integração nos currículos escolares, apoiada por políticas educacionais atualizadas, é crucial para o desenvolvimento de habilidades essenciais dos alunos, preparando-os para as exigências do mercado de trabalho e da vida em sociedade.

2.1.1 FORMAS DO ENSINO DO PENSAMENTO COMPUTACIONAL

O ensino do Pensamento Computacional pode ocorrer por diferentes abordagens e ferramentas pedagógicas. Entre as mais recorrentes na literatura destacam-se a programação baseada em blocos, a Computação Desplugada e a Robótica Educacional. A programação em blocos favorece a aprendizagem inicial ao abstrair a complexidade sintática e priorizar a construção lógica (Kelleher; Pausch, 2005; Wernthrop; Wilensky, 2015). Já a Computação Desplugada possibilita o desenvolvimento de habilidades computacionais sem o uso de dispositivos digitais, sendo especialmente relevante em contextos com poucos recursos tecnológicos (Bell et al., 2011; Rodrigues, 2017). Essas abordagens evidenciam diferentes possibilidades para o desenvolvimento do Pensamento Computacional em contextos educacionais.

Dentre essas abordagens, destaca-se a Robótica Educacional, por articular teoria e prática em atividades que favorecem experimentação, resolução de problemas e aprendizagem colaborativa. Segundo Reis, Sarmento e Zaramella (2014), a robótica proporciona um ambiente desafiador e lúdico, estimulando raciocínio lógico, criatividade e trabalho em grupo, elementos diretamente

relacionados ao Pensamento Computacional. Além disso, a robótica educacional favorece a aplicação concreta de conceitos computacionais por meio da construção e programação de artefatos, aproximando os estudantes de situações reais de resolução de problemas. Embora sua difusão ainda enfrente desafios, como custos de kits e formação docente (Albuquerque et al., 2016), seu potencial pedagógico a torna uma estratégia relevante para o ensino do Pensamento Computacional, especialmente em propostas que utilizam plataformas acessíveis, como o Arduino.

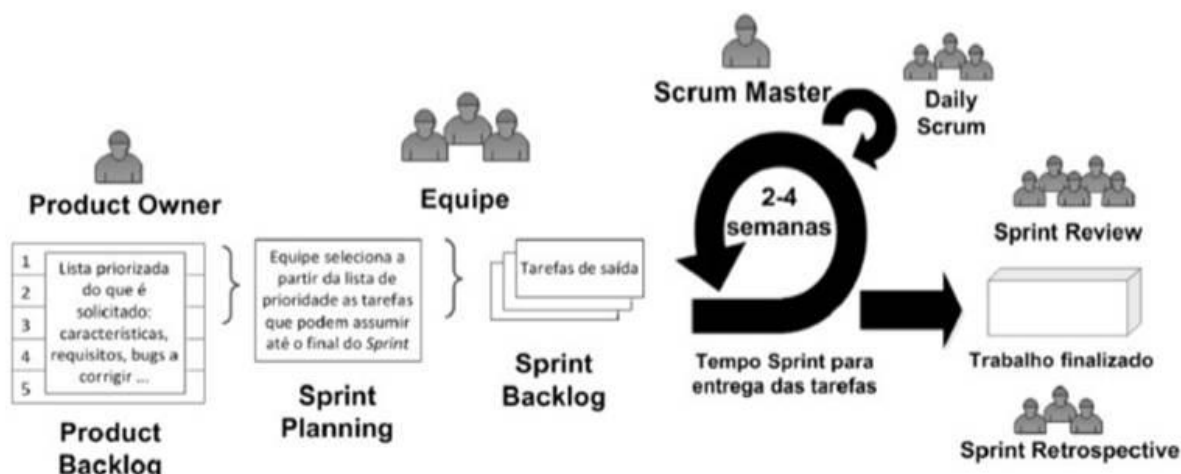
2.2 EDUSCRUM

O EduScrum é a adaptação do *método de desenvolvimento de software* Scrum, para o ambiente educacional. Os princípios fundamentais do Scrum foram inicialmente desenvolvidos por Takeuchi e Nonaka (1986), sendo posteriormente aplicados ao desenvolvimento de softwares por Degrace e Stahl (1990). As diretrizes, funções, eventos e artefatos que estruturam o Scrum atualmente estão documentados no "Guia Scrum", onde é descrito como um método utilizado para gerenciar o desenvolvimento de produtos complexos desde o início da década de 1990. Segundo Schwaber e Sutherland (2017, p. 3), o Scrum permite a incorporação de diferentes processos e técnicas dentro de sua estrutura.

Já o EduScrum (usam das palavras "Educação" e "Scrum") foi criado em 2011 pelo professor de Química Willy Wijnands, do Colégio Ashram, localizado em Alphen aan den Rijn, na Holanda. Wijnands implementou essa metodologia com alunos entre 12 e 18 anos, promovendo um aprendizado mais colaborativo e estruturado (Sutherland, 2017).

Na Figura 1, observa-se a representação do método EduScrum aplicado em sala de aula, no qual existem três papéis essenciais: *Product Owner* (professor), *Scrum Master* (líder da equipe) e equipe de estudantes. No qual, o professor assume definir as tarefas da *sprint*; o líder do grupo, juntamente com a equipe, seleciona as tarefas para serem realizadas dentro daquela *sprint*; e a equipe de estudantes desenvolvem as tarefas da *sprint*, no qual cada *sprint* dura aproximadamente de 2 a 4 semanas. Ao final de cada *sprint*, a equipe entrega o produto e juntos fazem um *feedback* das tarefas realizadas e como melhorar nas próximas *sprints*.

Figura 1: Esquematisação da aplicação do Método *EduScrum*



Fonte: Silva *et al.*, 2016

No contexto educacional, o professor, como Product Owner, é responsável por: i) definir os conteúdos de aprendizagem; ii) facilitar a implementação do EduScrum e o desenvolvimento individual e coletivo dos alunos; iii) monitorar e aprimorar a qualidade dos resultados educacionais; e iv) avaliar e interpretar o desempenho acadêmico.

Os conteúdos de aprendizagem compreendem os conhecimentos e habilidades necessários para atingir determinados objetivos educacionais, englobando competências cognitivas, motoras, afetivas, interpessoais e sociais (Zabala, 1998). A composição das equipes de estudantes é conduzida pelo professor por meio de um formulário de autoavaliação, no qual os alunos refletem sobre suas competências e características. A partir dessa análise, são formados grupos auto-organizáveis e autogerenciáveis, garantindo diversidade de habilidades e motivação para a realização das atividades.

A motivação dos estudantes para participar das atividades e assimilar os conteúdos influencia diretamente o êxito do processo educacional. Assim, é fundamental que os docentes considerem esse aspecto na estruturação das aulas (Lourenço; Paiva, 2010; Zabala, 1998). Para estimular o engajamento, Wijnands e Stolze (2019) sugerem a aplicação do "círculo dourado" de Sinek (2018), que enfatiza a importância de compreender o "porquê" de uma atividade antes de abordar "o que" e "como" realizá-la. Dessa forma, contextualizar o aprendizado auxilia na construção de significados e na assimilação dos conceitos, conforme preconizado por Ausubel,

que destaca a necessidade de relacionar novos conhecimentos à estrutura cognitiva prévia dos alunos (Carvalho; Porto; Belhot, 2001, p. 87).

Após essa etapa, o professor disponibiliza aos alunos o backlog do trimestre, que consiste em "uma lista ordenada dos conteúdos de aprendizagem e métodos de trabalho que os alunos deverão adquirir durante a disciplina" (Delhij; Van Solingen; Wijnands, 2016, p. 17). O backlog é trabalhado por meio de sprints, definidos por Wijnands e Stolze (2019, p. 100) como "um período determinado de tempo no qual as equipes trabalham de acordo com o ciclo PDCA (Plan, Do, Check, Act)". Dessa maneira, o professor estabelece a carga de trabalho a ser cumprida em um determinado intervalo de tempo.

Durante a realização das atividades, o professor atua como mentor das equipes, oferecendo *feedbacks*, esclarecendo dúvidas, orientando metodologias de estudo, ministrando aulas expositivas, fornecendo materiais complementares e auxiliando no trabalho em equipe e nos eventos do EduScrum. A avaliação dos alunos considera não apenas o desempenho individual, mas também a colaboração em equipe e as entregas realizadas. Esse processo avaliativo abrange aspectos conceituais (o que o aluno deve saber), procedimentais (o que o aluno deve ser capaz de fazer) e atitudinais (como o aluno deve se comportar) (Coll et al., 2000). Por fim, o professor incentiva os alunos na busca pelo Kaizen, um conceito japonês que enfatiza a melhoria contínua, tanto no desenvolvimento individual quanto na colaboração em equipe (Wijnands; Stolze, 2019).

Os estudantes desempenham um papel central no EduScrum, sendo responsáveis pelo desenvolvimento, pela entrega e pela qualidade do trabalho realizado durante o sprint. Além disso, um dos alunos assume a função de Scrum Master, cuja responsabilidade é facilitar a colaboração entre os membros da equipe e garantir a correta aplicação do *framework* EduScrum, promovendo a realização dos eventos essenciais para a metodologia. A seleção do Scrum Master é feita internamente pela própria equipe, promovendo autonomia e senso de responsabilidade entre os estudantes.

Assim como o Scrum tradicional, o EduScrum é estruturado em uma série de eventos fundamentais para o desenvolvimento da aprendizagem. Esses eventos representam momentos formais de inspeção e adaptação do processo educacional, permitindo ao professor avaliar o progresso dos alunos e fornecer *feedbacks*

direcionados. Dessa forma, a metodologia contribui para a evolução contínua das competências e habilidades dos estudantes e das equipes.

A Figura 2 ilustra de maneira simplificada os ciclos de desenvolvimento no EduScrum. Observa-se que o processo se inicia com a definição da lista de atividades do projeto, que estabelece as tarefas iniciais e os objetivos principais. Em seguida, ocorrem os sprints, estruturados em ciclos semanais, nos quais são realizadas reuniões diárias para acompanhamento do progresso. Finalmente, o processo culmina na entrega do artefato, produto ou atividade concluída, refletindo o aprendizado e a evolução da equipe ao longo do ciclo.

Figura 2: Representação das etapas de desenvolvimento do EduScrum.



Fonte: Borges, Schmitt e Nakle (2014).

3 METODOLOGIA

Esta pesquisa, quanto aos objetivos, caracteriza-se como exploratória, de natureza aplicada, com abordagem qualitativa e procedimento de pesquisa-ação, a seguir são explicados cada um desses pontos. Segundo Gil (2017), as pesquisas exploratórias têm como objetivo proporcionar maior familiaridade com o problema, visando torná-lo mais explícito, explorando hipóteses. A pesquisa aplicada, de acordo com Prodanov e Freitas (2013), tem o objetivo de gerar conhecimentos voltados para a aplicação prática, visando a solução de problemas específicos e essa pesquisa aplicou uma metodologia em contexto real de ensino, gerando importantes conhecimentos na área de estudo. A abordagem qualitativa é uma estratégia

metodológica dedicada à exploração de fenômenos sociais em seu ambiente natural, enfatizando a interpretação e o significado atribuídos pelos indivíduos a esses fenômenos (Minayo, 2014), o que foi feito com captação das experiências dos alunos, permitindo uma compreensão mais aprofundada das percepções dos participantes.

Por fim, a pesquisa-ação caracteriza-se pela articulação entre investigação e intervenção, envolvendo pesquisadores e participantes de forma cooperativa na compreensão e enfrentamento de problemas reais (Thiollent, 1986).

A pesquisa foi realizada no Instituto Federal do Piauí (IFPI), Campus Teresina Zona Sul, onde o campus exerce papel relevante ao oferecer educação técnica e profissional à comunidade. O estudo teve como foco o Curso Técnico em Informática, ofertado na forma concomitante ao ensino médio e estruturado em núcleos formativos que garantem formação interdisciplinar. A condução da pesquisa seguiu rigorosamente os aspectos éticos previstos nas Resoluções CNS nº 466/2012 e nº 510/2016, tendo sido aprovada pelo Comitê de Ética em Pesquisa (Parecer nº 7.058.582/2024; CAAE: 78585124.1.0000.5210). Todos os participantes receberam Termo de Assentimento (TALE) ou Termo de Consentimento Livre e Esclarecido (TCLE), com garantia de esclarecimentos prévios, voluntariedade, anonimato e confidencialidade dos dados, preservando sua autonomia, valores e direitos.

A seleção dos participantes da pesquisa ocorreu por auto seleção, envolvendo os 10 primeiros estudantes interessados em participar do curso voltado à aplicação da metodologia ativa EduScrum para o ensino do Pensamento Computacional no Curso Técnico em Informática do IFPI–Zonas Sul. Para garantir a relevância da amostra e a validade dos resultados, foram definidos critérios de inclusão — como estar regularmente matriculado, ter disponibilidade para as atividades e assinar o TALE/TCLE — e critérios de exclusão, que abrangeram alunos afastados ou que trancaram o curso, aqueles que não assinaram os termos éticos e participantes de outras pesquisas simultâneas que pudessem comprometer os resultados. Esses critérios asseguraram que os envolvidos contribuíssem de forma adequada aos objetivos do estudo.

O curso desenvolvido no contexto da pesquisa foi estruturado em 10 encontros presenciais, com duração de 3 horas cada, totalizando 30 horas de atividades. Durante os encontros, os alunos participaram de atividades teórico-práticas voltadas ao ensino de Pensamento Computacional por meio da utilização da metodologia EduScrum, incluindo momentos de organização das tarefas, desenvolvimento do

projeto e acompanhamento das etapas propostas. Ao longo do curso, os estudantes desenvolveram um projeto de semáforo utilizando Arduino, permitindo a aplicação dos conceitos abordados em situações reais de aprendizagem. A Tabela 1 apresenta a organização detalhada de cada encontro, especificando os conteúdos abordados e os procedimentos metodológicos adotados ao longo das dez aulas, desde a apresentação inicial do projeto até a etapa de apresentação final das maquetes desenvolvidas pelos grupos.

Tabela 1: Estrutura do curso: conteúdos e metodologia por encontro

AULA	CONTEÚDO	METODOLOGIA
1	Apresentação do projeto semáforo; Pensamento Computacional	Inicialmente, foi apresentado aos estudantes o projeto do semáforo, destacando seus objetivos, etapas de desenvolvimento e produto final esperado. Em seguida, foram introduzidos os conceitos fundamentais do Pensamento Computacional, com ênfase em decomposição, reconhecimento de padrões e elaboração de algoritmos. Para a aplicação desses conceitos, foram propostos desafios desplugados e plugados, possibilitando aos estudantes experimentar estratégias de resolução de problemas e relacionar os conceitos trabalhados às etapas do projeto.
2	Arduino; Acionamento de LED com Arduino	Inicialmente, os estudantes foram organizados em grupos para a aplicação da metodologia Eduscrum, sendo definido um líder para cada equipe. Posteriormente, foram apresentados os conceitos básicos relacionados ao Arduino e ao funcionamento de LEDs, seguidos da demonstração e construção de um algoritmo para seu acionamento. Como atividade prática, os grupos foram desafiados a programar o acionamento alternado dos LEDs, mobilizando conceitos de algoritmos, reconhecimento de padrões e abstração. Ao final da atividade, foi realizado um momento de feedback colaborativo, conduzido com a participação dos próprios estudantes. Os grupos discutiram as dificuldades encontradas durante a realização do desafio e avaliaram as estratégias utilizadas.
3	Funções: automatização do semáforo	Foram apresentados os conceitos de funções na programação, destacando sua utilização na organização e reutilização de códigos e na automatização de comandos. A partir de exemplos

		práticos, os estudantes foram orientados a identificar padrões e estruturar algoritmos utilizando funções. Como desafio da aula, cada grupo deveria programar três LEDs para representar o funcionamento de um semáforo, aplicando os conceitos de algoritmos, reconhecimento de padrões e abstração. Ao final da atividade, os estudantes realizaram um feedback colaborativo sobre o desenvolvimento do desafio, discutindo as dificuldades encontradas, as estratégias utilizadas e os aspectos que poderiam ser aprimorados. A partir dessa discussão, foram levantadas propostas de melhoria para orientar o planejamento e a execução das atividades da aula seguinte.
4	Como projetar uma maquete.	Foi apresentado aos estudantes o produto final do projeto: a construção de uma maquete representando um cruzamento viário equipado com dois semáforos sincronizados, de modo a organizar e controlar o fluxo do cruzamento. Em seguida, foram disponibilizados os materiais necessários para a construção da maquete. Os estudantes, organizados em grupos, foram responsáveis pelo planejamento e desenvolvimento do projeto, realizando discussões, elaboração de desenhos e definição da estrutura da maquete. A distribuição das responsabilidades foi realizada pelos próprios estudantes, que discutiram entre si as tarefas necessárias, organizaram o trabalho da equipe e acompanharam o desenvolvimento das diferentes etapas do projeto. Dessa forma, buscou-se estimular a autonomia, a colaboração e a corresponsabilidade dos participantes na construção do produto final.
5	Construção da maquete: parte 1	Durante a etapa de construção da maquete, os grupos tiveram autonomia para decompor o projeto em partes menores, definindo as etapas necessárias para a construção do produto final. A cada semana, as equipes deveriam entregar um artefato ou incremento do projeto, que seria utilizado como base para o desenvolvimento da etapa seguinte. Ao final de cada ciclo, os grupos analisavam o que havia sido desenvolvido, identificavam dificuldades e definiam os próximos passos para o ciclo seguinte. Assim, o desenvolvimento da maquete ocorreu de forma incremental e iterativa, permitindo que os estudantes aprimorassem continuamente o produto a partir das experiências e dos feedbacks obtidos ao longo do projeto.

6	Construção da maquete: 2	Nessa etapa, os alunos deram início à construção das vias e à programação dos dois semáforos, artefatos que deveriam ser concluídos e entregues ao final da aula. Ao término da atividade, os grupos realizaram um momento de retrospectiva, avaliando coletivamente o desempenho do trabalho em equipe e discutindo possíveis pontos de melhoria para os ciclos seguintes.
7	Construção da maquete: 3	Nesta etapa, os esforços dos grupos concentraram-se na confecção da parte elétrica dos semáforos, incluindo a fiação e os suportes destinados à fixação dos LEDs, elaborados em conformidade com a lógica de funcionamento dos semáforos utilizados em vias públicas — considerando, portanto, a alternância de cores e os intervalos de tempo característicos desse tipo de sinalização. Paralelamente, os grupos deram início à construção de outras edificações voltadas à ambientação da maquete, contribuindo para a composição do cenário urbano representado. Ao término da atividade, os grupos realizaram um momento de retrospectiva, avaliando coletivamente o desempenho do trabalho em equipe e discutindo possíveis pontos de melhoria para os ciclos seguintes.
8	Construção da maquete: 4	Nesta etapa, os grupos concentraram-se na finalização da maquete, dedicando-se à construção da vegetação e dos demais ornamentos que compunham o cenário da cidade representada. Ao final do sprint, os grupos realizaram uma sessão de retrospectiva, momento em que compartilharam entre si o andamento dos trabalhos, apontaram dificuldades enfrentadas ao longo do processo e sugeriram melhorias para os próximos ciclos de desenvolvimento.
9	Construção da maquete: 5	A pedido dos próprios alunos, essa aula foi destinada à finalização dos detalhes e aos testes dos semáforos já instalados em seus respectivos locais na maquete. Os grupos realizaram os últimos ajustes necessários e, ao final da atividade, promoveram uma nova sessão de retrospectiva, na qual avaliaram o trabalho desenvolvido e alinharam os pontos a serem apresentados na aula seguinte — reforçando, mais uma vez, a dinâmica de reflexão contínua.

10	Apresentação do projeto final	Nesta aula, os grupos apresentaram suas maquetes em formato de exposição, momento em que os professores avaliaram a programação dos semáforos, a estrutura da maquete e a lógica de funcionamento dos semáforos em conformidade com o padrão adotado nas vias públicas. Ao final da apresentação, foi solicitado aos alunos um feedback quanto às aulas desenvolvidas ao longo do projeto e ao desempenho de cada grupo, encerrando o ciclo de sprints com uma última sessão de retrospectiva coletiva. Esse conjunto de etapas ilustra a aplicação prática dos princípios do EduScrum ao longo de todo o processo de construção da maquete, evidenciando o papel da reflexão contínua e do trabalho colaborativo na melhoria progressiva do produto final.
----	-------------------------------	---

Fonte: elaborada pelo autor (2026).

A coleta de dados da pesquisa foi realizada por meio de questionários qualitativos aplicados online via Google Forms durante e ao final do curso, entre setembro e outubro de 2024. Esse instrumento, elaborado com base no referencial teórico, reuniu informações essenciais para analisar a aplicação da metodologia ativa EduScrum para o ensino de Pensamento Computacional. O questionário incluía perguntas direcionadas e estruturadas, elaboradas com base nos objetivos da pesquisa, permitindo a coleta de percepções dos participantes de forma organizada e comparável. As questões foram avaliadas por meio da Escala Likert, possibilitando a mensuração das respostas. A aplicação ocorreu de forma individual, sem a utilização de gravações, e todo o material coletado será armazenado por cinco anos, conforme as normas éticas da Resolução nº 466/12 e as orientações do Comitê de Ética em Pesquisa.

4 ANÁLISE DOS DADOS

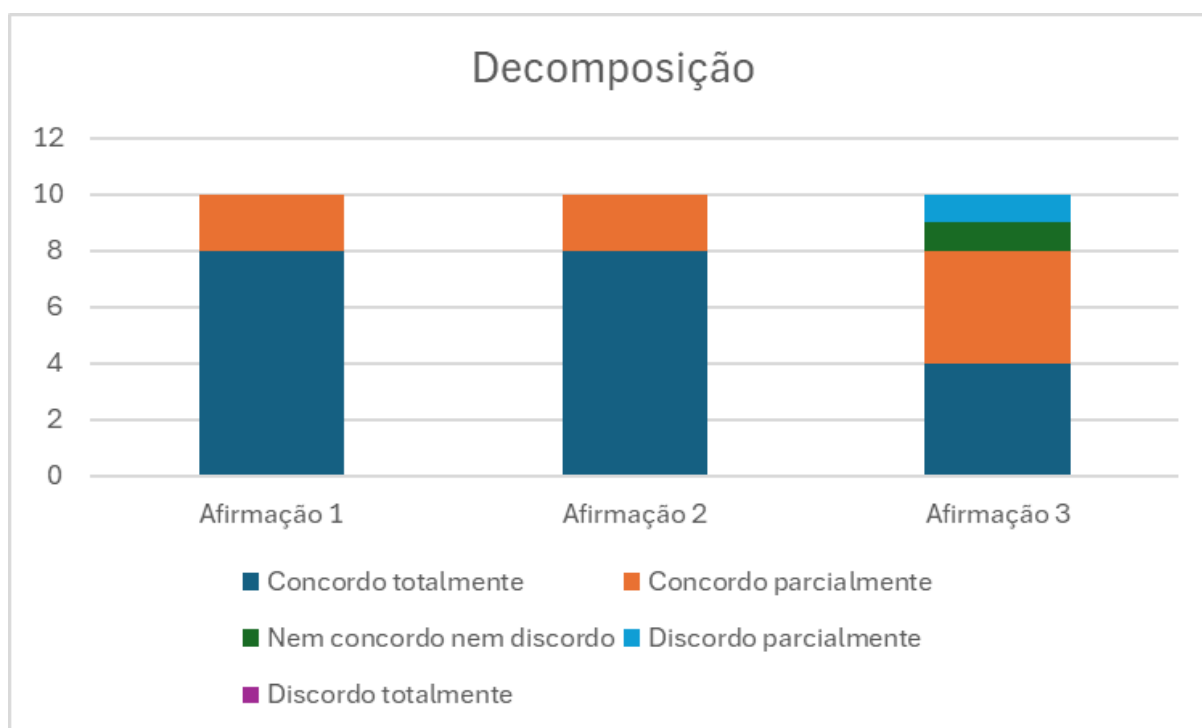
Os dados coletados por meio do questionário aplicado aos 10 alunos participantes do curso, cujo projeto final consistiu na construção de um semáforo, foram analisados e apresentados nesta seção. O instrumento foi composto por 15 afirmações estruturadas com base nos pilares do Pensamento Computacional — decomposição, reconhecimento de padrões, abstração e algoritmos —, além de contemplar aspectos relacionados ao trabalho colaborativo.

As respostas, obtidas por meio de uma escala Likert de cinco pontos, foram analisadas por meio de estatística descritiva, com ênfase no cálculo das médias por questão e por dimensão investigada. A seguir, apresentam-se os resultados organizados de acordo com os pilares do Pensamento Computacional, possibilitando uma análise mais detalhada das habilidades desenvolvidas pelos estudantes.

4.1 DECOMPOSIÇÃO

Considerando que a decomposição, segundo Vicari, Moreira e Menezes (2018), consiste na capacidade de resolver um problema por meio de sua divisão em partes menores e mais fáceis de compreender, as três primeiras afirmações do questionário foram elaboradas com o objetivo de analisar esse pilar do Pensamento Computacional. Um exemplo disso é o Gráfico 1 que apresenta a distribuição das respostas dos estudantes para essas afirmações, permitindo visualizar a percepção dos participantes quanto ao desenvolvimento da habilidade de decomposição durante a realização do projeto do semáforo.

Gráfico 1 – Resultados do questionário sobre o pilar Decomposição



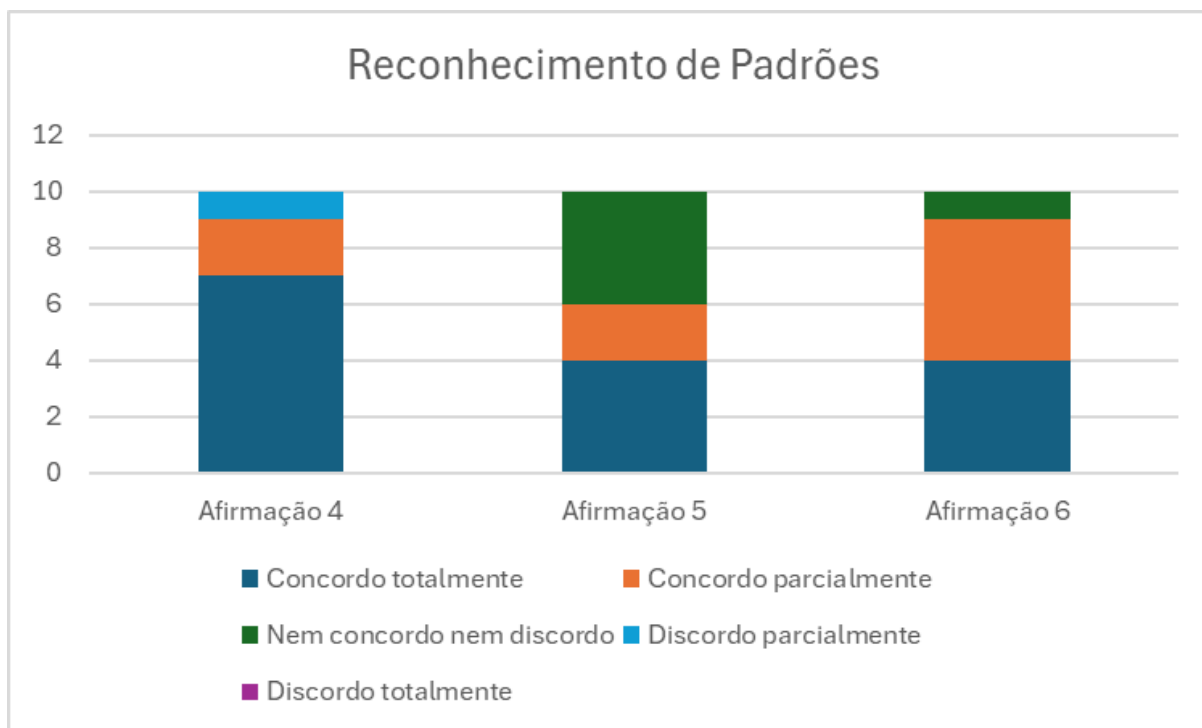
Fonte: Elaborado pelo autor (2026)

Analisando o Gráfico 1, observa-se que a maioria dos alunos concorda totalmente que conseguiu dividir o projeto do semáforo em partes menores e que resolver uma etapa por vez contribuiu para a compreensão do funcionamento do sistema. Esses resultados indicam que os estudantes desenvolveram habilidades relacionadas à organização e estruturação do problema. No entanto, na terceira afirmação — relacionada ao planejamento prévio das etapas antes da programação — as respostas mostraram maior variação, com parte dos alunos indicando concordância parcial e alguns posicionamentos neutros ou de discordância. Esse resultado sugere que, embora os alunos consigam executar a decomposição durante a resolução do problema, nem todos realizam esse processo de forma antecipada e sistematizada.

De modo geral, os resultados apontam para um elevado desenvolvimento do pilar Decomposição, com média de 4,57, evidenciando que os alunos foram capazes de dividir o problema em partes menores, compreender suas etapas e conduzir o projeto de forma estruturada. No qual esse resultado pode estar associado à dinâmica de organização das atividades em etapas (*sprint*) ao longo do projeto, o que favorece a prática da decomposição de forma aplicada.

4.2 RECONHECIMENTO DE PADRÕES

Segundo Vicari, Moreira e Menezes (2018), o reconhecimento de padrões refere-se à capacidade de identificar semelhanças e repetições em um problema. Nesse sentido, as afirmações 4, 5 e 6 do questionário foram elaboradas com o objetivo de analisar esse pilar do Pensamento Computacional. O Gráfico 2 apresenta a distribuição das respostas dos estudantes para essas afirmações, permitindo visualizar como os participantes perceberam o uso de padrões, sequências e reutilização de soluções durante o desenvolvimento do projeto do semáforo.

Gráfico 2 – Resultados do questionário sobre o pilar Reconhecimento de Padrões

Fonte: Elaborado pelo autor (2026)

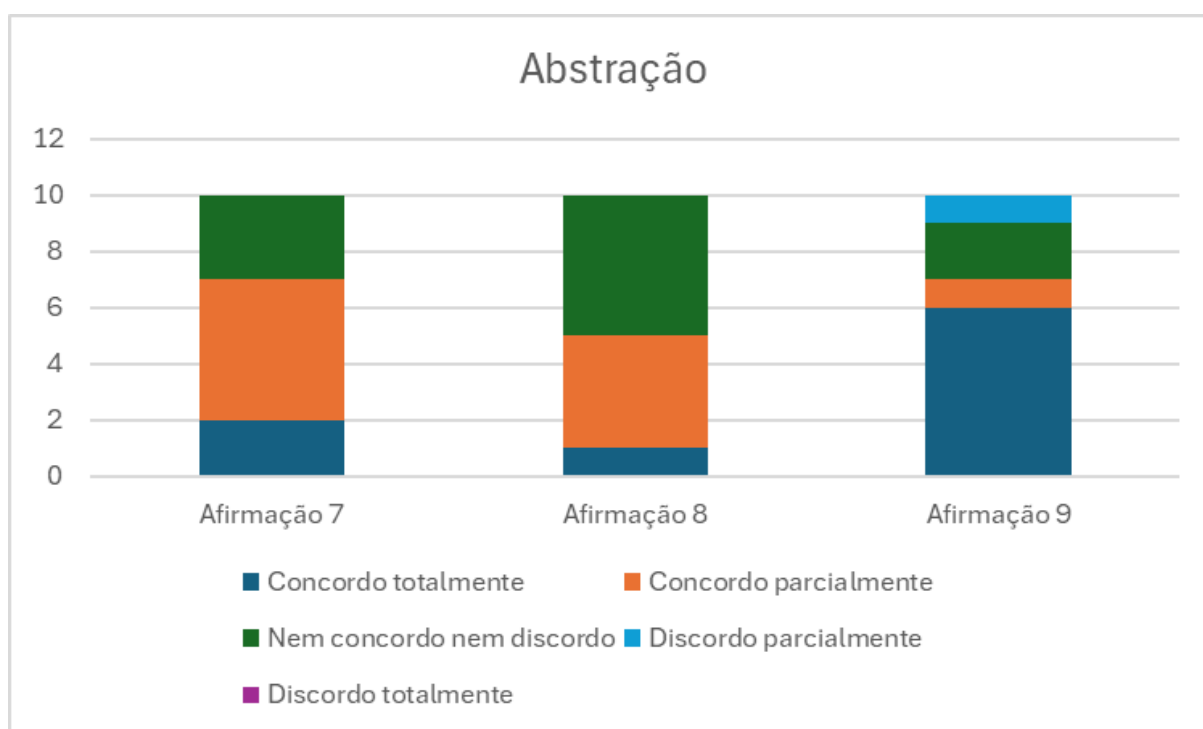
Analisando o Gráfico 2, observa-se que a maioria dos alunos concorda totalmente que o funcionamento do semáforo segue uma sequência repetitiva, evidenciando a capacidade de identificar padrões no comportamento do sistema. Entretanto, nas afirmações relacionadas à identificação de semelhanças entre erros e à reutilização de soluções previamente utilizadas, os resultados apresentaram maior dispersão, com presença significativa de respostas neutras e concordâncias parciais. Esse aspecto indica que os alunos ainda encontram dificuldades em reconhecer padrões em situações mais complexas, como análise de erros e reaproveitamento de estratégias.

De modo geral, o pilar Reconhecimento de Padrões apresentou média de 4,27, sugerindo que os alunos desenvolveram essa habilidade, especialmente no que se refere à identificação de sequências repetitivas, embora ainda haja necessidade de aprofundamento em contextos que envolvem análise e generalização de soluções. Isso sugere que atividades que envolvem depuração e comparação de soluções podem contribuir para o fortalecimento desse pilar.

4.3 ABSTRAÇÃO

Segundo Vicari, Moreira e Menezes (2018), a abstração refere-se à remover detalhes desnecessários e focar nas questões mais importantes. Nesse sentido, as afirmações 7, 8 e 9 do questionário foram elaboradas com o objetivo de analisar esse pilar do Pensamento Computacional. O Gráfico 3 apresenta a distribuição das respostas dos estudantes para essas afirmações, permitindo visualizar como os participantes perceberam habilidades relacionadas à identificação de elementos essenciais, filtragem de informações e compreensão geral do projeto desenvolvido.

Gráfico 3 – Resultados do questionário sobre o pilar Abstração



Fonte: Elaborado pelo autor (2026)

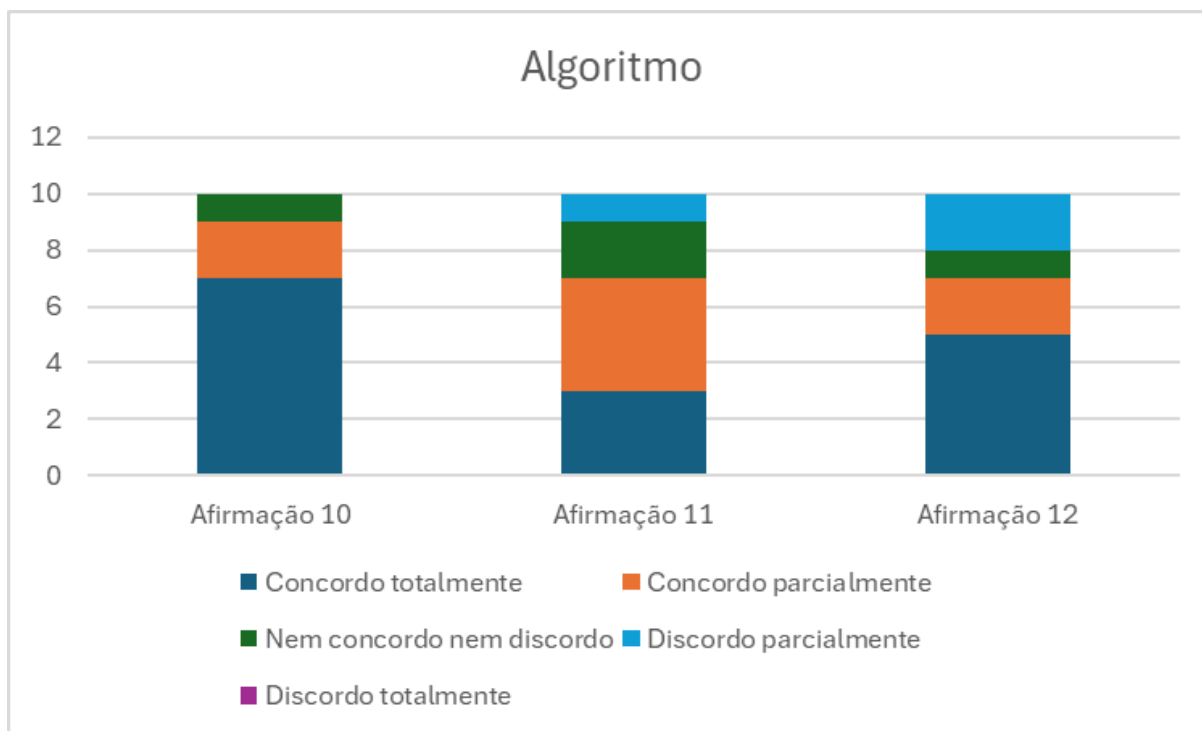
Analisando o Gráfico 3 podemos observar que a afirmação 7, que trata da capacidade de identificar os elementos mais importantes para o funcionamento do semáforo, apresentou predominância de respostas entre “concordo parcialmente” e “nem concordo nem discordo”, indicando que os alunos ainda encontram dificuldades em selecionar os aspectos essenciais do problema. Da mesma maneira, a afirmação 8, relacionada à capacidade de ignorar detalhes não essenciais, também apresentou maior concentração de respostas intermediárias, o que reforça a dificuldade dos

estudantes em filtrar informações irrelevantes durante a resolução do problema. Por outro lado, a afirmação 9, que aborda a compreensão do funcionamento geral do projeto mesmo sem o domínio de todos os detalhes técnicos, apresentou maior incidência de respostas em “concordo totalmente”, evidenciando que os alunos conseguem compreender o sistema como um todo.

Esses resultados indicam que, embora os estudantes consigam formar uma visão global do funcionamento do projeto, ainda apresentam limitações no processo de abstração, especialmente no que se refere à identificação de elementos essenciais e à eliminação de informações desnecessárias. De modo geral, o pilar Abstração apresentou média de 3,9, valor inferior aos demais pilares analisados, indicando um nível de desenvolvimento satisfatório, porém ainda em processo de consolidação.

4.4 ALGORITMOS

Segundo Vicari, Moreira e Menezes (2018), o algoritmo é um conjunto claro de instruções ou etapas que precisam ser seguidas para resolver o problema. Nesse sentido, as afirmações 10, 11 e 12 do questionário foram elaboradas com o objetivo de analisar esse pilar do Pensamento Computacional. O Gráfico 4 apresenta a distribuição das respostas dos estudantes para essas afirmações, permitindo visualizar como os participantes perceberam habilidades relacionadas à organização lógica de etapas, ao planejamento das soluções e à compreensão da lógica do projeto desenvolvido.

Gráfico 4 – Resultados do questionário sobre o pilar Algoritmo

Fonte: Elaborado pelo autor (2026)

Analisando o Gráfico 4, observa-se que a afirmação 10, relacionada à organização de uma sequência lógica de passos, apresentou predominância de respostas em “concordo totalmente”, indicando que a maioria dos alunos conseguiu estruturar corretamente o funcionamento do semáforo. Em relação à afirmação 11, que trata do planejamento dos passos antes da programação, os resultados apresentaram maior distribuição entre “concordo parcialmente” e “nem concordo nem discordo”, além da presença de respostas em “discordo parcialmente”. Esse resultado sugere que nem todos os alunos realizam o planejamento prévio de forma consistente, evidenciando uma dificuldade semelhante à observada no pilar Decomposição. Já a afirmação 12, referente à capacidade de explicar a lógica do funcionamento do sistema, apresentou predominância de respostas positivas, embora com menor intensidade em “concordo totalmente” quando comparada à afirmação 10. Isso indica que os alunos compreendem a lógica do projeto, mas ainda estão em processo de consolidação dessa compreensão.

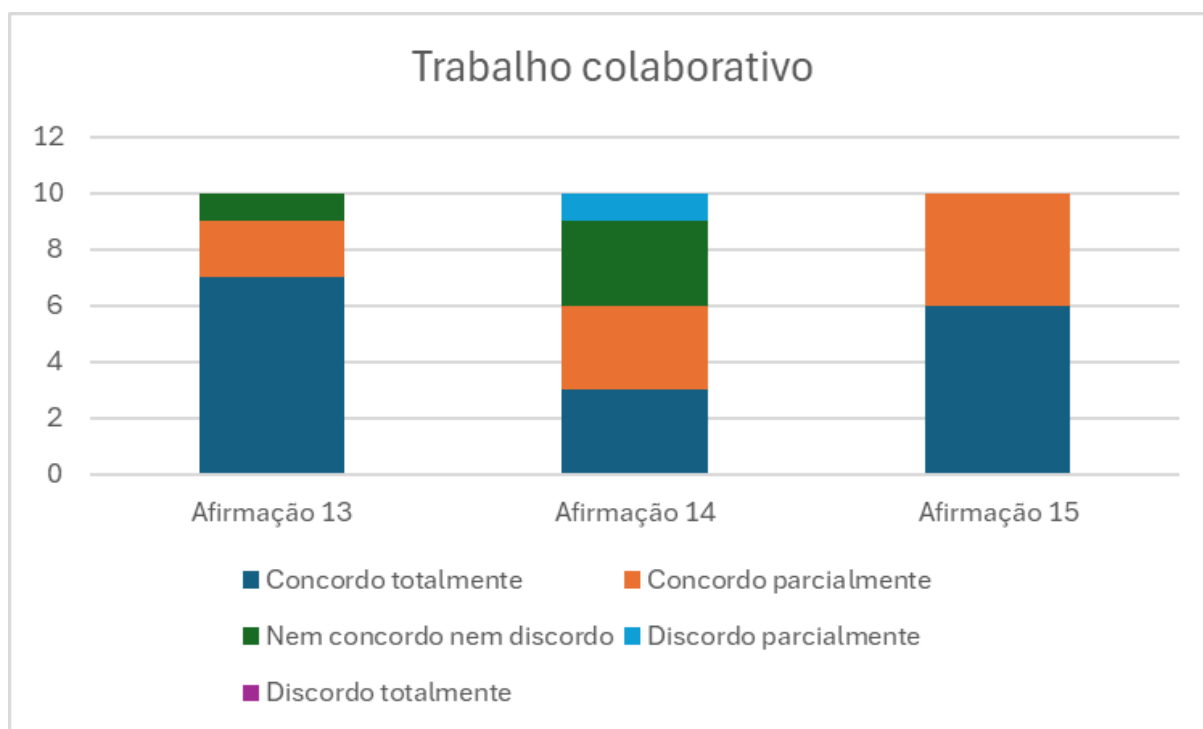
De modo geral, o pilar Algoritmos apresentou média de 4,17, indicando um bom nível de desenvolvimento dessa habilidade. Os resultados evidenciam que os alunos conseguem estruturar sequências lógicas e compreender o funcionamento do

sistema, embora ainda apresentem dificuldades no planejamento antecipado das soluções.

4.5 TRABALHO COLABORATIVO

Assim como proposto por Sutherland (2017), o EduScrum pode ser compreendido como uma ferramenta de ensino que favorece o trabalho colaborativo, a organização em equipe e o desenvolvimento de atividades de forma estruturada. Nesse contexto, as afirmações 13, 14 e 15 foram elaboradas com o objetivo de analisar a percepção dos alunos em relação ao trabalho em grupo durante o desenvolvimento do projeto. O Gráfico 5 apresenta a distribuição das respostas dos estudantes para essas afirmações, permitindo visualizar como os participantes perceberam aspectos relacionados à colaboração, organização das equipes e troca de ideias ao longo das atividades desenvolvidas.

Gráfico 5 – Resultados do questionário sobre o pilar Trabalho Colaborativo no contexto do EduScrum



Fonte: Elaborado pelo autor (2026)

Analizando o Gráfico 5, a afirmação 13, que aborda a contribuição do trabalho em grupo para a aprendizagem, apresentou predominância de respostas em “concordo totalmente” (7 respostas) e “concordo parcialmente” (2 respostas), com apenas uma resposta neutra. Esse resultado indica que a maioria dos alunos reconhece o trabalho colaborativo como um fator positivo para o aprendizado. Em relação à afirmação 14, que trata da organização do grupo, os resultados mostraram maior dispersão, com respostas distribuídas entre “concordo totalmente” (3), “concordo parcialmente” (3), “nem concordo nem discordo” (3) e “discordo parcialmente” (1). Essa variação sugere que, embora parte dos alunos perceba uma boa organização, outros ainda identificam dificuldades no processo de divisão de tarefas e gestão do grupo. Já a afirmação 15, relacionada à troca de ideias para resolução de problemas, apresentou resultados bastante positivos, com 6 respostas em “concordo totalmente” e 4 em “concordo parcialmente”. Esse dado evidencia que a interação entre os alunos contribuiu significativamente para a construção do conhecimento.

De modo geral, o conjunto das afirmações apresentou média de 4,33, indicando uma avaliação positiva do trabalho colaborativo pelos estudantes. Esses resultados reforçam a efetividade de abordagens colaborativas no ensino, sugerindo que a estrutura adotada ao longo do projeto favoreceu a aprendizagem ativa dos alunos, ainda que existam desafios relacionados à organização das equipes.

5 CONSIDERAÇÕES FINAIS

Conclui-se que os alunos apresentaram avanços nos quatro pilares do Pensamento Computacional (decomposição, reconhecimento de padrões, abstração e algoritmo) por meio das atividades propostas durante o curso, utilizando o Eduscrum como ferramenta de ensino. Os resultados evidenciaram maior desempenho nos pilares de decomposição e algoritmos, indicando que os alunos conseguiram dividir o problema em partes menores e organizar sequências lógicas para sua resolução. O reconhecimento de padrões também apresentou resultados positivos, especialmente na identificação de sequências repetitivas, embora tenha demonstrado limitações na análise de erros e na reutilização de soluções. Já o pilar abstração apresentou menor média, indicando dificuldades na identificação de elementos essenciais e na eliminação de informações irrelevantes.

No que se diz respeito ao trabalho colaborativo que está ligado ao uso da ferramenta do Eduscrum, os alunos avaliaram positivamente a aprendizagem em grupo, destacando a importância da troca de ideias para a resolução de problemas. No entanto, foram identificadas dificuldades na organização das equipes, evidenciando que esse aspecto ainda requer maior acompanhamento pedagógico. Ressalta-se que os alunos não foram informados sobre a utilização da metodologia adotada, permitindo uma análise mais espontânea das práticas vivenciadas.

Dessa forma, conclui-se que as estratégias utilizadas ao longo do projeto contribuíram para o desenvolvimento do Pensamento Computacional, indicando que a abordagem adotada foi adequada para esse contexto educacional. Por fim, sugere-se que trabalhos futuros aprofundem o desenvolvimento da abstração e ampliem o uso de metodologias ativas no ensino de Computação, visando potencializar ainda mais a aprendizagem dos estudantes além de desenvolver um material pedagógico que auxilie no ensino do Pensamento Computacional utilizando o EduScrum como ferramenta, com o objetivo de apoiar tanto professores quanto alunos no processo de ensino-aprendizagem.

REFERÊNCIAS

- ALBUQUERQUE, M. F. A. et al. **Robô eco-sustentável para aplicação em robótica educativa:** utilizando lixo tecnológico. *Cadernos de Graduação – Ciências Exatas e Tecnológicas*, Maceió, v. 3, n. 3, p. 215–228, 2016.
- ARAÚJO, K. F.; SILVA, T. da. **A inserção do pensamento computacional nos currículos do novo ensino médio no Brasil.** *Revista e-Curriculum*, v. 22, 2024.
- AZEVEDO, G. T. **Pensamento computacional e aprendizagem de matemática no ensino médio com uso de ferramentas tecnológicas.** *Revista Eletrônica de Educação Matemática*, v. 19, 2024.
- BELL, T.; WITTEN, I. H.; FELLOWS, M.; ADAMS, R.; MCKENZIE, J. **Computer Science Unplugged:** an enrichment and extension programme for primary-aged children. 3. ed. Christchurch: University of Canterbury, 2011.
- BITTENCOURT, I. I.; SILVA, T. M. **Programação e robótica na educação:** potencialidades para o desenvolvimento do pensamento computacional. In: *SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO*, 20., 2017.
- BORGES, K. S.; SCHMITT, M. A. R.; NAKLE, S. M. **EduScrum:** projetos de aprendizagem colaborativa baseados em Scrum. *Novas Tecnologias na Educação*, Porto Alegre, v. 12, n. 1, jul. 2014.
- BRASIL. **Ministério da Educação.** Portaria nº 1.432, de 4 de outubro de 2022. Brasília, 2022.
- CARVALHO, M. J. S.; PORTO, A. J. V.; BELHOT, R. V. **Aprendizagem significativa:** a teoria de Ausubel. São Carlos: Universidade de São Paulo, 2001.
- COLL, C.; POZO, J. I.; SARRAMONA, J.; VALLS, E. **Os conteúdos na reforma:** ensino e aprendizagem de conceitos, procedimentos e atitudes. Porto Alegre: Artmed, 2000.
- COSTA JÚNIOR, J. F. et al. **Um estudo sobre o uso da escala de Likert na coleta de dados qualitativos e sua correlação com as ferramentas estatísticas.** *Contribuciones a las Ciencias Sociales*, São José dos Pinhais, v. 17, n. 1, p. 360–376, 2024.
- DEGRACE, P.; STAHL, L. H. **Wicked problems, righteous solutions:** a catalogue of modern software engineering paradigms. Englewood Cliffs: Prentice-Hall, 1990.
- DELHIJ, D.; VAN SOLINGEN, R.; WIJNANDS, W. **EduScrum Guide.** 2016.
- GIL, A. C. **Métodos e técnicas de pesquisa social.** 6. ed. São Paulo: Atlas, 2017.

HUANG, X.; QIAO, C. **Enhancing computational thinking skills through artificial intelligence education at a STEAM high school.** Science & Education, 2022.

KELLEHER, C.; PAUSCH, R. **Lowering the barriers to programming:** a taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys, v. 37, n. 2, p. 83–137, 2005.

LOURENÇO, C.; PAIVA, M. **Planejamento e prática docente:** reflexões e ações. Belo Horizonte: UFMG, 2010.

MINAYO, M. C. S. **O desafio do conhecimento:** pesquisa qualitativa em saúde. 14. ed. São Paulo: Hucitec, 2014.

PRODANOV, C. C.; FREITAS, E. C. **Metodologia do trabalho científico:** métodos e técnicas da pesquisa e do trabalho acadêmico. 2. ed. Novo Hamburgo: Feevale, 2013.

REIS, C. A. S.; SARMENTO, H. R.; ZARAMELLA, V. **Ferramenta de auxílio ao desenvolvimento do pensamento computacional:** uma plataforma robótica controlada por smartphone. 2014. Trabalho de Conclusão de Curso (Graduação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2014.

RODRIGUES, S. S. **Computação desplugada no ensino fundamental I:** uma experiência metodológica numa escola pública na Paraíba. 2017. Trabalho de Conclusão de Curso (Licenciatura em Computação) – Universidade Federal da Paraíba, João Pessoa, 2017.

SCHWABER, K.; SUTHERLAND, J. **Guia do Scrum:** o guia definitivo para o Scrum: as regras do jogo. 2017.

SILVA, D. P. et al. **Aplicação de robótica na educação de forma gradual para o estímulo do pensamento computacional.** In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2016.

SINEK, S. **Comece pelo porquê:** como grandes líderes inspiram pessoas e equipes a agir. Rio de Janeiro: Sextante, 2018.

SUTHERLAND, J. **Scrum:** a arte de fazer o dobro do trabalho na metade do tempo. Rio de Janeiro: Leya, 2017.

TAKEUCHI, H.; NONAKA, I. **The new product development game.** Harvard Business Review, v. 64, n. 1, p. 137–146, 1986.

THIOLLENT, M. **Metodologia da pesquisa-ação.** São Paulo: Cortez, 1986.

VALENTE, J. A. **Pensamento computacional:** uma abordagem para a educação. São Paulo: Penso, 2017.

VICARI, R. M.; MOREIRA, A. F.; MENEZES, P. F. B. **Pensamento computacional: revisão bibliográfica.** 2018.

VIEIRA, K. D.; HAI, A. A. **O pensamento computacional na educação para um currículo integrado à cultura e ao mundo digital.** Acta Scientiarum. Education, v. 45, 2022.

WERNTROP, D.; WILENSKY, U. **The challenges of studying blocks-based programming environments.** In: IEEE BLOCKS AND BEYOND WORKSHOP, 2015.

WIJNANDS, W.; STOLZE, R. **EduScrum Guide.** 2019.

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

ZABALA, A. **A prática educativa: como ensinar.** Porto Alegre: Artmed, 1998.

ZANETTI, H.; OLIVEIRA, C. **Práticas de ensino de programação de computadores com robótica pedagógica e aplicação de pensamento computacional.** In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 2015.

APÊNDICE

QUESTIONÁRIO DE PESQUISA

Este questionário tem como objetivo compreender como foi seu aprendizado durante o desenvolvimento do projeto do Ensino de Pensamento Computacional. Não existem respostas certas ou erradas. Responda com sinceridade, de acordo com sua experiência ao longo do projeto. Suas respostas serão utilizadas apenas para fins de pesquisa.

1. Consegui dividir o projeto do semáforo em partes menores para facilitar sua execução.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente

2. Resolver uma parte do projeto por vez ajudou a entender melhor o funcionamento do semáforo.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente

3. Antes de começar a programar, pensei no projeto em etapas.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente

4. Percebi que o funcionamento do semáforo segue uma sequência que se repete.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente

5. Consegui identificar semelhanças entre erros que aconteceram durante o projeto.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente
6. Usei ideias ou soluções que já haviam sido utilizadas anteriormente.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente
7. Consegui identificar quais elementos eram mais importantes para o funcionamento do semáforo.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente
8. Ignorei detalhes que não eram essenciais para resolver os problemas principais.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente
9. Entendi o funcionamento geral do projeto mesmo sem conhecer todos os detalhes técnicos.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo
 - ☐ Concordo parcialmente
 - ☐ Concordo totalmente
10. Consegui organizar uma sequência lógica de passos para o semáforo funcionar corretamente.
 - ☐ Discordo totalmente
 - ☐ Discordo parcialmente
 - ☐ Nem concordo nem discordo

- Concordo parcialmente
- Concordo totalmente

11. Pensei nos passos do projeto antes de começar a programar.

- Discordo totalmente
- Discordo parcialmente
- Nem concordo nem discordo
- Concordo parcialmente
- Concordo totalmente

12. Eu conseguiria explicar a lógica do funcionamento do semáforo para outra pessoa.

- Discordo totalmente
- Discordo parcialmente
- Nem concordo nem discordo
- Concordo parcialmente
- Concordo totalmente

13. Trabalhar em grupo ajudou no meu aprendizado durante o projeto.

- Discordo totalmente
- Discordo parcialmente
- Nem concordo nem discordo
- Concordo parcialmente
- Concordo totalmente

14. O grupo conseguiu se organizar bem para realizar as tarefas do projeto.

- Discordo totalmente
- Discordo parcialmente
- Nem concordo nem discordo
- Concordo parcialmente
- Concordo totalmente

15. Conversar com colegas ajudou a encontrar soluções para os problemas enfrentados.

- Discordo totalmente
- Discordo parcialmente
- Nem concordo nem discordo
- Concordo parcialmente
- Concordo totalmente