

Milene Giaretta

A RELAÇÃO DO PENSAMENTO
COMPUTACIONAL E O DESENVOLVIMENTO
DO ARGUMENTO DEDUTIVO EM CRIANÇAS:
um estudo sobre a programação em bloco

Passo Fundo

2024

Milene Giaretta

**A RELAÇÃO DO PENSAMENTO COMPUTACIONAL E O
DESENVOLVIMENTO DO ARGUMENTO DEDUTIVO
EM CRIANÇAS: um estudo sobre a programação em bloco**

Tese apresentada ao Programa de Pós-Graduação em Educação, do Instituto de Humanidades, Ciências, Educação e Criatividade, da Universidade de Passo Fundo, como requisito parcial para a obtenção do grau de Doutor em Educação, nas diretrizes dos Processos Educativos e Linguagem, sob a orientação do Professor Dr. Adriano Canabarro Teixeira.

Passo Fundo

2024

G435r Giaretta, Milene

A relação do pensamento computacional e o desenvolvimento do argumento dedutivo em crianças [recurso eletrônico] : um estudo sobre a programação em bloco / Milene Giaretta. – 2024.

13 MB : PDF.

Orientador: Prof. Dr. Adriano Canabarro Teixeira.

Tese (Doutorado em Educação) – Universidade de Passo Fundo, 2024.

1. Tecnologia educacional. 2. Raciocínio. 3. Pensamento computacional. 4. Lógica. I. Teixeira, Adriano Canabarro, orientador. II. Título.

CDU: 37:004

Catalogação: Bibliotecária Juliana Langaro Silveira – CRB 10/2427

Milene Giaretta

A relação do pensamento computacional e o
desenvolvimento do argumento dedutivo em crianças: um
estudo sobre a programação em bloco

A banca examinadora abaixo, APROVA em 30 de agosto de 2024, a Tese apresentada ao Programa de Pós-Graduação em Educação da Universidade de Passo Fundo, como requisito parcial de exigência para obtenção de grau de Doutor em Educação, na linha de pesquisa Processos Educativos e Linguagem.

Dr. Adriano Canabarro Teixeira - Orientador
Universidade de Passo Fundo - UPF

Dr. Adão Caron Cambraia
Instituto Federal Farroupilha - IFFAR

Dra. Ana Maria de Oliveira Pereira
Universidade Federal da Fronteira Sul - UFFS

Dr. Juliano Tonezer da Silva
Universidade de Passo Fundo - UPF

Dr. Miguel da Silva Rossetto
Universidade de Passo Fundo - UPF

AGRADECIMENTOS

Agradeço primeiramente a Deus por ter me dado a oportunidade de estar aqui e por ter me iluminado e amparado ao longo deste caminho. Sua presença foi uma fonte constante de força e inspiração.

Agradeço à minha mãe, que sempre esteve ao meu lado, acreditando em mim e me apoiando incondicionalmente. Seu amor, compreensão e encorajamento foram fundamentais para superar os desafios. Você soube compreender minhas ausências, amparar os momentos de tristeza e angústia, e vibrar com as conquistas ao longo dessa jornada.

Agradeço ao meu pai, a estrela mais brilhante no meu céu, que ilumina meu caminho e celebra minhas conquistas a partir de onde está. Que você continue a brilhar intensamente, paizão. Sua filha está prestes a conquistar o doutorado, e a sua luz sempre foi meu guia.

Agradeço ao meu irmão André, que mesmo morando na Austrália e estando distante, sempre soube trazer palavras de alívio e conforto quando necessário. Seu apoio emocional foi uma importante fonte de força ao longo do doutorado.

Agradeço ao meu namorado Moisés, que chegou no fim do processo de doutorado, mas se mostrou uma presença constante e compreensiva. Sua paciência imperturbável e apoio foram essenciais para que eu continuasse acreditando em minha capacidade e força. Sua confiança em mim foi uma motivação constante.

Agradeço à minha afilhada Antonella, cuja chegada iluminou minha vida de maneira inesperada. Mesmo sendo pequena e sem compreender completamente o que estava acontecendo, sua presença trouxe uma renovação energia e força que foram fundamentais para concluir esta etapa.

Agradeço à direção e aos professores da Escola Estadual de Ensino Fundamental Mathias Lorenzon pela acolhida calorosa e pelo apoio durante a realização da pesquisa empírica. A colaboração foi essencial para o desenvolvimento e sucesso desta pesquisa.

Agradeço ao meu orientador Adriano Teixeira, que acompanhou cada passo deste processo com dedicação e compreensão. Sua orientação, paciência e apoio durante os momentos difíceis foram cruciais para a conclusão deste trabalho. Sou eternamente grata por seu empenho e orientação ao longo desses anos.

A todos que, de alguma forma, contribuíram para esta jornada, meu sincero agradecimento. Cada apoio, cada palavra de incentivo e cada gesto de compreensão foram essenciais para a realização deste trabalho.

"A verdadeira sabedoria está na capacidade de ver o familiar no novo e o novo no familiar."

Albert Einstein

RESUMO

O avanço tecnológico tem transformado profundamente a sociedade, especialmente no campo educacional, onde o uso de novas tecnologias é essencial para a disseminação e construção do conhecimento. Inovações tecnológicas integradas ao ambiente escolar têm o potencial de promover avanços significativos no processo de ensino e aprendizagem. Nesse contexto, o século XXI é marcado por mudanças nas relações humanas, impactando diretamente a educação. O pensamento computacional surge como uma habilidade relevante, pois concerne a uma capacidade cognitiva que abrange a análise de problemas complexos e a criação de soluções inovadoras. Paralelamente, a construção de argumentos dedutivos válidos é importante para o desenvolvimento do pensamento crítico e da comunicação eficaz, sendo uma competência relevante para a vida acadêmica e profissional. No entanto, a interação entre pensamento computacional e argumentação dedutiva ainda é pouco explorada. Esta pesquisa busca preencher essa lacuna, investigando como o pensamento computacional pode influenciar a capacidade de produzir argumentos válidos em alunos do Ensino Fundamental II. O objetivo principal é estabelecer conexões entre o pensamento computacional e a capacidade de desenvolvimento da argumentação dedutiva, utilizando um enfoque pedagógico que integre o ensino de programação de computadores, centrado na resolução de problemas, para alunos do Ensino Fundamental II. A metodologia adotada é qualitativa, com elementos de pesquisa observacional e estudo de caso. Ao longo da pesquisa, buscou-se explorar habilidades como abstração, decomposição e reconhecimento de padrões e algoritmo, características do pensamento computacional que contribuem para o desenvolvimento da argumentação dedutiva. Os resultados indicam que essas competências ajudam os alunos a identificar e organizar premissas de forma lógica, o que favorece a formulação de argumentos válidos. A pesquisa empírica revela que o pensamento computacional não apenas fortalece habilidades técnicas, mas também promove capacidades críticas de resolução de problemas e pensamento analítico, diretamente relacionadas à argumentação eficaz. No entanto, ainda há lacunas sobre como essas habilidades se desenvolvem em diferentes contextos educacionais e a longo prazo. Futuras pesquisas podem focar na adaptação dessa abordagem para outros níveis de ensino e no acompanhamento sobre o modo como essas competências impactam o desenvolvimento acadêmico e profissional ao longo do tempo.

PALAVRAS CHAVE: Educação, pensamento computacional, argumentos dedutivos, resolução de problemas, raciocínio lógico.

ABSTRACT

Technological advances have profoundly transformed society, especially in the educational field, where the use of new technologies is essential for the dissemination and construction of knowledge. Technological innovations integrated into the school environment have the potential to promote significant advances in the teaching and learning process. In this context, the 21st century is marked by changes in human relations, directly impacting education. Computational thinking emerges as a relevant skill, as it concerns a cognitive capacity that encompasses the analysis of complex problems and the creation of innovative solutions. At the same time, the construction of valid deductive arguments is important for the development of critical thinking and effective communication, being a relevant skill for academic and professional life. However, the interaction between computational thinking and deductive argumentation is still little explored. This research seeks to fill this gap by investigating how computational thinking can influence the ability to produce valid arguments in elementary school students. The main objective is to establish connections between computational thinking and the ability to develop deductive argumentation, using a pedagogical approach that integrates the teaching of computer programming, centered on problem-solving, for students in Elementary School II. The methodology adopted is qualitative, with elements of observational research and case study. Throughout the research, we sought to explore skills such as abstraction, decomposition, and pattern recognition and algorithm, characteristics of computational thinking that contribute to the development of deductive argumentation. The results indicate that these skills help students identify and organize premises in a logical way, which favors the formulation of valid arguments. Empirical research reveals that computational thinking not only strengthens technical skills, but also promotes critical problem-solving and analytical thinking capabilities, directly related to effective argumentation. However, there are still gaps in how these skills develop in different educational contexts and in the long term. Future research can focus on adapting this approach to other levels of education and monitoring how these skills impact academic and professional development over time.

KEY WORDS: Education, computational thinking, deductive arguments, problem solving, logical reasoning.

LISTA DE FIGURAS

Figura 1 - Representação das teses localizadas e distribuídas em áreas do conhecimento - CNPq.....	22
Figura 2 - Número de teses defendidas por estado brasileiro no período de 2016 a 2019.....	25
Figura 3 - Distribuição das teses defendidas em cada ano.....	26
Figura 4 - Exemplos de argumento dedutivo utilizando-se dos indicadores de inferência.....	57
Figura 5 - Pilares do pensamento computacional.....	73
Figura 6 - Estrutura metodológica da pesquisa.....	79
Figura 7 - Blocos de comandos utilizados para a programação no Scratch.....	86
Figura 8 - Blocos de comandos do Scratch e ordem lógica de uma programação.....	88
Figura 9 - Tela inicial do Scratch: áreas, botões, abas e elementos gráficos.....	88
Figura 10 - Blocos da função movimento no <i>software</i> Scratch.....	90
Figura 11 - Blocos da função aparência no <i>software</i> Scratch.....	90
Figura 12 - Blocos da função som no <i>software</i> Scratch.....	91
Figura 13 - Blocos da função eventos no <i>software</i> Scratch.....	92
Figura 14 - Blocos da função controle no <i>software</i> Scratch.....	92
Figura 15 - Blocos da função sensores no <i>software</i> Scratch.....	93
Figura 16 - Blocos da função operadores no <i>software</i> Scratch.....	94
Figura 17 - Blocos da função variáveis no <i>software</i> Scratch.....	94
Figura 18 - Extensões do <i>software</i> Scratch.....	95
Figura 19 - Esquema elaborado pelo aluno B.....	115
Figura 20 - Esquema elaborado pelo aluno B.....	116
Figura 21 - Esquema elaborado pelo aluno B.....	118
Figura 22 - Esquema elaborado pelo aluno B.....	119
Figura 23 - Esquema elaborado pelo aluno B.....	120
Figura 24 - Esquema elaborado pelo aluno B.....	122
Figura 25 - Esquema elaborado pelo aluno B.....	124
Figura 26 - Esquema elaborado pelo aluno B.....	126
Figura 27 - Esquema elaborado pelo aluno B.....	127
Figura 28 - Esquema elaborado pelo aluno B.....	129
Figura 29 - Desempenho dos alunos ao responder o teste de pensamento computacional.....	132
Figura 30 - Percentual de acertos dos alunos A e B ao responder o teste de pensamento computacional.....	132

LISTA DE QUADROS

Quadro 1 - Relação de Universidades e PPGs envolvidos nos 36 estudos localizados.....	23
Quadro 2 - Exemplo 1 de raciocínio dedutivo.....	37
Quadro 3 - Exemplo 2 de raciocínio dedutivo.....	37
Quadro 4 - Exemplos de raciocínio indutivo.....	38
Quadro 5 - Exemplos de raciocínio abdutivo.....	40
Quadro 6 - Exemplos de raciocínio analógico.....	41
Quadro 7 - Esquema inferencial de falácia.....	45
Quadro 8 - Exemplo considerando o esquema inferencial de falácia.....	46
Quadro 9 - Esquema inferencial de negação de falácia.....	46
Quadro 10 - Exemplo considerando o esquema inferencial de negação de falácia.....	46
Quadro 11 - Exemplo de argumento indutivo.....	48
Quadro 12 - Exemplo de argumento indutivo com o acréscimo de uma premissa universal...	48
Quadro 13 - Esquema inferencial ao silogismo estatístico (argumento indutivo).....	49
Quadro 14 - Exemplos aplicando o esquema de argumento indutivo estatístico.....	49
Quadro 15 - Versões 1 e 2 de esquema inferencial de argumento por analogia.....	51
Quadro 16 - Exemplo 1 aplicando o esquema de argumento por analogia.....	51
Quadro 17 - Versão 3 de esquema inferencial de argumento por analogia.....	52
Quadro 18 - Esquema das quatro formas válidas de argumento dedutivo.....	54
Quadro 19 - Exemplo de argumento dedutivo utilizando <i>Modus ponens</i> (MP).....	54
Quadro 20 - Indicadores de premissas em um argumento.....	56
Quadro 21 - Indicadores de conclusão em um argumento.....	56
Quadro 22 - Pilares do pensamento computacional abordados no teste.....	84
Quadro 23 - Proposta da atividade 1.....	98
Quadro 24 - Proposta da atividade 2.....	100
Quadro 25 - Proposta da atividade 3.....	101
Quadro 26 - Proposta da atividade 4.....	103
Quadro 27 - Proposta da atividade 5.....	104
Quadro 28 - Proposta da atividade 6.....	106
Quadro 29 - Proposta da atividade 7.....	108
Quadro 30 - Proposta da atividade 8.....	109
Quadro 31 - Proposta da atividade 9.....	111
Quadro 32 - Proposta da atividade 10.....	112

Quadro 33 - Erros e acertos de cada aluno no teste de pensamento computacional.....	135
Quadro 34 - Cálculos estatísticos.....	136
Quadro 35 - Esquema das quatro formas válidas de argumento dedutivo.....	137

SUMÁRIO

1 INTRODUÇÃO.....	12
2 RECORRENDO AO ESTADO DO CONHECIMENTO.....	18
2.1 Seleção dos descritores de busca.....	19
2.2 Caminho metodológico e seus resultados quantitativos.....	21
2.3 Explorando os estudos selecionados de forma qualitativa.....	26
2.4 Considerações parciais.....	30
3 RACIOCÍNIO LÓGICO E O PROCESSO CONSTRUTIVO DO ARGUMENTO.....	32
3.1 Raciocínio lógico e suas frentes.....	33
3.1.1 <i>Raciocínio dedutivo</i>	34
3.1.2 <i>Raciocínio indutivo</i>	36
3.1.3 <i>Raciocínio abdutivo</i>	38
3.1.4 <i>Raciocínio analógico</i>	39
3.2 Processo argumentativo.....	41
3.2.1 <i>Distinção entre validade e verdade - Falácias</i>	43
3.2.2 <i>Argumento indutivo</i>	46
3.2.3 <i>Argumento dedutivo</i>	52
3.3 Considerações parciais.....	57
4 PENSAMENTO COMPUTACIONAL.....	59
4.1 Explorando o pensamento computacional.....	61
4.1.1 <i>Pilares do pensamento computacional</i>	74
4.2 Considerações parciais.....	76
5 METODOLOGIA.....	78
5.1 Categorias de análise.....	79
5.2 Estrutura metodológica.....	79
5.3 Metodologia da pesquisa empírica.....	81
5.4 Scratch.....	86
5.5 Organização das atividades propostas.....	97
5.6 Considerações parciais.....	112
6 RESULTADOS DA PESQUISA EMPÍRICA.....	114
6.1 Observação participativa dos encontros.....	115
6.2 Teste de pensamento computacional.....	132
6.3 Questionário de argumento dedutivo.....	135
6.4 Considerações parciais.....	140
7 CONSIDERAÇÕES FINAIS.....	143
REFERÊNCIAS.....	148
ANEXO A - Questionário sobre argumento dedutivo.....	153
ANEXO B - Teste de pensamento computacional.....	167
ANEXO C - Termo de consentimento.....	180
ANEXO D - Termo de autorização da escola.....	182

1 INTRODUÇÃO

A motivação para este estudo surge da minha trajetória acadêmica e pessoal com a programação de computadores. Meu primeiro contato com essa área ocorreu em 2014, durante a disciplina "Internet como Recurso Didático", ministrada pelo Professor Adriano Canabarro Teixeira na minha graduação em Matemática, pela Universidade de Passo Fundo. O encantamento com a programação levou-me a aprofundar meus estudos, resultando na minha entrada no Programa de Pós-Graduação em Ensino de Ciências e Matemática em 2017, sob a orientação do mesmo professor. O tema do meu mestrado focou no pensamento computacional e raciocínio lógico. Dando continuidade a essa linha de pesquisa, iniciei o doutorado em Educação em 2019, também sob a orientação do Professor Adriano. Esta trajetória reflete um compromisso contínuo com a integração da programação de computadores, do pensamento computacional e do raciocínio lógico na educação.

Ao longo dos anos, o avanço tecnológico expandiu-se de maneira abrangente e veloz, deixando uma marca evidente em nossa sociedade. Esse progresso provocou diversas transformações na vida humana, manifestando-se por meio das mudanças nas interações sociais. O uso das tecnologias proporciona facilidades, inovações e alterações na forma como nos comunicamos, seja com indivíduos próximos ou distantes, durante momentos com amigos ou familiares, no contexto escolar e até mesmo no ambiente de trabalho. De acordo com Correia e Silva,

o uso das novas tecnologias não deve ser encarado como modismo ou com finalidade puramente comercial para aumentar o número de alunos, mas como ferramentas para a disseminação e construção do conhecimento. Uma das novas tecnologias que a escola tem procurado para estimular o aprendizado de seus alunos é o computador. (2005, p.12).

Essas inovações tecnológicas estão integralmente incorporadas ao cotidiano de estudantes e professores, proporcionando acesso a uma ampla gama de recursos didáticos modernos no ambiente escolar. Esse cenário visa promover um avanço significativo no processo de ensino e aprendizagem. Dessa forma, é plausível compreender que o século XXI está sendo caracterizado como a era da metamorfose nas relações humanas, sendo que os reflexos desse processo se evidenciam no âmbito educacional. Conforme a perspectiva de Toffler, o ambiente em que estamos imersos representa uma era de inovações, uma vez que

a alvorada desta nova civilização é o fato mais explosivo das nossas vidas. É o evento central a chave para compreender os anos imediatamente à frente. É um

evento tão profundo como a Primeira Onda de mudança, desencadeada há dez mil anos pela descoberta da agricultura, ou o terremoto da Segunda Onda de mudança, provocado pela revolução industrial. Somos os filhos da transformação seguinte, a Terceira Onda. (TOFFLER, 2014, p. 223).

A expressão "Terceira Onda" (TOFFLER, 2014), destacada pelo autor, denota as sucessivas transformações que nossa sociedade vem vivenciando, impulsionadas pelos contínuos avanços tecnológicos. Esses progressos motivam alternativas de comunicação e disseminação instantânea de informações, especialmente no contexto educacional. O avanço das tecnologias de informação não apenas possibilita, mas também incentiva a criação de ferramentas que têm potencial e devem ser integradas pelos educadores em sala de aula. Isso possibilita a disponibilização eficaz de informações, jogos e recursos para os alunos, promovendo um processo educativo dinâmico, eficiente e inovador, caracterizado pela troca de saberes entre professor e aluno, construção conjunta de conhecimento.

No âmbito educacional, os avanços tecnológicos estão intrinsecamente ligados à transformação e à capacidade criativa dos indivíduos, dando origem a modernos modelos de ensino e aprendizagem. Essas inovações proporcionam um ambiente propício para a evolução constante dos métodos educacionais, visto que “já não há um momento determinado em que qualquer pessoa possa dizer que não há mais o que aprender. Ao contrário, a sensação é a de que quanto mais se aprende mais há para estudar, para se atualizar” (KENSKI, 2011, p. 41).

No contexto educacional atual, caracterizado por uma paisagem em constante evolução impulsionada pela rápida marcha da tecnologia, os desafios que enfrentamos são cada vez mais complexos e multifacetados. A necessidade imediata de preparar os alunos com as habilidades necessárias para prosperar em um ambiente digitalmente enriquecido nunca foi tão importante. O pensamento computacional, nesse sentido, surge como uma resposta dinâmica e abrangente a essa demanda crescente. Mais do que apenas uma habilidade técnica isolada, o pensamento computacional representa um conjunto de capacidades cognitivas fundamentais, indo além da simples codificação para envolver a análise de problemas complexos, a criação de soluções inovadoras e a compreensão dos sistemas digitais em um nível mais profundo, visto que

[...] na sociedade de hoje, as tecnologias digitais são um símbolo de possibilidade e progresso. Quando as crianças aprendem a usá-las para se expressar e compartilhar ideias por meio da programação, elas começam a se ver de novas formas e a enxergar a possibilidade de contribuir ativamente para a sociedade. Elas começam a se ver como parte do futuro. [...] começam a se perceber como criadores, desenvolvendo confiança e orgulho em sua capacidade de criar coisas e de se expressar fluentemente com novas tecnologias (RESNICK, 2020, p. 47).

Nesse contexto, Papert percebeu que os computadores podem ser utilizados não apenas para fornecer informações e instruções, mas também para permitir que as crianças experimentem, explorem e se expressem de maneira mais ativa (MIT, 2016). Ele foi pioneiro ao introduzir a ideia de que a programação de computadores e a depuração oferecem às crianças uma oportunidade de refletir sobre seu próprio pensamento e compreender melhor seu processo de aprendizado (MIT, 2016).

Ao incorporar o pensamento computacional em seu conjunto de habilidades, os alunos não apenas adquirem a capacidade de programar, mas também desenvolvem uma mentalidade analítica e uma abordagem sistemática para resolver uma variedade de desafios. Para Fávero et al. (2017) “a resolução de problemas enquanto habilidade básica deve ser vista como uma competência mínima para que os estudantes possam inserir-se no mundo do conhecimento e a respeito dele produzir uma leitura crítica” (p. 4).

Isso os capacita não apenas a se adaptarem às rápidas mudanças tecnológicas, mas também a liderar e moldar o futuro digital com confiança e criatividade. Além disso, ao abordar problemas de maneira lógica e criativa, os alunos cultivam uma mentalidade de resolução de problemas que é importante não apenas no mundo da tecnologia, mas em todas as esferas da vida, e

ao aprender a programar, você também se torna um pensador melhor. Por exemplo, você aprende a dividir problemas complexos em partes mais simples e a depurá-los, e a refinar melhor os projetos por meio de repetições ao longo do tempo. [...] Ao aprender a depurar programas de computador, você estará mais bem preparado para descobrir o que deu errado se preparar uma receita que não der certo na cozinha ou caso você se perca ao seguir as orientações de direção dadas por outra pessoa (RESNICK, 2020, p. 45).

Por outro lado, a construção de argumentos válidos desempenha um papel importante no desenvolvimento do pensamento crítico e na comunicação eficaz. Fávero et al. (2017, p. 92) nos coloca que “argumento é um recurso escrito ou falado para negar ou afirmar fatos por meio da razão, do raciocínio, para convencer alguém sobre algo, alterando-lhe a opinião ou o comportamento.”

Neste viés, classifica-se como bom argumento

[...] aquele que sustenta efetivamente uma conclusão, ou seja, aquele que apresenta os elementos que corroboram de fato uma ideia. Vale salientar que argumentar não é simplesmente ter uma opinião sobre algo. É preciso que as evidências tenham ligação direta com a conclusão para fundamentá-las. Caso contrário, não terão validade lógica (FÁVERO et al., 2017, p. 3-4).

A habilidade de articular ideias de maneira clara e persuasiva, apoiadas por evidências sólidas e raciocínio coerente, é uma competência essencial em todas as áreas da vida acadêmica e profissional. Ao aprender a construir argumentos sólidos, os alunos não apenas refinam suas habilidades de comunicação, mas também fortalecem sua capacidade de análise e síntese de informações, tornando-se pensadores mais perspicazes e participantes ativos no processo de aprendizagem e na sociedade em geral. (SACRINI, 2016).

Assim, ao combinar o pensamento computacional com a construção de argumento dedutivo, os educadores podem oferecer aos alunos uma base sólida para navegar no mundo digital em constante mudança com confiança e habilidade. Essas habilidades não apenas os capacitam a prosperar em suas carreiras futuras, mas também a se tornarem cidadãos informados e engajados em um mundo cada vez mais complexo e interconectado.

O contexto educacional atual enfrenta desafios complexos diante da rápida evolução tecnológica e da necessidade de preparar os alunos para um futuro cada vez mais digital. Nesse cenário, o pensamento computacional emerge como um conjunto essencial de habilidades cognitivas, transcendendo a mera codificação e abraçando a capacidade de abordar problemas complexos de maneira lógica e criativa. Simultaneamente, a construção de argumentos válidos torna-se uma habilidade importante para o desenvolvimento do pensamento crítico, permitindo aos estudantes comunicar ideias de forma coerente e persuasiva.

No entanto, o entendimento profundo da interseção entre o pensamento computacional e a capacidade de produzir argumentos válidos, especialmente entre alunos, permanece uma lacuna em grande parte inexplorada. Esta pesquisa visa preencher essa lacuna, buscando responder à pergunta central: qual o papel do pensamento computacional na capacidade de produzir argumentos válidos? Para esclarecer a questão central, busca-se responder aos seguintes questionamentos secundários: O que já foi produzido na área de educação relacionando pensamento computacional e argumentação dedutiva? Qual o vínculo existente entre raciocínio lógico e argumentação dedutiva? Quais as articulações necessárias para ocorrer conexão entre as premissas e a conclusão na construção de um argumento dedutivo? Quais as conexões do pensamento computacional na formação de um ser argumentativo que resolve problemas quando programa computadores?

O objetivo geral é estabelecer conexões entre o pensamento computacional e a capacidade de desenvolvimento da argumentação dedutiva, utilizando um enfoque pedagógico que integre o ensino de programação de computadores, centrado na resolução de problemas, para alunos do Ensino Fundamental II.

Os objetivos específicos desdobram-se em quatro camadas distintas: (1) Investigar pesquisas que articulam o desenvolvimento do pensamento computacional e a construção de argumentos dedutivos. (2) Compreender o papel do raciocínio lógico no processo de construção do argumento dedutivo. (3) Identificar articulações entre o desenvolvimento do pensamento computacional e a construção de argumentos dedutivos. (4) Analisar os desdobramentos do pensamento computacional no processo de complexificação da capacidade de argumentação.

Para atingir esse objetivo, optamos por uma abordagem metodológica qualitativa, complementada com pesquisa de observação e estudo de caso, com alunos do Ensino Fundamental II da Escola Estadual de Ensino Fundamental Mathias Lorenzon, localizada no município de Getúlio Vargas-RS, que estão envolvidos no projeto Conect +. Este projeto, voltado para o ensino de programação com foco na resolução de problemas, proporciona um espaço para observar como o pensamento computacional pode influenciar a capacidade de construir argumentos dedutivos válidos. A escolha dessa metodologia qualitativa permite uma análise mais detalhada das interações entre alunos e as ferramentas de programação, assim como a maneira como esses estudantes aplicam habilidades cognitivas, como abstração, decomposição e reconhecimento de padrões, na formulação de argumentos.

A metodologia não só possibilitou investigar como essas competências se desenvolvem em tempo real, mas também permitiu a identificação de lacunas no processo de ensino e aprendizagem, abrindo espaço para a melhoria das práticas educacionais e sugerindo novos caminhos para pesquisas futuras.

A estrutura desta pesquisa seguiu uma sequência lógica em cinco capítulos. O primeiro aborda o estado do conhecimento, delineando as tendências existentes e apontando para as áreas que requerem uma compreensão mais aprofundada. O segundo capítulo se concentra em uma análise teórica, buscando dados que esclareçam a relação entre raciocínio lógico e argumentação dedutiva, enquanto o terceiro capítulo promove uma compreensão das relações entre pensamento computacional, raciocínio lógico e argumentação dedutiva. O quarto capítulo explora o viés metodológico, por meio de uma abordagem qualitativa, utilizando a observação direta e o estudo de caso. Por fim, o quinto capítulo apresenta os resultados da pesquisa, analisando as descobertas empíricas e organizando-as de maneira a responder à questão central e às hipóteses levantadas ao longo do estudo.

Ao concluir este estudo, almejamos não apenas enriquecer o conhecimento teórico sobre a ligação entre pensamento computacional e argumentação dedutiva, mas também proporcionar conhecimentos valiosos para a prática educativa. Nosso objetivo é oferecer uma

visão mais clara de como essas habilidades cognitivas podem ser integradas no ambiente escolar, promovendo um aprendizado mais eficaz e adaptado às demandas do século XXI. Ao revelar as dinâmicas que conectam o pensamento computacional à construção de argumentos válidos, esperamos inspirar novas abordagens pedagógicas que estimulem o desenvolvimento crítico e criativo dos alunos. Com isso, visamos contribuir para a formação de cidadãos capacitados para enfrentar desafios complexos e se destacar em um mundo digital em constante evolução.

2 RECORRENDO AO ESTADO DO CONHECIMENTO

Diante do processo de construção de uma tese, o pesquisador precisa considerar e escolher ideias que se aproximam aos seus objetivos. Essas escolhas refletem os significados coletados nas esferas que trilhamos. Minayo (1996) nos coloca que fazer uma pesquisa exige fixar-se em uma área de interesse, onde se junta a curiosidade, para determinar o objetivo e o problema que será investigado. Assim sendo, ocorre um recorte das possibilidades existentes na temática selecionada e evidenciadas por outros autores, visto que

é necessário “descobrir” algo que ainda não foi dito por eles. Quando se fala em “descoberta”, em especial no campo humanista, não cogitamos de invenções revolucionárias como a descoberta da fissão do átomo, a teoria da relatividade ou uma vacina contra o câncer: podem ser descobertas mais modestas, considerando-se resultado “científico” até mesmo uma maneira nova de ler e entender um texto clássico, a identificação de um manuscrito que lança nova luz sobre a biografia de um autor, uma reorganização e releitura de estudos precedentes que conduzem à maturação e sistematização das ideias que se encontravam dispersas em outros textos. Em qualquer caso, o estudioso deve produzir um trabalho que, teoricamente, os outros estudiosos do ramo não deveriam ignorar, porquanto diz algo de novo sobre o assunto. (ECO, 2016, p. 2-3).

Para produzir uma pesquisa de âmbito acadêmico, é preciso inteirar-se com o que tem sido elaborado sobre a temática escolhida para o estudo, com a finalidade de avançar com as investigações e não debruçar-se em pontos sem relevância para o campo científico, ou que já foram feitas. Segundo Romanowski e Ens,

esses estudos são justificados por possibilitarem uma visão geral do que vem sendo produzido na área e uma ordenação que permite aos interessados perceberem a evolução das pesquisas na área, bem como suas características e foco, além de identificar as lacunas existentes. (20016, p.41).

Estes estudos são chamados de “estado da arte” e o seu objetivo “é descrever o estado atual de uma dada área de pesquisa: o que já se sabe, quais as principais lacunas, onde se encontram os principais entraves teóricos e/ou metodológicos” (LUNA, 2019, p. 87-88). Portanto, para fazer um “estado da arte” é necessário fazer uma análise de todo material coletado através das dissertações, teses, artigos de periódicos, congressos, seminários e eventos da área pesquisada. Entretanto, é possível fazer pesquisas delimitadas, ou seja, estudar apenas uma fonte de publicação, onde uma temática é analisada em apenas uma fonte de conteúdo, produzida em um determinado local e em um intervalo de tempo delimitado

através de anos. Estas pesquisas são denominadas de “estado do conhecimento” (ROMANOWSKI; ENS, 2006).

Tanto o “estado da arte” quanto o “estado do conhecimento”, conservam como objetivo basilar a “análise da produção realizada para, então, compreender como o tema está sendo tratado, quais das tendências teóricas estão envolvidas e quais vertentes metodológicas foram utilizadas, gerando, assim, mais conhecimento na área” (LESSA, 2018, p. 24). Atualmente, as pesquisas têm feito essa sondagem e análise de trabalhos sobre a temática estudada, para fortalecer a relevância da pesquisa; e auxiliar o pesquisador na estruturação do problema de pesquisa, referencial teórico e metodológico.

Nesta perspectiva, elaboramos um “estado do conhecimento”, para responder o seguinte questionamento: O que já foi produzido na área de educação relacionando pensamento computacional e argumentação dedutiva? Tendo como objetivo, investigar pesquisas que articulam o desenvolvimento do pensamento computacional e a construção de argumentos dedutivos, que mapeou estudos advindos de teses que trataram o pensamento computacional e o desenvolvimento da argumentação dedutiva.

Assim, teremos a oportunidade de explorar brevemente os conceitos utilizados como descritores de busca para a pesquisa, além de nos aproximar, conhecer e considerar o que tem sido pesquisado sobre a temática, a quantidade de pesquisas que já foram feitas unindo as duas áreas de relevância para este estudo, qual o problema de partida, perspectiva teórica e metodológica, os resultados e analisar os caminhos que podemos percorrer. Visto que “[...] se trata efetivamente de pesquisa original, onde é necessário conhecer a fundo o quanto foi dito sobre o mesmo argumento pelos demais estudiosos” (ECO, 2016, p. 2).

2.1 Seleção dos descritores de busca

Para fazer uma busca no Catálogo de Teses e Dissertações¹ da Capes², é necessário inserir termos. Estes termos são pesquisados em todo banco de dados e trata-se de elementos contidos no “Painel de informações quantitativas (teses e dissertações)”. Desta forma, selecionamos quatro descritores, sendo: "pensamento computacional", "programação de computadores", "raciocínio lógico" e "argumento dedutivo", com base no projeto de pesquisa de tese.

¹ Acesso disponível em <<https://catalogodeteses.capes.gov.br/catalogo-teses/#!/>>.

² Capes é a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior. Trata-se de uma fundação vinculada ao Ministério da Educação do Brasil que atua na expansão e consolidação da pós-graduação stricto sensu em todos os estados brasileiros.

Optamos pelo pensamento computacional ao perceber que a tecnologia tem se mostrado presente cada vez mais cedo na vida das crianças. É preciso que ela seja incluída e utilizada a favor de alunos e professores também na educação. É a tecnologia que compõe a base do pensamento computacional, um método que possibilita a criança ou o adulto a conseguir soluções para problemas de maneira lógica, eficaz e criativa a partir de fundamentos da computação. (WING, 2006).

O mundo moderno exige mudanças de pensamento em relação à educação e à tomada de decisões. Ser um cidadão mais crítico, lógico, que sabe trabalhar em equipe e resolver problemas que surgem no dia a dia é fundamental para sobreviver em um mundo competitivo e repleto de desafios na vida pessoal e profissional.

Para Papert (1971) a programação de computadores, também utilizada atualmente como ferramenta de aprendizagem, envolve o contato com novas linguagens e também com um novo modo de pensar, como o exemplo das habilidades de abstração e análise. Quando as crianças começam a aprender programação, um novo olhar, mais participativo e inovador, começa a se formar e os aparelhos tecnológicos ganham um novo significado na vida delas.

Ainda em Papert (1971), evidencia-se que o ato de programar consiste em solucionar problemas utilizando o computador e sua capacidade de processamento de dados e informações como aliados. No universo da computação, programar significa criar soluções – como jogos, animações e aplicativos digitais – para resolver problemas diversos. Ao utilizar a programação como uma prática pedagógica, um dos resultados é o desenvolvimento do pensamento computacional.

A escolha pelo raciocínio lógico está atrelada ao ato de pensar e às diversas facetas do termo, mas principalmente aos processos mentais que irão articular as informações recebidas, com o objetivo de encontrar soluções e possíveis esclarecimentos ao que foi apresentado (resolução de problemas), a fim de construir significados (PIAGET, 1978). A noção de lógica está ligada a duas vertentes: o uso de raciocínio na execução de atividades e o estudo filosófico do raciocínio válido. Já o termo raciocínio é caracterizado como uma operação lógica discursiva e mental, a qual é necessária para organizar dados (imagens, palavras ou números) através do uso de premissas a fim de alcançar uma conclusão (ARISTÓTELES, 2016).

Dentro deste viés do raciocínio lógico surgem os argumentos. Este possui formas próprias capazes de evidenciar que uma conclusão é derivada daquilo que fora estabelecido nas premissas ou proposições dadas anteriormente. Há quatro formas de se proceder quando se pretende formar uma argumentação, são elas: argumento de autoridade, argumento por

analogia, argumento indutivo e o argumento dedutivo. Este último será foco de nossa busca, visto que dentro da dedução, todos os termos envolvidos nas premissas estabelecem relações que podem ser encontradas na conclusão.

Neste viés, argumento dedutivo é aquele que procede de proposições cada vez mais universais para proposições particulares, proporcionando o que chamamos de demonstração, pois que sua inferência (a conclusão é extraída das premissas) é a inclusão de um termo menos extenso em outro de maior extensão (WESTON, 2009).

Diante destes conceitos, busca-se a utilização das ferramentas contidas no pensamento computacional para aprendizagem da programação, considerando seus inúmeros benefícios, pois ao programar lidamos com alguns desafios mentais, que nos forçam a pensar de forma abstrata, sequencial e organizada. Assim como jogar xadrez ou estudar um instrumento musical cria novos padrões de conexões neuronais, programar também estimula partes do cérebro relacionadas à lógica e ao próprio raciocínio lógico. Portanto, este processo de raciocínio lógico deverá ser explicado e esmiuçado através do argumento dedutivo, ou seja, expor suas premissas particulares do processo de raciocínio lógico, explicar seus testes, condições e verdades, para alcançar uma conclusão.

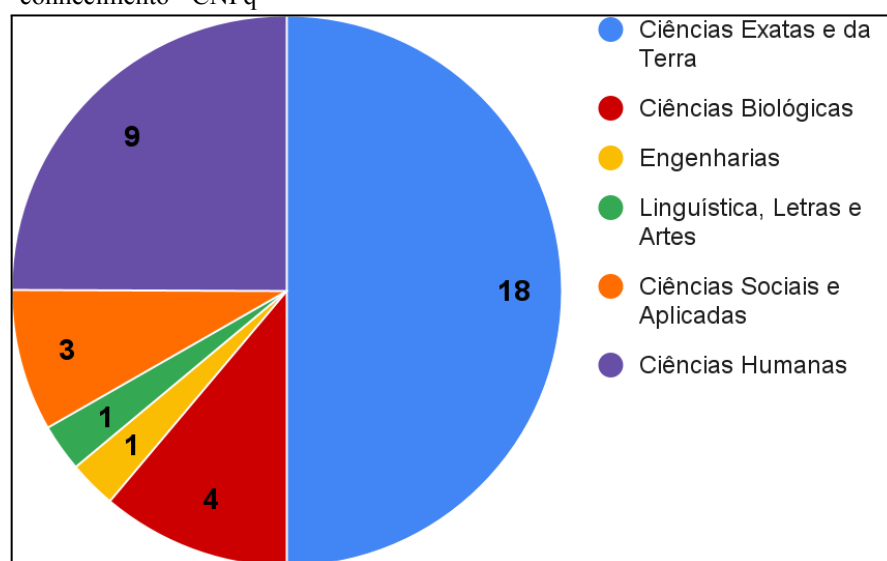
2.2 Caminho metodológico e seus resultados quantitativos

Esta investigação foi feita de forma coletiva, visto que foi realizada com a contribuição de bolsistas de iniciação científica do Grupo de Pesquisa em Cultura Digital (GEPID), da Universidade de Passo Fundo. Assim, coletamos informações de teses contidas na plataforma digital, a partir dos descritores estabelecidos e o recorte de tempo determinado, junto ao repositório brasileiro, denominado Catálogo de Teses e Dissertações da Capes. Desta forma, acessamos e coletamos dados de cada estudo, analisando-os no aspecto quantitativo e qualitativo, para adquirir condições de delimitar o tema e fazer as escolhas teóricas e metodológicas desta tese de doutorado.

Estas buscas foram realizadas com os descritores "pensamento computacional", "argumento dedutivo", "raciocínio lógico" e "programação de computadores", aplicando primeiramente filtros para teses no período de 2016 a 2019. Utilizamos quatro descritores ao perceber que em buscas separadas sobre "pensamento computacional" e "argumento dedutivo", haviam alguns trabalhos que se repetiam, além de trazer uma ampla gama de estudos e distintas áreas, pois alguns trabalhos fazem referência somente a alguns aspectos do pensamento computacional, como complemento de outra área; e outros reportam-se ao

argumento dedutivo, como forma de demonstrar proposições jurídicas, da área de ciências humanas e sociais, por exemplo. Com o intuito de diminuir a diversidade de teses e colher dados educacionais, com vistas ao raciocínio lógico, adicionamos os descritores "raciocínio lógico" e "programação de computadores". Tal variedade de áreas que abordam esses conceitos fica evidenciado na Figura 1.

Figura 1 - Representação das teses localizadas e distribuídas em áreas do conhecimento - CNPq



Fonte: Autora, 2020.

As pesquisas que utilizam-se dos conceitos de pensamento computacional e/ou argumento dedutivo têm sido desenvolvidas em diferentes áreas do conhecimento, em distintos programas, entretanto, buscamos estudos que empenharam-se com a educação e que atendiam as demandas do processo de ensino e aprendizagem, com vistas para o desenvolvimento do raciocínio lógico, envolvendo a informática educativa.

Para organizar os dados coletados, inserimos elementos encontrados em uma planilha compartilhada, onde colocamos o link de acesso para cada tese, autor, data da defesa, título, área do conhecimento (ciência da computação, direito, educação, engenharia, psicologia, etc.), instituição de ensino, local (cidade e estado) e se utilizou algum *software* no decorrer do trabalho (recurso tecnológico), questões de pesquisa, objetivos, metodologia e resultados. Após preencher as informações, as produções foram examinadas quanti e qualitativamente, assim selecionamos dois estudos, que vem ao encontro da temática tratada nesta tese.

Nesta busca por teses que de alguma maneira articulam o pensamento computacional e o argumento dedutivo, foram encontrados 36 estudos, a partir dos descritores já citados. Tais estudos estão relacionados no Quadro 1, e são advindos de 30 programas de Pós-Graduação,

oriundos de 24 universidades, com destaque a Universidade Federal do Rio Grande do Sul (UFRGS) com sete teses, sendo seis do Programa de Pós-Graduação em Informática na Educação (PGIE) e uma do Programa de Pós-Graduação em Computação (PPGC). A Pontifícia Universidade Católica de São Paulo (PUC-SP) aparece com quatro teses, sendo duas do PPG em Educação, uma do PPG em Psicologia e uma do PPG em Tecnologias da Inteligência e Design Digital. Há um pequeno realce da Universidade Federal da Bahia (UFBA) e da Universidade Federal de São Paulo (UNIFESP) em relação ao restante, com duas teses cada. Sendo uma tese do PPG em Engenharia Industrial e uma do PPG em Ensino, Filosofia e História das Ciências na UFBA; uma tese do PPG em Psicobiologia e outra do PPG em Psiquiatria e Psicologia Médica na UNIFESP.

Quadro 1 - Relação de Universidades e PPGs envolvidos nos 36 estudos localizados

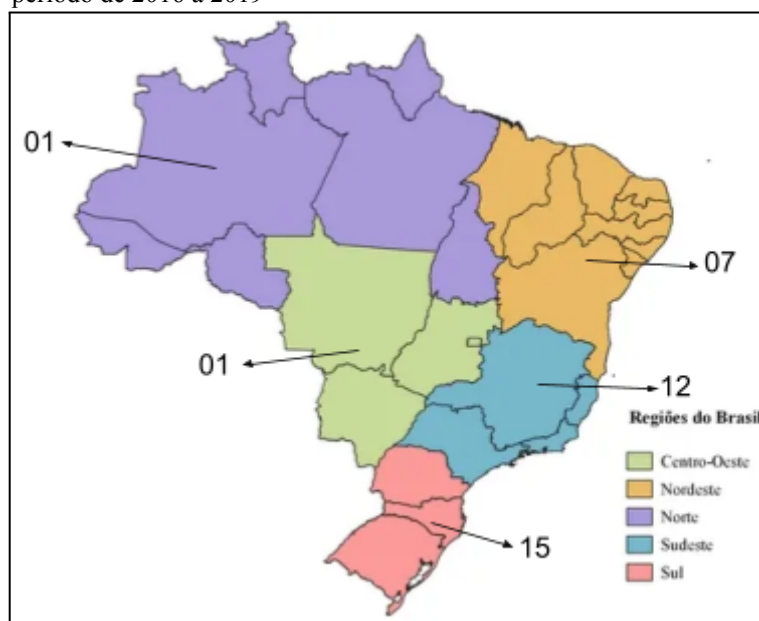
UNIVERSIDADE	PROGRAMA DE PÓS-GRADUAÇÃO	Nº DE TESES
FURG	Educação em Ciência Química da Vida e Saúde	1
PUC-MG	Geografia	1
PUC-RS	Ciência da Computação	1
PUC-SP	Tecnologias da Inteligência e Design Digital	1
PUC-SP	Psicologia	1
PUC-SP	Educação	2
UEM	Educação para a Ciência e a Matemática	1
UENF	Cognição e Linguagem	1
UFAL	Dinâmica do Espaço Habitado	1
UFAM	Informática	1
UFBA	Engenharia Industrial	1
UFBA	Ensino, Filosofia e História das Ciências	1
UFCG	Ciência da Computação	1
UFPE	Ciências da Computação	1
UFPR	Direito	1
UFRGS	Informática na Educação	6
UFRGS	Computação	1
UFRN	Sistemas e Computação	1
UFSC	Educação Científica e Tecnológica	1

UMESP	Comunicação Social	1
UNB	Psicologia do Desenvolvimento Escolar	1
UNEB	Educação e Contemporaneidade	1
UNESP	Educação para a Ciência	1
UNICAMP	Multiunidades em Ensino de Ciências e Matemática	1
UNIFESP	Psiquiatria e Psicologia Médica	1
UNIFESP	Psicobiologia	1
UNIOESTE	Letras	1
UPF	Educação	1
USP	Ciências da Reabilitação	1
UTFPR	Ensino de Ciência e Tecnologia	1

Fonte: Autora, 2020.

Estes trabalhos derivam de parte dos estados brasileiros, concentrando a sua maioria no estado do Rio Grande do Sul e São Paulo. A região Sul (vermelho) totaliza 15 estudos, a região Sudeste (azul) aparece com 12 estudos, a região Nordeste (amarelo) apresenta 7 estudos e as regiões Norte (lilás) e Centro Oeste (verde) tem 1 estudo cada uma, conforme representado na Figura 2.

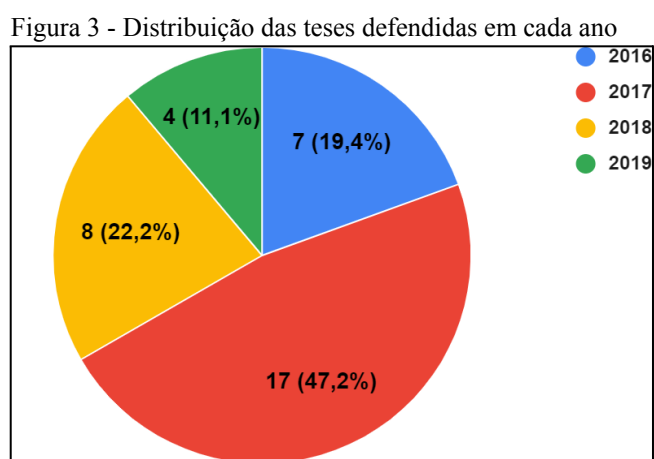
Figura 2 - Número de teses defendidas por estado brasileiro no período de 2016 a 2019



Fonte: Mapa brasileiro adaptado pela autora, 2020.

Ao filtrar a coluna de “data da defesa” percebemos que existe uma certa discrepância, dado que o ano de 2017 obteve o maior número de defesas, sendo 17 dos 36 estudos localizados, representando 47,2% do total. Nos anos de 2016 e 2018 uma relativa igualdade de defesas, e uma significativa queda de defesas nesta área em 2019, conforme a representação gráfica da Figura 3. Vale ressaltar que coincidentemente, os dois estudos selecionados para a construção deste estado do conhecimento, foram defendidos no ano de 2017, representando a totalidade de 100%.

Acredita-se que os números de 2017 relacionam-se com a Base Nacional Comum Curricular (BNCC), que começou a ser elaborada em 2015, a partir de uma análise aprofundada de documentos curriculares brasileiros feita por 116 profissionais de educação. Nos anos de 2015 e 2016 foram realizadas consultas públicas para possibilitar a contribuição direta da população na estruturação da BNCC. Em 2017, considerando todas as versões do documento, o MEC concluiu a sistematização e estabeleceu que os sistemas e redes de ensino do país teriam até 2020 para implementar a BNCC, que contempla o desenvolvimento de competências e habilidades relacionadas ao uso analítico e consciente das tecnologias digitais presentes em todas as áreas do conhecimento (BRASIL, 2017).



Fonte: Autora, 2020.

Outro ponto levantado foi a utilização de algum *software* no decorrer dos estudos selecionados, em virtude de estarmos vivenciando um imenso avanço tecnológico, que pode ser evidenciado na construção das teses analisadas, visto que este recurso está sendo mais explorado e utilizado, auxiliando nas questões metodológicas, coleta de dados ou análise de dados. Em geral, as tecnologias digitais utilizadas tinham cunho educacional, objetos de aprendizagem on-line, plataformas para cursos a distância, aplicativos de redes sociais e sites.

Dentre os 36 estudos separados para esta observação, 61,1% deles utilizou recursos tecnológicos, seja para exploração, juízo ou decomposição de dados, auxiliando nas conclusões do trabalho. Destes 22 trabalhos que exploraram de alguma forma os recursos tecnológicos, utilizando 32 *Softwares* diferentes, apenas dez trabalhos recorreram à programação de computadores. Os *Softwares* mais usados foram: Scratch, Tinkercad, Rstudio e Mplus - O Scratch foi explorado em seis trabalhos; o Tinkercad, Rstudio e Mplus foram aproveitados em duas teses diferentes cada um deles.

Para a análise qualitativa, fizemos a leitura completa dos dois trabalhos, visando informar-se sobre o que e como foi elaborado por Christian Puhlmann Brackmann (2017) e Rafael Marimon Boucinha (2017). Ambas as teses foram produzidas junto ao PGIE da UFRGS, na linha de pesquisa Paradigmas para a Pesquisa sobre o Ensino Científico e Tecnológico, e orientadas pelo professor Dante Augusto Couto Barone.

2.3 Explorando os estudos selecionados de forma qualitativa

Conhecer o que foi explorado por Boucinha (2017) e Brackmann (2017), a maneira como atuaram e seus resultados trouxe muitas contribuições para o desenvolvimento desta pesquisa de doutorado, visto que uma tese exige do autor uma cooperação com a área de pesquisa, além de dizer algo que ainda não foi dito. Segundo Eco (2016), quem escreve deve aproveitar a experiência que adquiriu com escritos anteriores e produzir algo novo que outros estudiosos não deveriam jamais ignorar. É preciso, assim, conhecer para avançar. Desta forma, exploramos as duas teses em pontos chave: o problema de pesquisa e os objetivos, referencial teórico, a metodologia e os resultados.

Na sua tese de doutorado, intitulada “Aprendizagem do Pensamento Computacional e Desenvolvimento do Raciocínio”, Boucinha (2017, p. 12) estrutura sua pesquisa através do seguinte questionamento: “Qual a relação entre a construção do Pensamento Computacional e o desenvolvimento do raciocínio?”. Através desta pergunta principal, o autor ramificou outros questionamentos: “Como ensinar Pensamento Computacional? Como avaliar o PC dos alunos? Quais efeitos do PC na capacidade de raciocinar? Quais tipos de raciocínio estão relacionados ao PC? O Pensamento Computacional é um fator independente ou expressa a capacidade de raciocínio?”. Seu objetivo principal foi “investigar a relação entre a construção do Pensamento Computacional e o desenvolvimento do raciocínio” (BOUCINHA, 2017, p. 13), sub dividindo em quatro objetivos específicos:

1. Desenvolver metodologia para ensino de Pensamento Computacional;
2. Avaliar o Pensamento Computacional dos alunos;
3. Verificar o efeito do desenvolvimento do Pensamento Computacional na capacidade de raciocínio dos alunos;
4. Analisar a relação entre Pensamento Computacional e os diferentes tipos de raciocínio: raciocínio verbal, raciocínio numérico, raciocínio espacial, raciocínio abstrato e raciocínio mecânico. (BOUCINHA, 2017, p. 13).

Para uma tese na área de educação, consideramos que os objetivos (geral e específicos) poderiam estar vinculados com o processo pedagógico que envolve os sujeitos que “aprendem” pensamento computacional. Desta forma, podemos nos diferenciar de Boucinha (2017) ao estabelecer relações entre o pensamento computacional e o processo de desenvolvimento da argumentação dedutiva, a partir do recurso de ensinar programação de computadores pautado na resolução de problemas, ou seja, identificar o papel do pensamento computacional na capacidade de produzir argumentos válidos.

Boucinha (2017) definiu o Scratch, o teste de Pensamento Computacional e o teste BPR-5 (raciocínio abstrato, verbal, espacial, numérico e mecânico) - Bateria de Provas de Raciocínio - os seus objetos de investigação, elaborando um curso de desenvolvimento de games através da Teoria de Aprendizagem Significativa e a Teoria da Aprendizagem Experiencial, sob referencial teórico de Ausebel (1980) e Kolb (1984); Roman (2015) e Seymour Papert são base para o desenvolvimento do pensamento computacional.

Boucinha (2017) classificou a sua pesquisa como quantitativa, de natureza aplicada e explicativa, caracterizando um estudo “quase-experimental”. Assim, desenvolveu um curso de desenvolvimento de games para 50 alunos, do 6º ao 9º ano do Ensino Fundamental, com duração de 7 semanas, sendo um encontro por semana, totalizando 10 horas/aula. Onde aplicou um pré-teste que avaliou o pensamento computacional e o BPR-5, executou o curso através do Scratch e o pós-teste (pensamento computacional e BPR-5), em seguida comparou as médias do teste inicial e final, com base em estatísticas.

Em seus resultados, Boucinha (2017) identificou que houve um incremento na capacidade pensamento computacional dos alunos com o teste de pensamento computacional e constatou que e houve um aumento na média em todos os testes do BPR-5 (raciocínio verbal, abstrato, mecânico, espacial e numérico). Boucinha (2017) também afirma que o ensino de Pensamento Computacional aprimora a capacidade de raciocínio dos alunos, além de detectar uma correlação entre Pensamento Computacional e capacidade de raciocínio (abstrato, verbal, espacial, numérico e mecânico).

Em linhas gerais a revisão da literatura feita por Boucinha (2017) acerca das teorias de aprendizagem, em especial sobre a Teoria de Aprendizagem Significativa de Ausebel (1980) e

Teoria da Aprendizagem Experiencial de Kolb (1984), serviram de base para elaboração de uma proposta pedagógica para ensino do Pensamento Computacional, a qual resultou no Curso de Desenvolvimento de Games utilizado na pesquisa. Entretanto, alguns limites do estudo também foram ponderados pelo pesquisador. Primeiramente, ampliar a amostra e utilizar também alunos com características sociodemográficas distintas da população analisada, para assim, confirmar os resultados da pesquisa feita por ele. Por fim, coloca que o desenvolvimento do Pensamento Computacional deve ser devidamente estudado em outras pesquisas, envolvendo também adultos e a população idosa, verificando se ocorrem os mesmos benefícios na capacidade de raciocínio numa população adulta.

A partir das contribuições dadas pela tese de Boucinha (2017), concluímos sobre alguns aspectos com potencial de aprofundamento e investigação que podem ser um diferencial para a tese. O primeiro aspecto trata-se do objeto de pesquisa, visto que o foco é a resolução de problemas e a capacidade argumentativa apresentada neste processo de explicar os seus caminhos e o porquê de sua resposta, visto que Boucinha (2017) já nos mostrou que o Scratch contribui com o desenvolvimento do pensamento computacional e o raciocínio; o segundo aspecto trata-se de uma análise teórica, com outros autores e caminhos sobre aspectos didáticos e cognitivos.

Outro aspecto de aprofundamento diz respeito ao potencial que a utilização da programação de computadores, através do Scratch pode oferecer, uma vez que permite a explicitação, explicação, relações e caminhos escolhidos pelo estudante, até chegar a uma resposta, independente da mesma ser correta ou não. Boucinha (2017) mostra que explorou os recursos contidos no Scratch, mas não aprofunda o processo contido na aplicação da metodologia utilizada, podendo ser um ponto determinante para o resultado que coletou.

Em síntese, Boucinha (2017) testou e comprovou que o ensino do pensamento computacional contribui para a melhora da capacidade de raciocínio dos indivíduos, bem como existe uma correlação entre pensamento computacional e capacidade de raciocínio verbal, abstrato e mecânico. O autor percebe a importância de novos estudos para ampliar suas amostras, utilizando alunos com característica sociodemográfica diferente da população estudada, como também considerar adultos e idosos, para verificar se ocorrem os mesmos benefícios na capacidade de raciocinar deste público.

Na sua tese de doutorado, intitulada “Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica”, Brackmann (2017, p. 21) coloca como problema de pesquisa da sua tese: “Qual a eficácia da abordagem de ensino desplugado para promover o desenvolvimento do Pensamento Computacional na Educação Básica?”

Diante deste questionamento, o autor colocou que seu objetivo geral foi de “verificar a possibilidade de desenvolver o Pensamento Computacional na Educação Básica utilizando exclusivamente atividades desplugadas” (BRACKMANN, 2017, p. 21). Este objetivo principal, dividiu-se em seis objetivos específicos, visto que utilizou-se do “tripé educacional” e aplicou o estudo à pesquisa, ensino e extensão. Os objetivos específicos são:

1. Desenvolver objetos de aprendizagem de Pensamento Computacional para utilização desplugada;
2. Realizar intervenções em sala de aula com as atividades desplugadas desenvolvidas;
3. Avaliar o Pensamento Computacional dos estudantes que participaram da intervenção e do grupo controle;
4. Verificar o efeito das atividades propostas no desenvolvimento do Pensamento Computacional;
5. Comparar os resultados obtidos nas intervenções realizadas no Brasil e Espanha;
6. Comparar os efeitos das intervenções entre gêneros. (BRACKMANN, 2017, p. 21-22).

Estamos diante de uma tese construída na área de educação, assim, consideramos que os objetivos mostram-se vinculados ao processo pedagógico que envolve a amostra de pesquisa, com vistas à avaliação do pensamento computacional. Desta maneira, podemos nos diferenciar de Brackmann (2017) ao não avaliar somente o pensamento computacional, mas ao estabelecer relações entre o pensamento computacional e o processo de desenvolvimento da argumentação dedutiva, de forma plugada (programação de computadores).

Brackmann (2017) propôs uma abordagem desplugada para o ensino de Pensamento Computacional em escolas da Educação Básica, com a criação de objetos de aprendizagem e sua aplicação, para verificar a eficácia desta abordagem. Utilizou como referencial teórico Jeannette Wing, Seymour Papert e Cynthia Solomon desde a definição de Pensamento Computacional, perpassando pela fase de aplicação e conclusões do trabalho. Utilizou uma abordagem quali-quantitativa com vistas a natureza aplicada e com objetivos exploratórios, ou seja, uma abordagem “quase-experimental”, visto que trabalhou com a Educação Básica utilizando atividades desplugadas, onde atuou com um pré-teste, aulas de pensamento computacional desplugado (duração de 10h) e um pós-teste. Sendo uma amostra de 73 crianças espanholas e 63 crianças brasileiras.

Brackmann (2017) propôs uma investigação em quatro etapas, sendo duas piloto com o objetivo exploratório e de refinamento dos instrumentos, pois não utilizou os dados para análise estatística. As outras duas etapas ocorreram da mesma forma, porém uma aplicação ocorreu na Espanha e outra no Brasil, após, cruzou os dados estatisticamente e constatou que houve uma melhoria significativa no desempenho escolar dos estudantes, que tiveram

atividades de Pensamento Computacional em ambos os países (Brasil e Espanha) em que aplicou o estudo.

Em síntese, Brackmann (2017) testou e comprovou a eficácia de uma abordagem através do pensamento computacional, apontando melhora no desempenho escolar dos indivíduos participantes da pesquisa. O autor percebe que em trabalhos futuros é possível verificar se o material desplugado apresentará a mesma eficácia em estudantes de escolas particulares, bem como as atividades desplugadas podem auxiliar na aprendizagem de linguagens visuais; e desenvolver um meio de medir o pensamento computacional em alunos dos anos iniciais. Além da possibilidade de identificar o ponto em que a abordagem desplugada perde seu efeito e a utilização de máquinas (computadores) passa a ser indicada.

A partir das contribuições dadas pelas teses de Boucinha (2017) e Brackmann (2017), concluimos sobre alguns aspectos com potencial de aprofundamento e investigação que podem ser um diferencial para a tese. O primeiro aspecto trata-se do objeto de pesquisa, visto que o foco é a resolução de problemas e a capacidade argumentativa apresentada neste processo de explicar os seus caminhos e o porquê de sua resposta, articulado ao pensamento computacional, visto que Brackmann (2017) já nos mostrou que mesmo utilizando-se de atividades desplugadas, é possível desenvolver o pensamento computacional, bem como Boucinha (2017) detectou que existe uma correlação entre pensamento computacional e capacidade de raciocínio.

Outro aspecto a se destacar tem relação com a não utilização do computador, para abordar o pensamento computacional, mas acreditamos que a tecnologia possa ser uma ferramenta válida para as intervenções didáticas, e poderiam incentivar o diálogo ao abordar os quatro pilares envolvidos no conceito de pensamento computacional, justamente para compreender o raciocínio utilizado por cada um dos estudantes, a fim de formar seus argumentos.

2.4 Considerações parciais

A construção deste “estado do conhecimento” viabilizou um mapeamento das teses produzidas no período de 2016 a 2019, dentro das perspectivas educacionais e o encontro de dois trabalhos similares ao que estamos propondo, no que diz respeito à temática. Estes achados, nos auxiliaram na reflexão sobre as lacunas existentes nas teses analisadas em relação a temática e as percepções teórico metodológicas que criam possibilidades de avanço no campo da pesquisa em educação. Desta forma, passaremos a discutir o cenário teórico, no

sentido de refletir e construir a ciência, dentro dos conceitos de pensamento computacional, raciocínio lógico e argumentação dedutiva.

Com base nas contribuições dadas pelos estudos apresentados, mostra-se viável pensar sobre aspectos a serem trabalhados na tese. O primeiro refere-se a possibilidade de um ambiente de programação de computadores, com vistas ao pensamento computacional, oferecendo uma correlação com a capacidade de raciocínio, ou seja, “[...] ao ensinar um computador a ‘pensar’, a criança embarca numa exploração sobre a maneira como ela própria pensa [...]” (PAPERT, 1985, p. 35). Agora, cabe a nós investigar como esses espaços que ensinam programação de computadores, estabelecem relações entre o pensamento computacional e o processo de desenvolvimento da argumentação dedutiva, bem como se existem conexões do pensamento computacional na formação de um ser argumentativo que resolve problemas quando programa computadores.

Um segundo aspecto refere-se às metodologias utilizadas, para produção dos dados empíricos desta tese, tendo em vista explorar as declarações argumentativas sob a lógica de representação e compreensão de soluções de problemas, para tanto é necessário utilizar-se de estratégias que possibilitam fazer estas observações. Portanto, mostra-se significativo a utilização de um questionário e observação, para identificar os argumentos, além de um teste para quantificar o pensamento computacional. Esta abordagem empírica deve ocorrer com poucos indivíduos, operando a programação de computadores a fim de resolver problemas, buscando a percepção dos processos envolvidos na manifestação de um bom argumento.

3 RACIOCÍNIO LÓGICO E O PROCESSO CONSTRUTIVO DO ARGUMENTO

O atual sistema educacional permanece ancorado em uma metodologia tradicional, centrada em disciplinas teóricas, onde os alunos memorizam fórmulas e expressam mecanicamente seus conhecimentos, destacando-se pouco nas habilidades como autonomia, criatividade, resolução de problemas e reflexão crítica. Em meio às diversas habilidades necessárias para o desenvolvimento de crianças e adolescentes na fase escolar, destaca-se a importância do raciocínio lógico.

O raciocínio lógico é compreendido como a organização ou estruturação de raciocínios que permitem chegar a uma conclusão ou resolver um problema de acordo com normas específicas. Em termos simples, é a capacidade do indivíduo de organizar suas ideias e pensamentos para encontrar soluções, baseando-se em evidências e análises críticas das situações. Aristóteles (2016) associa a lógica a duas vertentes: o uso do raciocínio na execução de atividades e o estudo filosófico do raciocínio válido. O termo raciocínio é caracterizado como uma operação lógica discursiva e mental, necessária para organizar dados e chegar a uma conclusão.

No contexto da argumentação, Weston (2009) destaca que argumentar vai além de expor pontos de vista. É um processo que envolve o uso de raciocínio lógico discursivo e mental, essencial para organizar dados, seja por meio de imagens, palavras ou números, utilizando premissas para alcançar uma conclusão. Em resumo, argumentar representa o ato e a maneira de pensar e argumentar para chegar a uma determinada conclusão, ampliando a compreensão sobre o processo cognitivo e a lógica envolvida na formação de argumentos válidos, “significa apresentar um conjunto de razões ou provas que fundamentam uma conclusão” (WESTON, 2009, p. XI). Assim, os argumentos se fundamentam em pontos de vista embasados em razões sólidas, aptas a sustentar as conclusões, proporcionando a oportunidade de avaliar a consistência desses argumentos como um meio de investigação.

Desta maneira, a utilização do raciocínio lógico na formação educacional de jovens pode gerar pessoas críticas com senso argumentativo, e é com essa característica que desenvolvemos alunos capazes de criar, interpretar, responder e explicar situações problema. Neste sentido, busca-se compreender o papel do raciocínio lógico no processo de construção do argumento dedutivo, bem como identificar articulações entre o desenvolvimento do pensamento computacional e a construção de argumentos dedutivos, através dos seguintes questionamentos: Qual o vínculo existente entre raciocínio lógico e argumentação dedutiva?

Quais as articulações necessárias para ocorrer conexão entre as premissas e a conclusão na construção de um argumento dedutivo?

Para que estes pressupostos sejam atendidos, este capítulo está destinado à análise do raciocínio lógico no âmbito da argumentação e está estruturado para proporcionar uma compreensão abrangente desse processo cognitivo. Está delineada uma explanação detalhada dos principais tipos de raciocínio, destacando suas características distintas e implicações na construção argumentativa.

3.1 Raciocínio lógico e suas frentes

O raciocínio lógico está associado a conceitos que proporcionam organização e clareza nas situações cotidianas, preparando os jovens para lidar com circunstâncias mais desafiadoras. Conforme a perspectiva construtivista de Jean Piaget (1978), o ensino de Matemática através da imposição de fórmulas, exercícios repetitivos e conceitos limitados prejudica o aprendizado, resultando em estudantes passivos, desinteressados e carentes de criatividade.

Ao incorporar o raciocínio lógico na formação educacional dos jovens, desenvolvemos indivíduos críticos, dotados da habilidade de criar, interpretar, responder e explicar situações problema em contextos que envolvem não apenas a Matemática, mas também áreas como arte, filosofia, história, geografia, física ou química.

No âmbito da resolução de problemas, as ideias de Pólya destacam que

o problema pode ser modesto, mas se desafiar a curiosidade e puser em jogo as faculdades inventivas, quem o resolver pelos seus próprios meios experimentará o prazer e o triunfo da descoberta. Tais experiências, numa idade susceptível, poderão criar o gosto pelo trabalho mental e deixar, por toda a vida, uma marca indelével na mente e no carácter. (2006, Prefácio).

Para Rushkoff (2012, p.7), “quando nós, os humanos adquirimos linguagem, aprendemos não somente a ouvir, mas a falar. Quando ganhamos a escrita, nós aprendemos não apenas a ler, mas a escrever”. Assim, a aplicação do raciocínio lógico impacta de maneira positiva, oferecendo contribuições significativas em três dimensões essenciais: leitura, escrita e resolução de problemas. Estes, após uma sequência de estudos lógicos, passam a representar novas sistematizações, englobando a habilidade de ler com proficiência, expressar-se por escrito de forma competente e solucionar problemas de maneira eficaz.

Segundo o dicionário de filosofia de Abbagnano, raciocinar, em um sentido amplo, significa “[...] inferir uma proposição de uma ou mais proposições precedentes, e crer ou pretender que se creia nela como conclusão de qualquer outra coisa” (2007, p. 832). Ainda para o autor, raciocínio compreende “[...] qualquer procedimento de inferência ou prova; portanto, qualquer argumento, conclusão, inferência, indução, dedução ou analogia” (2007, p. 832).

O raciocínio lógico desempenha um importante papel na realização de inferências, iniciando-se com uma afirmação ou proposição inicial, seguida por uma afirmação intermediária e culminando em uma conclusão. Nesse sentido, ele não apenas serve como uma ferramenta analítica e sequencial, mas também como um meio de justificar, analisar, argumentar ou validar diversos raciocínios. Em outras palavras, o raciocínio lógico fundamenta-se em dados que podem ser comprovados, garantindo, assim, precisão e exatidão em suas inferências (WALTON, 2012).

Conforme destacado por Arslan, Gocmencelebi e Tapan (2009), a importância da habilidade de raciocinar transcende seu emprego exclusivo na disciplina de Matemática, desempenhando um papel significativo na vida do estudante de maneira abrangente. Os autores argumentam que,

a capacidade de raciocínio pode ajudar os alunos a entender e avaliar a sociedade científica e tecnológica. Porque o raciocínio é altamente eficaz para a capacidade dos alunos de analisar novas situações que são enfrentadas em todos os aspectos; fazer suposições lógicas, explicar seus pensamentos, chegar a conclusões e defender suas conclusões. (p. 2460).

Existem diversos tipos de raciocínio lógico, cada um trilhando uma linha específica de pensamento e analisando situações sob diferentes perspectivas. Contudo, a diversidade desses tipos, como dedução, indução, abdução e analogia, não implica que um seja mais apropriado do que o outro, uma vez que todos possibilitam alcançar um resultado. Cada abordagem oferece vantagens distintas, contribuindo para uma compreensão abrangente das questões em análise.

3.1.1 Raciocínio dedutivo

A importância desse raciocínio se encontra não só na Matemática. Aristóteles, no século IV a.C., já havia notado que “[...] uma pessoa que possui a habilidade dedutiva é capaz de compreender o universo de maneiras mais profundas e abrangentes” (AYALON; EVEN,

2010, p. 1134). O papel fundamental do raciocínio dedutivo é evidenciado na explicação e previsão científica. Ao partir de leis causais gerais e declarações que descrevem a condição inicial, o raciocínio dedutivo se estabelece como um domínio específico do conhecimento. Notavelmente, esse tipo de raciocínio pode auxiliar os alunos na identificação e resolução de conflitos cognitivos, como afirmam Park e Han (2002).

Ao conceber uma demonstração, o raciocínio dedutivo se destaca, sendo o único entre os demais que assegura a certeza de uma proposição matemática. Nesse contexto, ele se revela como uma ferramenta essencial, proporcionando um método rigoroso que valida e estabelece com segurança a veracidade de afirmações no campo matemático. Segundo Reid e Knipping (2010, p. 84), “[...] o raciocínio dedutivo é a base da demonstração, sendo um dos principais focos do estudo da lógica formal”. De acordo com Ayalon e Even (2008, p. 01),

o raciocínio dedutivo é único na medida em que é o processo de inferir conclusões a partir de informações conhecidas (chamadas premissas) baseadas em regras lógicas formais, em que conclusões são necessariamente derivadas da informação dada e não há necessidade de validá-las por experimentos.

Conforme destacado por Ayalon e Even (2008), os argumentos dedutivos válidos desempenham um importante papel na preservação da verdade, indicando que, se as premissas são verdadeiras, a conclusão também deve ser verdadeira. No contexto matemático, ao empregar o raciocínio dedutivo, parte-se de informações anteriores, como definições, axiomas, postulados e proposições já demonstradas, utilizando inferências lógicas para derivar novos resultados. Na prática matemática, essa abordagem se distingue pela quantidade de informações prévias disponíveis durante a demonstração de um resultado; frequentemente, não se utiliza apenas duas informações para chegar a uma terceira, mas sim uma cadeia de inferências que leva ao teorema desejado. No contexto do ensino de Matemática, essa abordagem é essencial para que os alunos compreendam a natureza rigorosa e confiável dessa disciplina.

Consequentemente, o raciocínio lógico por dedução é frequentemente vinculado ao raciocínio lógico matemático, uma vez que alcança um resultado ou conclusão com base em uma probabilidade evidente. Esse tipo de raciocínio segue uma regra específica e utiliza suas premissas para deduzir uma conclusão específica. Essa associação ressalta a natureza sistemática e precisa do raciocínio lógico por dedução, especialmente quando aplicado ao contexto matemático. Exemplo disto é:

Quadro 2 - Exemplo 1 de raciocínio dedutivo

PREMISSA: Todos os humanos são mortais.
PREMISSA: Maria é um ser humano.
CONCLUSÃO: Portanto, Maria é mortal.

Fonte: Construção da autora com base na teoria, 2023.

Neste exemplo, estamos utilizando o raciocínio dedutivo. A primeira premissa estabelece uma regra geral sobre a mortalidade dos humanos. A segunda premissa afirma que Maria é um ser humano. A conclusão, derivada logicamente das premissas, é que Maria é mortal. Esse é um exemplo clássico de dedução, onde a conclusão é necessariamente verdadeira se as premissas são verdadeiras.

Considerando outro exemplo,

Quadro 3 - Exemplo 2 de raciocínio dedutivo

PREMISSA: Todos os mamíferos têm glândulas mamárias.
PREMISSA: Um elefante é um mamífero.
CONCLUSÃO: Portanto, um elefante tem glândulas mamárias.

Fonte: Construção da autora com base na teoria, 2023.

Neste caso, a primeira premissa estabelece uma característica geral dos mamíferos. A segunda premissa afirma que um elefante é um mamífero. A conclusão deduzida logicamente é que um elefante possui glândulas mamárias. Esse exemplo ilustra como o raciocínio dedutivo opera, conectando informações gerais a uma conclusão específica com base na lógica.

3.1.2 Raciocínio indutivo

Com fundamentação nas concepções de Haverty et al. (2000) e Reid e Knipping (2010), é possível definir o raciocínio indutivo como o procedimento de deduzir uma regra geral por meio da observação e análise de instâncias específicas, delineando, assim, um entendimento mais amplo a respeito desse tipo de raciocínio, e, portanto, “[...] é um processo vital para a vida cotidiana e para a investigação científica em particular” (PAPAGEORGIO, 2009, p. 313). Em outras palavras,

a indução, então, é aquela operação da mente, pela qual inferimos que o que sabemos ser verdadeiro em um caso ou casos particulares, será verdadeiro em todos os casos que se assemelham ao primeiro, em certos aspectos designáveis. Em outras palavras, a indução é o processo pelo qual nós concluímos que o que é verdadeiro sobre certos indivíduos de uma classe é verdadeiro para toda a classe. A indução,

assim definida, é um processo de inferência que prossegue do conhecido para o desconhecido. (MILL, 1884, p. 210 apud REID; KNIPPING, 2010, p. 89).

O raciocínio indutivo no contexto do ensino de Matemática está intrinsecamente ligado à observação de padrões e relações que existem entre números e figuras, conforme apontado por Papageorgio (2009) e Neubert e Binko (1992). Com base nessas características, é possível conceber diversas atividades matemáticas comuns que incorporam uma abordagem indutiva.

A relevância do raciocínio indutivo no ensino matemático é notável, uma vez que ele não apenas possibilita, mas também evidencia que o processo de fazer matemática é construtivo. Em outras palavras, a aplicação desse tipo de raciocínio cria um espaço propício para discutir a importância das tentativas e erros, da formulação de conjecturas e da realização de generalizações na prática matemática.

O raciocínio indutivo na Matemática difere do raciocínio indutivo nas ciências empíricas, na medida em que há um teste final, embora não necessariamente um procedimento de decisão, que pode ser usado para determinar o que é uma indução correta. Isto é, se o que é induzido é verdadeiro, isto é: se pode ser deduzido pelo sistema dedutivo, então deve ser correto. (BROWN; TARNLUND, 1977, p. 1).

Em termos mais específicos, para alcançar validação matemática, o raciocínio indutivo necessita submeter-se a um processo dedutivo. A integração de ambos os tipos de raciocínio - dedutivo e indutivo - durante o ensino da Matemática pode propiciar momentos enriquecedores de discussão.

O raciocínio lógico por indução, em essência, envolve a determinação de uma regra por meio de um processo analítico e crítico, observando diversos exemplos que demonstram como uma determinada situação invariavelmente conduz a outra. Em outras palavras, a conclusão segue a premissa. Este método é frequentemente empregado por cientistas na formulação de generalizações fundamentadas em observações específicas. Exemplos disto é:

Quadro 4 - Exemplos de raciocínio indutivo

EXEMPLO 1: Números Primos OBSERVAÇÃO: 3, 5, 7 são números primos. PADRÃO IDENTIFICADO: Todos esses números primos são ímpares. CONCLUSÃO INDUTIVA: Todos os números primos são ímpares.
EXEMPLO 2: Sequência Numérica OBSERVAÇÃO: 1, 4, 9, 16 são quadrados perfeitos. PADRÃO IDENTIFICADO: Cada número na sequência é o quadrado do número natural subsequente (1^2, 2^2, 3^2, 4^2, ...).

CONCLUSÃO INDUTIVA: Todos os números na sequência são quadrados perfeitos.

Fonte: Construção da autora com base na teoria, 2023.

Esses exemplos ilustram como o raciocínio indutivo opera, partindo de observações específicas para derivar conclusões gerais. É importante notar que, embora essas conclusões possam ser válidas com base nas observações feitas, o raciocínio indutivo não fornece uma garantia absoluta, ao contrário do raciocínio dedutivo.

3.1.3 Raciocínio abdutivo

De acordo com Reid e Knipping (2010, p. 100), o raciocínio abdutivo pode ser compreendido como a contrapartida do raciocínio dedutivo, sendo também denominado como raciocínio de trás para frente. Os autores exemplificam essa forma de raciocínio por meio de um trecho do romance "*A Study in Scarlet*", protagonizado por Sherlock Holmes, e para ilustrar esse raciocínio,

a maioria das pessoas, se você descrever uma série de eventos para eles, dirá qual seria o resultado. Eles podem reunir esses eventos em suas mentes e argumentar com eles que algo acontecerá. Há poucas pessoas, no entanto, que, se lhes disserem o resultado, poderiam evoluir a partir de sua própria consciência interior, quais foram os passos que conduziram a esse resultado. Esse poder é o que quero dizer quando falo de raciocinar de trás para frente. (DOYLE, 1887 apud REID; KNIPPING, 2010, p. 100).

De forma mais simplificada, o raciocínio abdutivo busca entender as causas que levaram a ocorrência de um determinado evento, ou seja, procura inferir a explicação mais plausível para o evento em questão. Essa abordagem de raciocínio é comumente encontrada no dia a dia, como no caso do diagnóstico médico, em que, a partir de sintomas apresentados, o médico formula hipóteses sobre as possíveis causas do quadro clínico.

Peirce afirma que "todas as ideias da ciência a ela advém através da abdução. A abdução consiste em estudar os fatos e projetar uma teoria explicá-los" (2003, p. 207). Com base nesse autor, Souza (2015, p. 79) destaca que a abdução "[...] consiste em observar um aglomerado de elementos (de uma mesma natureza) que formam um conjunto de fatos e em permitir que esses fatos sugiram uma teoria" (2015, p. 79); entretanto, essa teoria possui uma natureza passível de falhas, ou seja, sua validade pode ser contestada.

Embora a Matemática seja uma ciência dedutiva por natureza, a sua produção engloba diversas formas de raciocínio. Conforme apontado por Souza, o processo de construção do conhecimento matemático

[...] pode ser fruto do método investigativo, iniciando com o raciocínio abdutivo como um ato inferencial, uma hipótese provisória que tem origem na pergunta (ou no ato de questionar), uma maneira de se iniciar esse processo de produção [...]. Isso nos leva à ideia da abdução como um raciocínio que abre possibilidades de uma nova inteligibilidade daquilo que se vê e do que se pode expressar quando elaboramos uma explicação acerca do que é visto. (2015, p. 83).

Segundo o autor, é possível perceber que o raciocínio lógico por abdução desempenha um papel significativo no processo de produção do conhecimento matemático. Talvez seja por esse motivo que ele não seja prontamente perceptível ao observarmos a Matemática já estruturada e formalizada. Exemplo disto é:

Quadro 5 - Exemplos de raciocínio abdutivo

EXEMPLO 1: Investigação de um Incêndio

OBSERVAÇÃO: Um incêndio começou em uma residência.

PADRÃO IDENTIFICADO: Há vestígios de um líquido inflamável no local.

CONCLUSÃO ABDUTIVA: O incêndio pode ter sido provocado intencionalmente.

EXEMPLO 2: Diagnóstico Médico

OBSERVAÇÃO: O paciente apresenta febre, dor de garganta e fadiga.

PADRÃO IDENTIFICADO: Esses sintomas são consistentes com uma infecção viral.

CONCLUSÃO ABDUTIVA: O paciente provavelmente está sofrendo de uma virose.

Fonte: Construção da autora com base na teoria, 2023.

Em ambos os exemplos, o raciocínio abdutivo opera ao formular a melhor explicação possível para os dados observados, sem garantia absoluta de precisão. Essas inferências são valiosas em situações em que a certeza total é difícil de alcançar, mas uma conclusão plausível pode ser obtida com base nas evidências disponíveis.

3.1.4 Raciocínio analógico

O raciocínio por analogia é bem frequente no ensino da Matemática. Ele consiste em “[...] elaborar conjecturas baseadas nas semelhanças entre dois casos, um bem conhecido (fonte) e um menos compreendido (alvo)” (REID; KNIPPING, 2010, p. 110), em outras palavras, esse tipo de raciocínio envolve a observação de um caso para o qual se dispõe de

informações suficientes e a tentativa de inferir possíveis resultados em uma situação diferente, fundamentada nas semelhanças identificadas entre os casos analisados. Segundo Pólya, a analogia “é uma espécie de semelhança. Objetos semelhantes coincidem uns com os outros em algum aspecto; objetos análogos coincidem em certas relações das suas respectivas partes.” (2006, p. 34).

Ainda nesta perspectiva,

a analogia permeia todo o nosso pensamento, a nossa fala cotidiana e as nossas conclusões triviais, assim como os modos de expressão artística e as mais elevadas conclusões científicas. Ela é empregada nos mais diferentes níveis. É comum o uso de analogias vagas, incompletas ou obscuras, porém a analogia pode alcançar-se ao nível do rigor matemático. Todos os tipos de analogia podem desempenhar uma função na descoberta da solução e, por isso, não devemos desprezar nenhum deles. (PÓLYA, 2006, p. 34).

Reid e Knipping (2010) ilustram esse tipo de raciocínio por meio de uma entrevista conduzida com um aluno do ensino fundamental. Os pesquisadores solicitaram que a criança explicasse por que a soma de dois números ímpares resulta em um número par. A resposta do aluno foi "porque negativo vezes negativo é positivo". Observa-se que o estudante estabeleceu uma conexão entre dois conceitos distintos da Matemática - números pares/ímpares e números positivos/negativos. Embora essa associação possa parecer peculiar, na realidade, esses conceitos compartilham semelhanças.

Assim, no raciocínio lógico por analogia, o indivíduo estabelece a premissa. Nesse contexto, a conclusão e as regras funcionam como fundamentos para sustentar uma determinada premissa, explicando o raciocínio que levou àquela conclusão. Exemplo disto é:

Quadro 6 - Exemplos de raciocínio analógico

EXEMPLO 1: Analogia em Biologia

PREMISSA: A célula é para um organismo o que um tijolo é para uma construção.

CONCLUSÃO ANALÓGICA: Assim como um tijolo é um componente essencial na construção de um edifício, as células são unidades fundamentais na constituição de um organismo.

EXEMPLO 2: Analogia em Economia

PREMISSA: O fluxo de dinheiro em uma economia é semelhante ao fluxo de sangue em um corpo humano.

CONCLUSÃO ANALÓGICA: Da mesma forma que o sangue circula para sustentar as funções vitais do corpo, o dinheiro circula para manter as atividades econômicas e o funcionamento saudável de uma sociedade.

Fonte: Construção da autora com base na teoria, 2023.

Esses exemplos ilustram como o raciocínio analógico opera ao estabelecer relações entre conceitos aparentemente diferentes, visando ampliar a compreensão de uma situação por meio de analogias. Evidencia-se isto no exemplo 1, porque um tijolo é uma unidade básica e essencial na construção de um edifício, a célula desempenha um papel fundamental como a unidade estrutural e funcional na constituição de um organismo. A analogia enfatiza a importância da célula como bloco de construção essencial para a vida, assim como um tijolo é para uma edificação. Da mesma forma, no exemplo 2, que o sangue circula para sustentar as funções vitais do corpo humano, o dinheiro circula para manter as atividades econômicas e garantir o funcionamento saudável de uma sociedade. Nesta analogia, destaca-se a ideia de que o dinheiro é essencial para a saúde e a vitalidade de uma economia, assim como o sangue é para o corpo humano.

Os modos de raciocínio, como dedução, indução, abdução e analogia, podem ser expressos por meio de três formas distintas: verbal, espacial ou abstrata. A expressão verbal envolve a habilidade de compreender e estruturar elementos verbais, resultando na criação de significados, ordem e relações entre eles. A representação espacial refere-se à aptidão para criar e manipular imagens mentais visuais, relacionando-se com a capacidade de visualização e raciocínio em três dimensões. Por fim, a expressão abstrata está associada ao pensamento abstrato e à habilidade de estabelecer conexões abstratas entre conceitos por meio de ideias inovadoras.

De maneira geral, o raciocínio desempenha um importante papel ao nos possibilitar resolver desafios práticos do cotidiano. Sua importância transcende a mera habilidade numérica, concentrando-se, sobretudo, na capacidade de encontrar soluções eficazes para dilemas diários. Ao lidar com cálculos, por exemplo, o raciocínio está mais intimamente ligado à nossa habilidade de traçar o caminho adequado para superar um impasse do que à destreza numérica em si. A lógica intrínseca a esse raciocínio se dedica ao estudo dos argumentos, revelando formas específicas que evidenciam como uma conclusão é derivada das premissas ou proposições estabelecidas anteriormente.

3.2 Processo argumentativo

Uma das competências mais essenciais para qualquer indivíduo que precisa tomar decisões é o pensamento crítico. Essa habilidade envolve a aplicação da lógica informal na análise e avaliação de argumentos, sendo o argumento o principal foco do pensamento crítico (WALTON, 2012). O propósito fundamental de um argumento é buscar a verdade,

fundamentado em premissas robustas, uma vez que argumentar significa apresentar um conjunto de razões ou evidências que sustentam uma conclusão (WESTON, 2009). Nessa perspectiva, os argumentos têm a finalidade de auxiliar as pessoas a pensar de maneira mais clara, lógica e racional.

Conforme Sacrini (2016), um argumento é uma estrutura lógica composta por dois tipos de declarações: premissas e conclusões. Todo argumento é constituído por uma ou mais premissas, que representam um tipo de declaração, e por uma única conclusão. As premissas são as razões que procuram determinar a validade da conclusão, que, por sua vez, é outra razão. Ainda no viés do autor os argumentos consistem em pelo menos três elementos: conclusão, premissas e inferências. A conclusão representa a tese ou posição que está sendo defendida no argumento. Já as premissas, são uma ou mais declarações que funcionam como justificativas ou razões para apoiar a conclusão; São as bases que sustentam o argumento. E a inferência refere-se à passagem lógica que conecta as premissas à conclusão, indicando como as razões apresentadas apoiam a tese defendida. Esses elementos, de maneira interligada, compõem a estrutura lógica de um argumento, visando apresentar uma linha de raciocínio que conduza a uma conclusão fundamentada.

A premissa constitui uma declaração fundamentada na assertiva de algo considerado verdadeiro, sendo crucial para o funcionamento adequado do argumento, “é preciso pressupor um ponto de partida de dados não discutidos, tomados, ainda que apenas contextualmente, como certos ou ao menos aceitáveis” (SACRINI, 2016, p. 21), possibilitando-nos avaliar as premissas ao fornecer evidências ou provas de veracidade, a fim de sustentar uma conclusão.

A conclusão representa o resultado final do processo lógico, utilizando as premissas como matéria-prima. É o aspecto que responde à pergunta "sobre o que estamos falando" no contexto da argumentação, sendo definido como "tentativas de fundamentar determinados pontos de vista com razões" (WESTON, 2009, p. XI). Dado que um argumento pode envolver várias premissas, mas apenas uma conclusão, essa conclusão pode se tornar a premissa de outro argumento, gerando um encadeamento lógico.

Assim, o argumento se apresenta como um método de investigação, pois não se baseia em opiniões, mas sim em razões, provas, explicações, e defende esse processo, abrindo espaço para que outras pessoas possam criar suas próprias razões e provas (WESTON, 2009).

Existem três abordagens ao formular uma argumentação: argumento inválido, argumento dedutivo e argumento indutivo. O argumento inválido é constituído por premissas falsas ou duvidosas, uma vez que a validade da conclusão depende de premissas verdadeiras, conferindo-lhe confiabilidade e eliminando a possibilidade de uma conclusão inválida. O

argumento dedutivo procede de proposições cada vez mais universais para proposições particulares, proporcionando uma forma de demonstração, onde a conclusão é extraída das premissas. Por outro lado, o argumento indutivo parte de proposições particulares ou termos mais restritos em direção a termos mais universais ou abrangentes.

3.2.1 Distinção entre validade e verdade - Falácias

Os argumentos podem ser classificados como válidos ou inválidos. Se um argumento é válido e suas premissas são verdadeiras, então a conclusão também deve ser verdadeira, ou seja, um argumento válido não pode ter premissas verdadeiras e uma conclusão falsa (WALTON, 2012). No entanto, a validade de um argumento está intrinsecamente ligada à verdade ou falsidade real de suas premissas, pois premissas fracas e duvidosas não oferecem um suporte adequado para a conclusão, resultando em uma falácia (SACRINI, 2016). Nesse contexto,

é importante notar que há falácias que são argumentos, isto é, estruturas linguísticas em que algumas sentenças oferecem suporte a conclusão, e há outras que não apresentam claramente essa estrutura, mas são enunciações que operam movimentos argumentativos no interior de uma discussão. Não se trata, nesse último caso, de defender uma tese determinada, e sim de promover certa ênfase em algum tópico, de introduzir novos temas etc. Seja como for, ainda que não defendam explicitamente uma conclusão, essas falácias violam certos procedimentos pressupostos para o debate racional e são, assim, consideradas como erros típicos do processo argumentativo. (SACRINI, 2016, p. 226).

Um argumento é composto por uma ou mais premissas que levam a uma conclusão. É fundamental que a conclusão derive das premissas. Quando a conclusão é uma consequência necessária das premissas, dizemos que o argumento é válido. Por outro lado, se a conclusão não é uma consequência necessária das premissas, consideramos o argumento como inválido. A validade de um argumento, portanto, está relacionada à sua estrutura, à forma como o argumento está organizado (SACRINI, 2016).

As falácias, por sua vez, são argumentos que aparentam ser corretos e conclusivos, mas contêm algum erro em sua estrutura ou conteúdo. Elas são frequentemente encontradas em raciocínios deficientes ou ilusórios que fazem um argumento falho parecer adequado, quando na verdade é limitado, frágil ou inconsistente (WESTON, 2009). Nessa perspectiva, Sacrini destaca que,

[...] a falaciosidade de um argumento nem sempre deve ser associada a um laço inferencial fraco; ela indica, antes, a transgressão de princípios estruturais por meio dos quais os argumentos operam, princípios que se referem seja à forma das inferências, seja ao papel das premissas, seja também às expectativas epistêmicas ligadas ao uso concreto dos argumentos. (SACRINI, 2016, p. 249).

Neste sentido, Sacrini também nos traz que

a intenção argumentativa de justificar por meio de razões teses não imediatamente óbvias ocorre por meio de diferentes tipos de força inferencial. Um resultado importante que deve ser extraído daqui é a distinção entre *validade* e *verdade* na estrutura argumentativa. Validade e não validade são atributos das relações inferenciais entre as sentenças. Nesse sentido, diz-se que os argumentos (tendo em vista as passagens lógicas de que se constituem) são válidos ou não válidos (e nesse último caso, fortes ou fracos). Por sua vez, verdade e falsidade são qualificativos que se aplicam às sentenças. [...] Basta assumir um sentido mínimo, que permita distinguir entre as sentenças que são confirmadas em relação àquilo que elas asseveram e aquelas que não confirmadas (SACRINI, 2016, p. 63).

Um exemplo de falácia ocorre quando se chega validamente à conclusão do consequente da condicional, afirma-se o consequente e, a partir disso, tenta-se concluir pelo antecedente. Esse tipo de raciocínio deixa espaço para contra exemplos, ou seja, a conclusão não é única e, de certa maneira, é questionável (SACRINI, 2016). O esquema inferencial dessa falácia é o seguinte:

Quadro 7 - Esquema inferencial de falácia

Se a, então b b a (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Diante do esquema acima, vamos considerar um exemplo:

Quadro 8 - Exemplo considerando o esquema inferencial de falácia

Se choveu, então o pátio está molhado O pátio está molhado Choveu (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Neste caso, afirmamos que o consequente de uma condicional não assegura a conclusão pelo antecedente dessa última. Utilizando o exemplo, é possível que o pátio esteja

molhado sem que tenha chovido, pois alguém pode tê-lo lavado ou um vazamento hídrico pode ter ocorrido; em resumo, vários contra exemplos são possíveis para esse tipo de inferência falaciosa (SACRINI, 2016).

Em outro exemplo, partimos de uma afirmação condicional e negamos o consequente. Como conclusão válida, segue-se a negação do antecedente. Nessa falácia, após afirmarmos a condicional, negamos o antecedente e concluímos pela negação do consequente, o que permite diversos contra exemplos (SACRINI, 2016). O esquema inferencial dessa falácia é o seguinte:

Quadro 9 - Esquema inferencial de negação de falácia

Se a, então b não a não b (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Diante do esquema acima, vamos considerar um exemplo:

Quadro 10 - Exemplo considerando o esquema inferencial de negação de falácia

Se choveu, então o pátio está molhado Não choveu O pátio não está molhado (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Os mesmos contra exemplos do esquema inferencial anterior são aplicáveis aqui. Assim, fica evidente que a negação do antecedente de uma condicional não oferece justificativas razoáveis para concluir pela negação do consequente (SACRINI, 2016).

Dessa forma, uma falácia pode ser considerada um erro resultante de raciocínio impróprio. A falácia possui a aparência de um argumento correto e válido, embora não o seja, indo de encontro aos argumentos verdadeiros que são sólidos. Nos argumentos verdadeiros, a conclusão segue obrigatoriamente das premissas, não apenas porque assume a hipotética verdade destas, mas também porque as premissas são efetivamente verdadeiras, sendo esse fato confirmado na conclusão.

3.2.2 Argumento indutivo

Um argumento indutivo é aquele em que se parte de experiências sobre fatos particulares para inferir conclusões gerais, que podem ser questionáveis ou até mesmo falsas. Em outras palavras, nos argumentos indutivos, a conclusão não se segue necessariamente das premissas assumidas como verdadeiras, mas sim com certo nível de força probatória, tornando esses argumentos não válidos (SACRINI, 2016, p. 52).

Ao afirmarmos que todos os homens que nascerem irão morrer porque até hoje ninguém deixou de morrer, estamos utilizando um argumento indutivo. Esses argumentos baseiam-se na experiência passada para sustentar uma conclusão. No entanto, mesmo se admitirmos a verdade das premissas, a conclusão em questão poderia ser recusada; ela não se impõe com necessidade, mas apenas com certo grau de força (SACRINI, 2016, p. 52).

Quanto ao grau de força de um argumento indutivo, Sacrini ainda nos esclarece que argumentos indutivos fracos "são aqueles para os quais há pelo menos um contra exemplo formulado com possibilidades lógicas plausíveis no contexto da discussão em vista" (SACRINI, 2016, p. 58) e ainda nos coloca que,

[...] cabe esclarecer que a apresentação dos argumentos indutivos por contraste com os dedutivos não implica que os argumentos indutivos são meros argumentos dedutivos falhos, frutos de uma incapacidade do arguidor de formular laços inferenciais necessários para as teses em vista. Se assim fosse, bastaria dispensar a própria noção de argumentos indutivos ao transformar todos eles em argumentos dedutivos. De fato, essa transformação parece ser possível, de modo produtivo, em vários casos. Com o acréscimo de certas premissas, obtém-se um laço necessário com a conclusão. Ora, se isso puder ser feito *sem que se incluam premissas falsas ou questionáveis no argumento*, não há problema. Conseguiu-se, nesses casos, reforçar o laço inferencial em vista. Ocorre que nem sempre isso parece ser possível, o que deixa margem para considerar a especificidade dos argumentos indutivos. (SACRINI, 2016, p. 52-53).

Considerando o seguinte exemplo:

Quadro 11 - Exemplo de argumento indutivo

- | |
|--|
| <ol style="list-style-type: none"> 1. O fumo é prejudicial à saúde. 2. O fumo deve ser proibido. (conclusão) |
|--|

Fonte: Construção da autora com base na teoria, 2023.

Estamos diante de um claro exemplo de argumento indutivo, uma vez que, mesmo considerando a premissa (1) como verdadeira, a conclusão (2) não se segue necessariamente. A formulação do argumento não esclarece de maneira exata por que o fumo deveria ser

proibido. Talvez, mesmo sendo prejudicial à saúde, o fumo possa ser aceito em alguns casos por seu caráter recreativo. Em resumo, o vínculo inferencial entre "ser prejudicial à saúde" e "ser proibido" não está estabelecido de modo necessário (SACRINI, 2016). Porém, o acréscimo de uma premissa universal pode facilmente atribuir necessidade ao laço em questão:

Quadro 12 - Exemplo de argumento indutivo com o acréscimo de uma premissa universal

- | |
|--|
| <ol style="list-style-type: none"> 1. O fumo é prejudicial à saúde. 2. Tudo que é prejudicial à saúde deve ser proibido. 3. O fumo deve ser proibido. (conclusão) |
|--|

Fonte: Construção da autora com base na teoria, 2023.

Nesse cenário, ao assumir a verdade das premissas, a conclusão torna-se irrecusável, estabelecendo-se de maneira logicamente válida. Entretanto, essa transformação do laço inferencial indutivo para dedutivo ocorre a um custo elevado, pois envolve a adição de uma premissa universal, que, se avaliada isoladamente, pode se tornar bastante polêmica. Tratando-se de argumentos, é crucial considerar a verdade e/ou aceitabilidade atual das premissas. Diante disso, é provável que esse argumento não seja aceito em muitos contextos de discussão como uma premissa básica. (SACRINI, 2016).

Dessa forma, evidenciamos a possibilidade lógica de transformar argumentos indutivos em dedutivos ao adicionar premissas sólidas que imponham formas válidas de inferência ao argumento considerado. No entanto, tal transformação nem sempre é útil, pois o sentido geral da argumentação, enquanto prática discursiva em contextos de discussões concretas, visa justificar teses não óbvias por meio de argumentos convincentes. Esse objetivo não pode ser alcançado partindo de premissas falsas ou problemáticas, e, consequentemente, nem todo argumento indutivo deve ser considerado falho por não ser transformado em dedutivo, uma vez que esses argumentos têm sua especificidade, justificando seu ponto com diferentes graus de força inferencial.

Em síntese, alguns padrões indutivos de inferências não sustentam a conclusão com necessidade e embasamento lógico, mas ainda assim podem justificá-la de maneira forte. Para isso, é fundamental explorar métodos chamados de esquemas inferenciais, que constituem os padrões indutivos de inferências, como o silogismo estatístico, a generalização e o argumento por analogia. (WALTON, 2012).

O silogismo estatístico, por exemplo, é um tipo de argumento indutivo que parte de um conjunto de informações sobre uma população ou uma classe de eventos, resultando em

uma conclusão referente a um membro ou a um subconjunto da população ou classe em questão. Esse tipo de argumento utiliza termos estatísticos como "a maioria", "muitos", "quase todos", "uns poucos", "raramente", "quase", "menos", "no mínimo", "nunca", entre outros. Para classificar um argumento indutivo como estatístico, é necessário examinar a conclusão, considerando o contexto do diálogo e a presença de termos estatísticos na conclusão. (WALTON, 2012).

O esquema deste tipo de inferência é:

Quadro 13 - Esquema inferencial ao silogismo estatístico (argumento indutivo)

Uma certa quantidade (ou n%) de F é G x é F x é G (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Aplicando este esquema de argumento indutivo estatístico a alguns exemplos temos (SACRINI, 2016):

Quadro 14 - Exemplos aplicando o esquema de argumento indutivo estatístico

EXEMPLO 1 96% dos votantes daquela seção eleitoral votaram no candidato A. João votou naquela seção. João (muito provavelmente) votou no candidato A. (conclusão)
EXEMPLO 2 A grande maioria dos brasileiros fala português. Pedro é um brasileiro. Pedro (provavelmente) fala português. (conclusão)
EXEMPLO 3 Quase toda quarta-feira José chega às dez da noite em casa. Hoje é quarta-feira. José vai (provavelmente) chegar às dez horas em casa. (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Os exemplos fornecidos representam diferentes tipos de argumentos estatísticos. O primeiro oferece dados estatísticos exatos, enquanto o segundo utiliza uma expressão imprecisa ("a grande maioria") para indicar a classe em questão. Além disso, os dois primeiros exemplos aplicam-se a populações, enquanto o terceiro restringe eventos relevantes a apenas uma pessoa.

No silogismo estatístico, o dado geral inicial nunca é universal. Em outras palavras, uma parte da classe de eventos em questão possui uma característica que se pretende atribuir a

um membro ou subconjunto dessa classe. Dessa forma, essa atribuição é indutiva, pois a possibilidade de o membro ou subconjunto em questão não compartilhar da característica destacada não pode ser descartada com base nos dados iniciais. O grau de força dessa inferência depende principalmente do escopo coberto pelas informações iniciais em relação à premissa geral do argumento. Se a premissa geral refere-se à quase totalidade ou quase nulidade da população ou classe em questão, a inferência torna-se forte; caso contrário, a relação inferencial torna-se duvidosa. (SACRINI, 2016).

A generalização é outro tipo de argumento indutivo que acumula informações sobre indivíduos para explicitar características que se aplicariam ao conjunto ou classe desses indivíduos. Essencialmente, busca-se tirar conclusões abrangentes com base em observações específicas. (SACRINI, 2016). Por exemplo,

suponha que eu esteja na biblioteca examinando livros em diversas estantes da sala de obras de referência e observe que todos têm um número de catálogo que começa com R. Eu concluo então, por generalização indutiva, que a maior parte ou todos os livros catalogados na sala de obras de referência têm números de catálogo que começam com R. Baseio minha premissa na observação de alguns poucos livros, um conjunto específico de livros, estendendo a conclusão ao grupo maior formado por todos os livros da sala de obras de referência. (WALTON, 2012, p. 278).

O exemplo fornecido, mesmo se considerarmos a premissa verdadeira, parece insuficiente para estabelecer a conclusão proposta. Determinar a amostra adequada para fazer uma generalização confiável sobre os catálogos da sala de obras de referência da biblioteca é uma tarefa desafiadora. No entanto, é mais fácil apontar quando a amostra é frágil diante do amplo escopo abrangido pela conclusão.

De maneira geral, para aceitar um argumento indutivo de generalização, é necessário compreender minimamente seus componentes e o procedimento adotado, a fim de avaliar as ideias que compõem essa generalização.

O argumento por analogia, por sua vez, é composto por premissas que destacam similaridades entre dois ou mais indivíduos ou eventos. A conclusão, que envolve aspectos similares não imediatamente óbvios, emerge dessas similaridades conhecidas. Esse tipo de argumento busca estabelecer uma relação entre situações diferentes com base nas semelhanças identificadas. (SACRINI, 2016).

O esquema inferencial para esse tipo de argumento ocorre em duas versões:

Quadro 15 - Versões 1 e 2 de esquema inferencial de argumento por analogia

VERSÃO 1	VERSÃO 2
a, b, c, d têm os atributos P e Q. a, b, c têm o atributo R. d tem também o atributo R. (conclusão)	a, b, c, d têm os atributos P e Q. a, b, c são F. d é F também. (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Agora aplicaremos as ideias do esquema inferencial em um exemplo (SACRINI, 2016), para demonstrar a operação do argumento,

Quadro 16 - Exemplo 1 aplicando o esquema de argumento por analogia

EXEMPLO 1
Esses dois insetos observados voam com zumbido e são vermelhos. Ambos são vespas. Esse outro inseto que agora surge também voa com zumbido e é vermelho. Ele também é uma vespa. (conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Como evidenciado, a identificação de semelhanças já conhecidas entre indivíduos permite concluir sobre outra semelhança existente entre eles. No exemplo apresentado, as similaridades entre dois insetos previamente conhecidos e um terceiro, que surge posteriormente, possibilitam classificar esse último como pertencente à mesma espécie dos primeiros.

É importante destacar que qualquer inferência analógica pode ser questionada, argumentando-se que a transição lógica de atributos conhecidos para desconhecidos é arbitrária e sujeita a inúmeros contra exemplos. Em contrapartida, é necessário considerar que todo argumento analógico pressupõe uma regra fundamental: "indivíduos ou eventos semelhantes em aspectos observados são, comumente, semelhantes em aspectos não observados". Essa regra exclui significativamente a possibilidade de existirem contra exemplos na transição lógica das semelhanças conhecidas entre indivíduos para uma semelhança não evidentemente conhecida. Portanto, o esquema inicial do argumento analógico deve ser complementado com o princípio da analogia, isto é, "uma premissa implícita de regra pela qual as demais premissas do argumento analógico (premissas de caso) são conectadas de maneira significativa" (SACRINI, 2016, p. 189). Apesar de muitas vezes não ser explicitamente mencionado no uso cotidiano desse argumento, esse princípio é essencial para a validade do raciocínio analógico (SACRINI, 2016).

Quadro 17 - Versão 3 de esquema inferencial de argumento por analogia

VERSÃO 3	
a, b, c, d têm os atributos P e Q.	
a, b, c têm o atributo R (ou: 2. a, b, c são F).	
Indivíduos ou eventos similares em aspectos observados são, comumente, similares em aspectos não observados.	
d tem também o atributo R (ou: 3. d é F também).	(conclusão)

Fonte: Construção da autora com base na teoria, 2023.

Dessa forma, destaca-se que a formulação do princípio da analogia é relativamente flexível, pois não pretende ser aplicável a todos os casos possíveis. Isso é evidenciado pela inclusão do advérbio "comumente", indicando que o princípio se aplica em muitos casos, mas não em todos. Essa adição explícita reconhece que o argumento por analogia é, essencialmente, um movimento inferencial indutivo (SACRINI, 2016).

A conclusão de um argumento por analogia não é estabelecida de maneira definitiva pela transição das similaridades conhecidas para uma similaridade desconhecida. Este tipo de argumento reconhece suas limitações, sendo aceito com diferentes graus de convencimento. No entanto, a avaliação de uma inferência analógica forte envolve três aspectos, conforme proposto por Sacrini

- a. cita nas premissas um número considerável de indivíduos similares em vários aspectos. No exemplo acima, tentou-se concluir que o inseto era um vespa com base em dois outros casos de insetos semelhantes. Se fossem mencionados quatro ou cinco insetos, com várias características semelhantes ao inseto mencionado na conclusão (um determinado tamanho, um determinado padrão de voo, um determinado comportamento na presença de pessoas etc.), a inferência analógica seria fortalecida.
- b. Cita similaridades relevantes para o estabelecimento de conclusão, de maneira a evitar aspectos triviais ou que não tenham nenhuma relação com o que se almeja afirmar na conclusão. No exemplo acima, de nada adiantaria asseverar que os insetos observados têm seis patas, bem como o inseto apontado na conclusão, pois é óbvio que todos os insetos possuem normalmente seis patas. Essa informação em nada contribui para justificar a conclusão, que tenciona a estabelecer qual tipo de inseto é aquele em questão e, assim, já pressupõe como dadas certas características evidentes acerca de insetos.
- c. Cita indivíduos nas premissas que, apesar de partilharem de características similares, são variados a ponto de conter importantes dessemelhanças. Essa consideração não é imediatamente óbvia, mas é uma das que mais permitem revelar a força da inferência analógica. O ponto em questão é que, quanto mais circunstâncias diversas forem incluídas nas premissas, menor é a chance de que a inferência seja criticável à luz de possíveis situações em que o laço lógico não se estabeleceria. (SACRINI, 2016, p. 189-190).

Assim, o argumento por analogia procura adquirir conhecimento sobre um membro de uma classe com base no conhecimento de alguns outros membros da mesma classe. Esse

método inferencial é útil para formular conclusões restritas em situações em que faltam dados abrangentes.

3.2.3 *Argumento dedutivo*

Os fundamentos essenciais de uma argumentação sólida residem nas proposições. "As proposições, ao contrário de perguntas, comandos, desafios e outros elementos da argumentação, são unidades de linguagem verdadeiras ou falsas" (WALTON, 2012, p. 151). Um argumento é considerado dedutivo apenas se a conclusão estiver necessariamente fundamentada nas proposições iniciais, tornando-o assim válido (SACRINI, 2016). Dessa forma, "o argumento dedutivo é aquele em que, se assumirmos a verdade das premissas, a conclusão segue necessariamente, atestando sua validade lógica" (SACRINI, 2016, p. 51-52).

A forma de um argumento pode ser representada por meio do uso de símbolos. Para cada forma de argumento, há uma declaração correspondente chamada condicional correspondente. Uma forma de argumento é válida "se e somente se" seu correspondente condicional for uma verdade lógica. A declaração é considerada uma forma lógica de verdade se for verdadeira em todas as interpretações. Uma forma de declaração pode ser comprovada como uma verdade lógica por meio de um argumento que a valida como uma tautologia (SACRINI, 2016).

A dedução, ou método dedutivo, parte de uma lei ou teoria geral para inferir conclusões relacionadas a casos específicos. Utilizando premissas validadas pela ciência e consideradas verdadeiras, o cientista formula verdades específicas. Assim, a conclusão só é considerada correta se todas as premissas forem verdadeiras. Portanto, o argumento dedutivo propõe premissas, segue uma razão, critérios e testa cada uma delas para validar as conclusões, justificando as informações por meio de linguagem natural (SACRINI, 2016).

Um argumento dedutivo, por exemplo, é válido apenas se suas premissas têm uma relação de necessidade com sua conclusão: assumindo as premissas como verdadeiras, é necessário admitir também a veracidade da conclusão. Isso implica que explicar, justificar e demonstrar conclusões são ações de grande importância no desenvolvimento do raciocínio lógico, ocorram elas dentro ou fora do contexto da Matemática (WESTON, 2009). "Mesmo quando as premissas são incertas, a forma dedutiva é um modo eficaz de organizar o argumento" (WESTON, 2009, p. 54).

Para construir ou avaliar argumentos, é de extrema importância seguir os esquemas das principais formas válidas de argumento dedutivo, com base em Weston (2009) e Walton (2012):

Quadro 18 - Esquema das quatro formas válidas de argumento dedutivo

<i>Modus ponens (MP)</i> Se [proposição A], então [proposição B]. [Proposição A]. Portanto, [proposição B].	<i>Modus tollens (MT)</i> Se [proposição P], então [proposição Q]. [Proposição Q] é falso. Portanto, [proposição P] é falso
<i>Silogismo hipotético (SH)</i> Se [proposição R], então [proposição S]. Se [proposição S], então [proposição T]. Portanto, se [proposição R], então [proposição T]	<i>Silogismo disjuntivo (SD)</i> Ou [proposição M] ou [proposição N]. [Proposição M] é falso. Portanto, [proposição N].

Fonte: Construção da autora com base na teoria, 2023.

Com base no seguinte argumento:

Quadro 19 - Exemplo de argumento dedutivo utilizando *Modus ponens (MP)*

Se não existe o fator acaso no xadrez, então o xadrez é um jogo de pura habilidade. Não existe o fator acaso no xadrez. Portanto, o xadrez é um jogo de pura habilidade.
--

Fonte: Construção da autora com base na teoria, 2023.

A forma chamada *modus ponens (MP)* é uma maneira válida de argumentar, afirmando A para obter B (WESTON, 2009). Se A representar "não existe o fator acaso no xadrez" e B "o xadrez é um jogo de pura habilidade", então o exemplo segue MP, confirmando um argumento dedutivo. Para desenvolver um argumento, é necessário "explicar e defender ambas as premissas, e elas exigem argumentos bastante distintos (por quê?)" (WESTON, 2009, p. 55), expondo-as com clareza, firmeza e separadamente.

A segunda maneira dedutiva válida de argumentar é o *modus tollens (MT)*, que é um modo de negar, ou seja, negando-se Q, nega-se P (WESTON, 2009). Neste caso, "Q é falso" representa somente a negação de Q, ou seja, está no lugar da frase "não é verdade que Q", e o mesmo ocorre com "P é falso".

Para representar MT, Weston (2009, p. 56) nos traz uma pequena história: "havia um cachorro nos estábulos, e, mesmo assim, embora alguém tenha estado lá e levado um cavalo, o animal não latiu. É evidente que o visitante era alguém que o cachorro conhecia bem...". E argumenta em MT: "Se não conhecesse bem o visitante, o cachorro teria latido. O cachorro não latiu. Portanto, o cachorro conhecia bem o visitante."

Se escrevermos este argumento utilizando o esquema de validade de argumentos dedutivos, é possível identificar as proposições, justificativas e a conclusão de forma clara, tornando o argumento válido. Usando P no lugar de "o cachorro **não** conhecia bem o visitante" e Q para "o cachorro latiu", "Q é falso" significa "o cachorro não latiu, e "P é falso" representa "não é verdade que o cachorro não conhecia bem o visitante", ou seja, o cachorro conhecia bem o visitante, por isso não latiu (WESTON, 2009).

Uma terceira forma dedutiva válida de argumentar é o silogismo hipotético (SH), que através de seu modelo busca explicar a conexão entre causa e efeito, ou seja, a conclusão liga uma causa a um efeito, ao mesmo tempo que as premissas explicam as etapas intermediárias (WESTON, 2009). Os SH são válidos para qualquer número de premissas, desde que cada premissa obedeça à forma "se R então S" e o R de uma premissa seja o S da seguinte (WALTON, 2012). Um exemplo é o seguinte: "se estudar outras culturas, você compreenderá a diversidade dos costumes humanos. Se compreender a diversidade de costumes humanos, questionará seus próprios costumes. Portanto, se estudar outras culturas, você questionará os próprios costumes."

Ao inserir as letras em negrito e sublinhadas que compõem este argumento no esquema de validade de argumentos dedutivos, identificamos as ligações entre causa e efeito, bem como o elo existente que fomenta a conclusão, visto que temos: se **e**, então **c**. Se **c**, então **q**. Portanto, se **e**, então **q**.

A quarta e última maneira dedutiva válida de argumentar é o silogismo disjuntivo. Trata-se de uma disjunção, que pelo menos uma das partes tem que ser verdadeira, no entanto se ambas forem falsas, ou ambas verdadeiras, não se trata de uma verdadeira disjunção, visto que,

usamos "ou" num sentido "exclusivo", no qual "**p** ou **q**" significa que ou **p**, ou **q** é verdadeiro, mas não ambos. "Eles virão por terra ou por mar", por exemplo, significa que eles não utilizarão as duas vias ao mesmo tempo. Nesse caso, somos capazes de inferir que, se vierem de um modo, não virão de outro. (WESTON, 2009, p. 59-60).

Em outras palavras, no caso do silogismo disjuntivo (SD), se afirmarmos que "Bob está ou com sarampo ou com catapora", reconhecemos que pelo menos uma das proposições é verdadeira, pois ambas não podem ser falsas simultaneamente. Portanto, para validar o argumento por SD, é necessário que pelo menos uma das hipóteses seja verdadeira ou falsa.

Fávero et al. (2017) apontam que em nossas conversas cotidianas, normalmente não usamos argumentos tão bem organizados como os que vimos até agora. Frequentemente,

apresentamos a conclusão antes das premissas ou até mesmo as inserimos no meio delas. O mais importante, porém, é saber identificar onde estão as premissas e a conclusão. Para isso, existem os indicadores de inferência, que se dividem em dois tipos: indicadores de premissas e indicadores de conclusão.

Quadro 20 - Indicadores de premissas em um argumento

Indicadores de premissas		
Pois	Admitindo que	Dado que
Desde que	Isto é verdade que	Sabendo-se que
Como	A razão é que	Supondo que
Porque	Em vista de	Contudo
Assumindo que	Como consequência de	Além disso (do mais)
Visto que	Como mostrado pelo fato de que	

Fonte: Adaptação de Fávero et al. (2017).

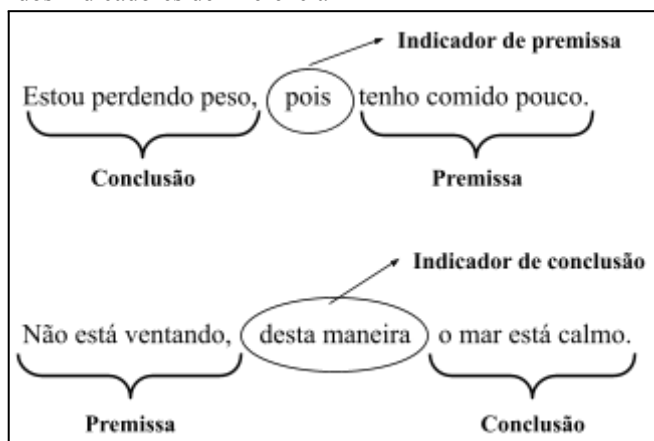
Quadro 21 - Indicadores de conclusão em um argumento

Indicadores de conclusão		
Portanto	De modo que	O(a) qual implica que
Por conseguinte	Então	O(a) qual acarreta que
Assim	Consequentemente	O(a) qual prova que
Dessa maneira	Assim sendo	O(a) qual significa que
Neste caso	Segue-se que	Do(da) qual inferimos que
Daí	Resulta que	Podemos deduzir que
Logo	Isso prova que	

Fonte: Adaptação de Fávero et al. (2017).

Para Fávero et al. (2017), se um indicador aparecer em uma frase, isso significa que uma premissa virá em seguida e que, antes dele, foi apresentada uma conclusão. O mesmo ocorre com o indicador de conclusão: depois dele, vem uma conclusão e, antes dele, foi apresentada uma premissa, como apresentado na seguinte figura.

Figura 4 - Exemplos de argumento dedutivo utilizando-se dos indicadores de inferência



Fonte: Adaptação de Fávero et al. (2017).

Nesse contexto, é evidente que os argumentos podem ser tanto válidos quanto inválidos. Se um argumento é válido, e suas premissas são verdadeiras, então a conclusão deve ser verdadeira. Isso porque "as premissas e a conclusão de um argumento são proposições, que são verdadeiras ou falsas" (WALTON, 2012, p. 183).

A validade de um argumento, no entanto, está intrinsecamente ligada à verdade ou falsidade de suas premissas e conclusões. "As premissas são as proposições que sustentam a conclusão de um argumento ou apresentam razões para ela" (WALTON, 2012, p. 183). No entanto, é importante destacar que apenas o argumento possui uma forma lógica, e a validade desse argumento não garante a verdade de sua conclusão. Isso ocorre porque um argumento válido pode ter premissas falsas e uma conclusão falsa, conforme

a validade ou não de um argumento é determinada por alguns termos-chave chamados constantes lógicas, o que significa que podem ser claramente definidos de maneira fixa ou constante. [...] As constantes foram os conectivos proposicionais "e", "não", "ou" e "se...então". [...] O argumento que tem uma forma válida é necessariamente um argumento válido. Logo, as formas válidas do argumento são elementos semânticos que ajudam na construção de novos argumentos e na reconstrução de antigos argumentos em sequências pragmáticas de argumentação. (WALTON, 2012, p. 183-184).

O argumento dedutivo está intrinsecamente vinculado à lógica, que é um ramo da semântica. Isso se deve ao fato de que a validade das formas MP, MT, HS e DS é definida pelo significado das palavras conectivas, como "não", "ou" e "se...então". Esses símbolos estabelecem relações diretas com a verdade e a falsidade das proposições. Esses conectivos determinam quais formas de argumento são válidas com base nas regras que regem as

relações de verdade e falsidade. Ao seguir essas regras, é possível comprovar a validade de uma determinada forma de argumento (WALTON, 2012, p. 184).

Ao conectar argumentos em um laço dedutivo, cria-se uma forma de prova que é absolutamente inquestionável. Cada passo desse processo se origina do passo anterior na sequência de deduções definidas em MP, MT, HS e DS. A lógica dedutiva, quando aplicada corretamente aos argumentos, elimina espaço para disputas, controvérsias ou contra exemplos, uma vez que, ao aceitar as premissas e com o argumento sendo válido, a conclusão segue inexoravelmente (WESTON, 2009; WALTON, 2012, p. 184).

A lógica dedutiva também oferece proteção contra a contradição, pois não permite conjuntos de proposições incoerentes. Contra exemplos são argumentos equivalentes que compartilham a mesma forma do argumento em questão, mas possuem premissas verdadeiras e uma conclusão falsa, evidenciando inconsistência e incoerência. Embora a incoerência não seja, em si, uma falácia, se o conjunto de compromissos de um argumentador parecer coletivamente incoerente, ele pode e deve ser desafiado a defender sua posição, removendo ou explicando a aparente incoerência (WALTON, 2012, p. 186).

Portanto, a forma de argumento dedutivo pode ser representada por meio de símbolos. Ao avaliar argumentos com esquemas de decomposição e divisão, é crucial estar atento à distinção entre o uso coletivo e o uso distributivo dos termos. Cada forma de argumento tem uma correspondência condicional, e um argumento é válido se e somente se o seu correspondente condicional é uma verdade lógica. A validade de uma forma de declaração pode ser demonstrada por meio de um argumento que mostra tratar-se de uma tautologia, apresentando uma prova convincente (WALTON, 2012, p. 186).

3.3 Considerações parciais

Diante deste desfecho buscou-se responder duas perguntas neste capítulo: Qual o vínculo existente entre raciocínio lógico e argumentação dedutiva? Quais as articulações necessárias para ocorrer conexão entre as premissas e a conclusão na construção de um argumento dedutivo?

O vínculo direto entre raciocínio lógico e argumentação dedutiva reside na aplicação de princípios fundamentais da lógica para derivar conclusões coerentes e consistentes a partir de premissas estabelecidas. O raciocínio lógico é a habilidade de pensar de forma estruturada, evitando contradições, enquanto a argumentação dedutiva é uma expressão específica desse

raciocínio, onde a conclusão segue necessariamente das premissas, obedecendo a regras definidas.

No âmbito do raciocínio lógico, busca-se a coerência e consistência, evitando contradições, e na argumentação dedutiva, utiliza-se regras de inferência, como *Modus Ponens*, *Modus Tollens*, Silogismo Hipotético e Silogismo Disjuntivo, para estabelecer a validade de um argumento.

As articulações específicas são fundamentais para conectar as premissas à conclusão de forma lógica. No *Modus Ponens*, se afirmamos que "se A, então B" (premissa 1), e sabemos que "A" é verdadeiro (premissa 2), concluímos necessariamente que "B" também é verdadeiro. O *Modus Tollens*, por sua vez, inverte esse processo, negando a conclusão para validar a negação da premissa condicional.

O Silogismo Hipotético constrói uma cadeia de condições, conectando premissas de modo que, se a primeira implica na segunda e a segunda implica na terceira, então a primeira implica diretamente na terceira. O Silogismo Disjuntivo, por sua vez, estabelece que se uma disjunção é verdadeira e uma das opções é falsa, a outra deve ser verdadeira.

Essas articulações são essenciais para garantir a validade de argumentos dedutivos, proporcionando uma conexão lógica clara entre as premissas e a conclusão. Elas servem como ferramentas para raciocínio válido, assegurando que, ao aceitar as premissas como verdadeiras, a conclusão inevitavelmente segue, reforçando a fundamentação lógica desses argumentos.

4 PENSAMENTO COMPUTACIONAL

Em todos os setores da sociedade, é evidente a presença e o impacto da tecnologia na vida diária das pessoas, especialmente dos adolescentes. O rápido avanço e a constante evolução das tecnologias têm reconfigurado o mundo, e nesse contexto, a computação, seja por meio de *smartphones*, internet, *tablets* ou computadores, tornou-se uma ferramenta valiosa para o trabalho e o estudo. Seguindo essa perspectiva, Resnick destaca que,

os jovens de hoje serão confrontados com situações novas e inesperadas durante toda a vida. Eles precisam aprender a lidar com as incertezas e mudanças usando a criatividade, não só em suas vidas profissionais, mas também nos âmbitos pessoal (como desenvolver e manter amizades em uma era de redes sociais) e cívico (como ter uma participação significativa em comunidades com limites e necessidades em constante mudança). (RESNICK, 2020, p. 4).

Portanto, a contextualização da tecnologia, a programação de computadores e a sua integração no ambiente escolar representa uma interseção entre as inovações digitais e a educação. A presença cada vez mais significativa da tecnologia no cotidiano das pessoas, aliada à constante evolução das ferramentas digitais, destaca a importância de incorporar conhecimentos de programação no ambiente educacional.

A tecnologia, em suas diversas formas, como computadores, *tablets* e *smartphones*, tornou-se onipresente na sociedade contemporânea. Esses dispositivos oferecem não apenas recursos de comunicação, mas também acesso a vastos recursos de aprendizado e informações. Nesse contexto, a introdução da programação de computadores nas escolas assume um papel fundamental, assim como Papert se posiciona,

o computador é a ferramenta que propicia à criança as condições de entrar em contato com algumas das mais profundas idéias em ciência, matemática e criação de modelos. [...] O computador deixa de ser o meio de transferir informação, e passa a ser a ferramenta com a qual a criança pode formalizar os seus conhecimentos intuitivos. O programa (a seqüência de ações ao computador) que a criança elabora é o espelho que reflete o seu conhecimento sobre um determinado assunto e o seu estilo de pensamento. Este programa, quando usado como objeto de reflexão, se torna uma poderosa fonte de aprendizagem. (1985, p.9-10).

Ao ensinar programação nas escolas, proporcionamos aos alunos as habilidades necessárias para compreender e interagir de maneira mais eficaz com a tecnologia que os cerca. A programação não se limita apenas à capacidade de criar *software*; ela desenvolve habilidades cognitivas, como pensamento lógico, resolução de problemas e criatividade. Essas

competências são essenciais em um mundo onde a tecnologia desempenha um papel cada vez mais central em diversas profissões e setores.

Além disso, a introdução da programação no currículo escolar promove a inclusão digital, capacitando os alunos a não apenas consumir tecnologia, mas também a compreender e contribuir ativamente para o seu desenvolvimento. Isso contribui para a formação de uma geração mais preparada para os desafios e oportunidades do mundo digital, visto que,

não basta adquirir a máquina, é preciso aprender a utilizá-la, a descobrir as melhores maneiras de obter da máquina auxílio nas necessidades de seu usuário. É preciso buscar informações, realizar cursos, pedir ajuda aos mais experientes, enfim, utilizar os mais diferentes meios para aprender a se relacionar com a inovação e ir além, começar a criar novas formas de uso e, daí, gerar outras utilizações. Essas novas aprendizagens, quando colocadas em prática, reorientam todos os nossos processos de descobertas, relações, valores e comportamentos. (KENSKI, 2012, p. 44).

Assim, a contextualização da tecnologia e da programação de computadores na escola não apenas acompanha as demandas da sociedade moderna, mas também prepara os estudantes para enfrentar um futuro no qual a proficiência tecnológica é uma competência essencial, e neste contexto os Parâmetros Curriculares Nacionais (PCN), que recomendam a utilização desses recursos tecnológicos no ensino como um meio simplificador para o avanço das habilidades dos estudantes, já esclarecia anos atrás, sobre a crescente necessidade do emprego de computadores pelos estudantes como ferramenta no processo de aprendizagem escolar. Isso possibilita que estejam atualizados em relação às novas tecnologias da informação e se capacitem para enfrentar as demandas sociais atuais e futuras (BRASIL, 1998).

A evolução tecnológica produziu muitas mudanças em nossa sociedade, influenciando também a educação. Para tanto, o pensamento computacional é uma realidade significativa e que precisa ser considerada pelas escolas, com esse propósito busca-se identificar articulações entre o desenvolvimento do pensamento computacional e a construção de argumentos dedutivos, através do seguinte questionamento: Quais as conexões do pensamento computacional na formação de um ser argumentativo que resolve problemas quando programa computadores?

Para tanto, este capítulo se propõe a explorar recursos digitais, com especificidade a programação de computadores em ambiente educacional, centrando-se nos fundamentos do pensamento computacional. Mais do que uma habilidade técnica, o pensamento computacional engloba capacidades cognitivas essenciais, como raciocínio lógico, resolução de problemas e criatividade, bem como, desdobrar seus pilares - decomposição,

reconhecimento de padrões, abstração e algoritmos - buscamos não apenas elucidar esses conceitos, mas também destacar sua aplicabilidade prática no contexto educacional.

4.1 Explorando o pensamento computacional

A maneira como vivemos e aprendemos tem passado por transformações significativas devido à tecnologia. Isso é evidenciado pela abundância de recursos e ferramentas digitais criadas, especialmente para facilitar o processo de ensino aprendizagem. Plataformas digitais, jogos, aplicativos e *softwares* educacionais são algumas das opções disponíveis para os educadores inovarem em suas aulas, tornando-as envolventes, interativas e atrativas. Como afirmou Papert, essas tecnologias oferecem oportunidades para uma abordagem mais interessante no campo educacional, visto que

a questão central da mudança na Educação é a tensão entre a tecnicização e a não tecnicização, e aqui o professor ocupa a posição central. Desde a criação da máquina de imprimir não houve tão grande impulso no potencial para encorajar a aprendizagem tecnicizada. Há, porém, outro lado: paradoxalmente, a mesma tecnologia possui o potencial de destecnicizar a aprendizagem. Se isto ocorresse, eu contaria com uma mudança muito maior do que o surgimento, em cada carteira, de um computador programado para conduzir o estudante através de passos do mesmo velho currículo. Contudo, não é necessário sofismar sobre que mudança tem o maior alcance. O que é necessário é reconhecer que a grande questão no futuro da Educação é se a tecnologia fortalecerá ou subverterá a tecnicidade do que se tornou o modelo teórico e, numa grande extensão, a realidade da Escola. (PAPERT, 2008, p.55).

O progresso tecnológico gerou diversas mudanças em nossa sociedade, incluindo o âmbito educacional. Nesse contexto, a educação passa por atualizações contínuas, visando manter o ensino alinhado às demandas contemporâneas de aprendizagem. Nessa perspectiva, Robinson destaca que,

no século XXI, a humanidade enfrenta alguns de seus maiores desafios. Nossa melhor solução é cultivar nossos talentos de imaginação, criatividade e inovação. Esta deveria ser uma das maiores prioridades da educação no mundo todo. A educação é a solução para o futuro e os riscos dificilmente poderiam ser maiores. (ROBINSON, 2019, p. 54).

A relação entre educação e tecnologia tem raízes antigas e foi desenvolvida ao longo de períodos distintos. A introdução do computador, em particular, indicou mudanças significativas na instrumentalização educacional. O que costumava ser feito manualmente em papel passou a ser realizado de maneira automatizada, por meio de programas específicos

para essas tarefas. Portanto, torna-se essencial a incorporação de computadores no ambiente escolar, possibilitando seu uso em sala de aula, uma vez que

o computador permite cada vez mais pesquisar, simular situações, testar conhecimentos específicos, descobrir novos conceitos, lugar e ideias. Com a internet pode-se modificar mais facilmente a forma de ensinar e aprender. Procurar estabelecer uma relação de empatia com os alunos, procurando conhecer seus interesses, formação e perspectivas para o futuro. É importante para o sucesso pedagógico a forma de relacionamento professor/aluno (MORAN; MASETTO; BEHRENF, 2000. p. 16).

Neste viés, surge a "segunda revolução" nos métodos e processos de ensino aprendizagem com a chegada da internet. Trata-se de um universo virtual que oferece um amplo fluxo de conectividade, informações e comunicação, visando simplificar e auxiliar nas atividades cotidianas por meio *de softwares* interativos.

Ao considerar a utilização da tecnologia no ambiente escolar percebe-se um processo essencial para alinhar a educação com as demandas e oportunidades da sociedade contemporânea. A presença universal da tecnologia no cotidiano das pessoas, juntamente com seu papel transformador em diversos setores, destaca a importância de integrar efetivamente essas inovações no contexto educacional, visto que

o ritmo acelerado de inovações tecnológicas exige um sistema educacional capaz de estimular nos estudantes o interesse pela aprendizagem. E que esse interesse diante de novos conhecimentos e técnicas seja mantido ao longo da sua vida profissional, que, provavelmente, tenderá a se realizar em áreas diversas de uma atividade produtiva cada vez mais sujeita ao impacto das novas tecnologias (SANCHO, 1998. p. 41).

Ao integrar a tecnologia no ambiente escolar, o aprendizado se torna mais dinâmico e próximo da realidade dos alunos, que, em grande parte, já têm contato frequente com dispositivos eletrônicos desde cedo. Isso facilita o engajamento e a participação ativa no processo de aprendizagem, tornando-o mais relevante e interessante para eles. Essa abordagem não apenas torna as aulas mais envolventes, mas também prepara os estudantes para uma participação mais efetiva em um mundo digital em constante evolução, visto que

[...] a educação e tecnologias são indissociáveis. Segundo o dicionário Aurélio, a educação diz respeito ao “processo de desenvolvimento da capacidade física, intelectual e moral da criança e do ser humano em geral, visando à sua melhor integração individual e social”. Para que ocorra essa integração, é preciso que conhecimentos, valores, hábitos, atitudes e comportamentos do grupo sejam ensinados e aprendidos, ou seja, que se utilize a educação para ensinar sobre as tecnologias que estão na base da identidade e da ação do grupo e que se faça uso delas para ensinar as bases dessa educação. (KENSKI, 2012, p. 43).

A tecnologia na escola também contribui para personalizar o processo de aprendizagem, permitindo adaptações conforme as necessidades individuais dos alunos. Ferramentas como *softwares* de aprendizado adaptativo podem fornecer um suporte mais direcionado, auxiliando alunos com ritmos de aprendizado diferentes. Além disso, a integração da tecnologia no ambiente escolar apoia o desenvolvimento de habilidades essenciais no século XXI, como pensamento crítico, resolução de problemas, colaboração e comunicação. Preparar os alunos para enfrentar os desafios de um mundo cada vez mais tecnológico é uma responsabilidade da educação contemporânea, e para isso

precisamos oferecer às crianças mais oportunidades de realizar explorações lúdicas, tanto com materiais físicos quanto com materiais digitais. O processo de explorações pode ser confuso e tortuoso, mas qualquer processo criativo é assim. Um plano cuidadoso pode ter resultados eficazes, mas não podemos planejar a criatividade. O pensamento criativo é resultado de explorações criativas. (RESNICK, 2020, p. 128 - 129).

No contexto das tecnologias digitais, novas formas de pensar e interagir estão sendo constantemente concebidas. As relações entre as pessoas, o trabalho, e até a própria inteligência estão profundamente ligadas à contínua evolução dos dispositivos digitais de diversas naturezas. A escrita, a leitura, a visão, a audição, a criação e a aprendizagem são cada vez mais moldadas por tecnologias digitais avançadas (LÉVY, 2010). Portanto, a introdução da tecnologia no ambiente escolar não apenas acompanha as mudanças da sociedade, mas também contribui para uma educação mais envolvente e adaptada às necessidades dos alunos em um mundo cada vez mais digitalizado. Isso proporciona oportunidades de aprendizado mais dinâmicas e flexíveis, que refletem o contexto atual e preparam os alunos para os desafios da era digital. Para tanto, a introdução da programação de computadores na escola é uma iniciativa educacional inovadora que visa preparar os alunos para os desafios do século XXI, uma vez que

eles precisam aprender a lidar com as incertezas e mudanças usando a criatividade, não só em suas vidas profissionais, mas também nos âmbitos pessoal (como desenvolver e manter amizades em uma era de redes sociais) e cívico (como ter uma participação significativa em comunidades com limites e necessidades em constante mudança). (RESNICK, 2020, p. 4).

A programação, muitas vezes considerada uma habilidade essencial no mundo digital atual, oferece benefícios significativos para o desenvolvimento cognitivo, a resolução de problemas e o pensamento lógico dos estudantes. Embora a maioria das pessoas não busque

se tornar jornalista ou escritor profissional, é crucial que todos adquiram habilidades de escrita. O mesmo princípio se aplica à programação, por motivos semelhantes. Mesmo que a maioria não siga uma carreira como programador profissional ou cientista da computação, a fluência em programação é uma habilidade valiosa para todos. Tornar-se proficiente, tanto na escrita quanto na programação, contribui para o desenvolvimento do pensamento, aprimora a expressão pessoal e fortalece a identidade individual (RESNICK, 2020). Neste contexto, a inclusão da programação no currículo escolar não se limita apenas à formação de futuros programadores, mas também proporciona uma compreensão mais profunda do mundo tecnológico que nos cerca.

Ao integrar a programação no ambiente escolar, os alunos são expostos a uma abordagem prática e interativa do aprendizado, promovendo o desenvolvimento de habilidades essenciais, como criatividade, trabalho em equipe e perseverança diante de desafios. Além disso, a programação incentiva a resolução de problemas de maneira estruturada, estimulando a lógica e a capacidade analítica. Para Papert,

os cidadãos do futuro precisam lidar com desafios, enfrentar um problema inesperado para o qual não há uma explicação preestabelecida. Precisamos adquirir habilidades necessárias para participar da construção do novo ou então nos resignarmos a uma vida de dependência. A verdadeira habilidade competitiva é a habilidade de aprender. Não devemos aprender a dar respostas certas ou erradas, temos de aprender a solucionar problemas. (2008, p.122).

A programação proporciona uma base sólida para compreender conceitos tecnológicos, fortalecendo a capacidade dos estudantes de participar ativamente em uma sociedade cada vez mais orientada pela tecnologia, mas esta introdução da programação no ambiente escolar não apenas equilibra a equação de inclusão digital, mas também contribui para uma educação mais abrangente e relevante. À medida que a sociedade avança em direção à automação e à digitalização, a habilidade de programar se torna uma ferramenta valiosa para a resolução de problemas complexos, a criação de soluções inovadoras e o entendimento crítico do mundo ao nosso redor. Neste aspecto, Almeida destaca que,

o conhecimento não é fornecido ao aluno para que ele dê as respostas. É o aluno que coloca o conhecimento no computador e indica as operações que devem ser executadas para produzir as respostas desejadas. O programa fornece importantes pistas sobre o pensamento do aluno, uma vez que o pensamento está descrito explicitamente e a resposta do computador permite comparar o previsto com o obtido. (2000, p. 33-34).

Essa inovadora ferramenta deve ser empregada como um meio para promover a reflexão crítica, o pensamento lógico, a habilidade na solução de problemas simples e complexos, a interpretação de questões, o trabalho em equipe e o estímulo à curiosidade. Dentro desse contexto, Pólya nos apresenta que

o problema pode ser modesto, mas se desafiar a curiosidade e puser em jogo as faculdades inventivas, quem o resolver pelos seus próprios meios experimentará o prazer e o triunfo da descoberta. Tais experiências, numa idade susceptível, poderão criar o gosto pelo trabalho mental e deixar, por toda a vida, uma marca indelével na mente e no carácter. (2006, Prefácio).

Neste contexto, a integração da programação de computadores no ambiente escolar está intrinsecamente conectada ao desenvolvimento do pensamento computacional. O pensamento computacional é uma habilidade cognitiva que envolve a capacidade de resolver problemas de maneira estruturada, decompor tarefas complexas em passos menores, identificar padrões e abstrair informações. Essa habilidade é fundamental na programação e, ao ser incorporada na educação, promove uma abordagem mais abrangente e eficaz para o ensino. Neste cenário, Papert se posiciona,

[...] uma das características importantes do trabalho com o computador é que o professor e aluno podem engajar-se numa verdadeira colaboração intelectual; juntos, podem tentar fazer com que o computador execute isto ou aquilo, e entender que ele realmente faz. Situações novas que nem o professor nem o aluno viram antes ocorrem frequentemente e assim o professor não tem que fingir que não sabe. Compartilhar o problema e a experiência de resolvê-lo permite a criança aprender com um adulto, não “fazendo o que o professor diz”, mas “fazendo o que o professor faz”. E uma das coisas que o professor faz é persistir num problema até que ele tenha sido completamente compreendido. (1985, p.143).

O estudante, ao cometer erros, é guiado a refletir sobre abordagens alternativas para a resolução do problema, proporcionando-lhe a oportunidade de aprender com suas próprias falhas. Esse cenário estimula a tentativa, uma vez que, ao ocorrer um equívoco, este se transforma em um objeto de análise, identificação e reformulação, desencadeando assim um processo de aprendizado e desenvolvimento. Esse procedimento estabelece um ciclo de descrição e aprimoramento, reflexão e depuração, um modelo que foi adotado na programação de computadores. De acordo com Valente,

o processo de descrever, refletir e depurar não acontece simplesmente colocando o aluno em frente do computador. A interação aluno-computador precisa ser mediada por um profissional que conhece o Logo, tanto do ponto de vista computacional, quanto pedagógico e do psicológico. Esse é o papel do mediador no ambiente Logo. (1998, p.5).

Correia e Silva ainda destacam que,

aqui, o erro deixa de ser uma arma de punição e passa a ser uma situação que nos leva a entender melhor nossas ações e conceitualização. É assim que a criança aprende uma série de conceitos antes de entrar na escola. Ela é livre para explorar e os erros são usados para depurar os conceitos e não para se tornarem a arma do professor. (2005, p.35).

Nesse contexto, atribui-se ao erro um papel significativo no processo de aprendizagem, proporcionando ao aluno a oportunidade de compreender as razões por trás do equívoco e procurar uma nova abordagem para a resolução do problema. Isso implica em uma abordagem investigativa, exploratória e autodidata, alinhada com a ideia de aprendizagem pela descoberta, conforme preconizado pela teoria construtivista de Jean Piaget.

Teixeira (2017) destaca que, nesse contexto, o aluno pode vivenciar experiências formais e aprender com seus erros, resolvendo problemas, concebendo soluções em etapas e tomando decisões. Essas competências, exercitadas na programação de computadores, caracterizam o pensamento computacional. Valente (1998) também enfatiza que o processo de identificar e corrigir erros oferece uma oportunidade única para o aluno aprender sobre um conceito específico envolvido na solução do problema ou sobre estratégias de resolução de problemas.

De acordo com Teixeira (2017, p. 23), o "aprendizado de programação de computadores desenvolve o pensamento". Isso ocorre porque, quando o programa não funciona como esperado, a pessoa retorna à linguagem, depura o programa e, nesse processo, também refina seus próprios pensamentos.

Resnick (2020) nos coloca que, Seymour Papert foi um destacado defensor do uso do computador como ferramenta educacional, introduziu o conceito de construcionismo. Esta abordagem pedagógica sugere que os indivíduos constroem seu próprio conhecimento por meio de ferramentas práticas, como o computador. Inspiradas no construtivismo e fundamentadas em diversos princípios teóricos, as ideias de Papert buscavam maximizar o uso das tecnologias no ensino. Seus estudos defendem que a construção do conhecimento

deve ser baseada na realização de ações concretas, resultando em produtos palpáveis desenvolvidos a partir dos interesses dos alunos.

O construcionismo valoriza a produção de textos, imagens, mapas conceituais ou apresentações em slides que estejam contextualizados. Isto é, esses produtos devem estar conectados à realidade do indivíduo ou ao ambiente onde serão aplicados. Dessa forma, o aprendizado se torna mais relevante e significativo, refletindo a aplicação prática dos conhecimentos adquiridos em contextos reais (RESNICK, 2020). Portanto,

o conhecimento não é fornecido ao aluno para que ele dê as respostas. É o aluno que coloca o conhecimento no computador e indica as operações que devem ser executadas para produzir as respostas desejadas. O programa fornece importantes pistas sobre o pensamento do aluno, uma vez que o pensamento está descrito explicitamente e a resposta do computador permite comparar o previsto com o obtido (2000, p. 33-34)

Papert (2008) argumenta que essa é a característica mais importante do construcionismo, pois facilita a formação cognitiva. Ele defende o uso do computador como uma ferramenta essencial no processo de construção do conhecimento. Adaptando os princípios do construtivismo cognitivo de Jean Piaget, Papert acredita que a tecnologia pode ser utilizada de maneira mais eficaz para melhorar o aprendizado.

Portanto, o erro é visto como um fator importante na aprendizagem, proporcionando oportunidades para que o aluno compreenda suas falhas e busque novas soluções. Isso envolve investigar, explorar e descobrir por si mesmo, alinhando-se com a teoria construcionista que defende a aprendizagem através da descoberta (RESNICK, 2020).

Ao introduzir a programação nas escolas, proporcionamos aos alunos uma oportunidade única de desenvolver o pensamento computacional. Termo este, que foi introduzido por Jeannette Wing no século XXI, e estava voltado para a educação. A autora o define como um conjunto essencial de competências e habilidades a serem desenvolvidas pelos alunos desde o Ensino Básico. Em sua abordagem, Wing esclarece que o pensamento computacional não visa fazer com que os seres humanos pensem como computadores, reconhecendo a natureza tediosa e enfadonha dos computadores em contraste com a esperteza e imaginação humanas. (2006, p.35).

Distingue-se do simples conhecimento em navegar na internet, enviar e-mails ou utilizar ferramentas computacionais como processadores de texto e planilhas de cálculo. O pensamento computacional, segundo Wing, refere-se à habilidade de utilizar o computador como uma extensão do poder cognitivo e operacional humano, buscando aumentar a

produtividade, inventividade e criatividade. Trata-se, essencialmente, de ativar funções cognitivas que podem ser realizadas pelo computador e, nesse contexto, envolve a capacidade de programá-lo para executar tais tarefas. Wing destaca que o pensamento computacional não é exclusivo dos cientistas da computação, sendo uma habilidade fundamental relevante para todos (2006, p.33). Ou seja, a criação de algoritmos, a resolução de problemas por meio da lógica de programação e a compreensão de como as instruções são executadas no ambiente digital são aspectos cruciais desse aprendizado.

Papert (1985) define o pensamento computacional como um conjunto de meios para resolver problemas, incorporando características específicas e disposições. Essa abordagem é essencial para o aprendizado da programação de computadores, que representa a linguagem pela qual o computador é capaz de compreender. Nesse contexto, Papert (1985) destaca que programar um computador proporciona à criança uma sensação de domínio sobre a máquina, estabelecendo um contato íntimo com ideias fundamentais da ciência e da matemática.

Conforme Raabe (2016), ensinar a programar por si só não é suficiente para desenvolver habilidades relacionadas ao pensamento computacional nos estudantes. É necessário uma mudança completa nos hábitos e métodos de ensino, fundamentada no construcionismo, permitindo que o estudante se torne o protagonista de sua aprendizagem através da construção de um objeto de seu interesse.

Segundo Papert, o ato de programar não apenas contagia a criança com um sentimento de controle sobre a máquina, mas também facilita a criação de modelos intelectuais significativos. Ele enfatiza que o aprendizado é transformado quando a criança programa, pois ela não apenas absorve conhecimento, mas realmente faz algo com esse conhecimento. O novo entendimento se torna uma fonte de empoderamento, sendo experimentado como tal desde o momento em que começa a se formar na mente da criança (1985, p. 37).

O pensamento computacional não se limita apenas à programação; ele serve como uma habilidade transversal que beneficia diversas áreas do conhecimento e da vida cotidiana. Ao aprender a pensar de maneira computacional, os alunos adquirem uma mentalidade analítica e sistemática que pode ser aplicada em diversas situações. Ainda neste contexto de programação de computadores e resolução de problemas, Jeannette Wing destaca que,

pensamento computacional envolve a resolução de problemas, projeção de sistemas, e compreensão do comportamento humano, através da extração de conceitos fundamentais da ciência da computação. O pensamento computacional inclui uma série de ferramentas mentais que refletem a vastidão do campo da ciência da computação. [...] Ao resolver um problema eficientemente, podemos questionar se uma solução aproximada é boa o suficiente e se falsos positivos ou falsos negativos

são permitidos. O pensamento computacional é reformular um problema aparentemente difícil em um problema que sabemos como resolver, talvez por redução, incorporação, transformação ou simulação. (2006, p. 33).

Em seus artigos posteriores, Wing (2010) define o pensamento computacional como os processos mentais envolvidos na formulação e resolução de problemas, garantindo que as soluções sejam representadas de maneira eficaz para execução por um agente de processamento de informações. Em uma publicação subsequente (WING, 2014), ela amplia essa definição ao descrever o pensamento computacional como os processos mentais relacionados à formulação de problemas e à expressão eficaz de suas soluções, de modo que tanto uma máquina quanto uma pessoa possam realizar. Além disso, Wing adiciona conceitos como "automação da abstração" e "o ato de pensar como um cientista da Computação".

De acordo com Raabe et al. (2020), Jeannette Wing introduziu o conceito de pensamento computacional, argumentando que essa abordagem vai além da resolução de problemas computacionais e pode ser aplicada em diversas áreas. Wing defende que o pensamento computacional é uma habilidade fundamental e que deveria ser integrada às atividades escolares. Esse tipo de pensamento desenvolve competências importantes, como a abstração, o reconhecimento de padrões, a reformulação de problemas e a capacidade de dividir um problema maior em partes menores.

A noção de que a programação de computadores contribui para uma melhoria no pensamento não é recente, sendo que Papert (1985) já destacava a importância dessa atividade para a construção do conhecimento e o desenvolvimento do pensamento. Em 1971, Papert também apontou que a tecnologia poderia ter um impacto significativo ao concretizar e esclarecer conceitos anteriormente sutis em áreas como psicologia, linguística, biologia, lógica e matemática. Essa viabilidade se torna concreta quando a criança tem a oportunidade de articular o trabalho de sua própria mente e, especialmente, a interação entre sua mente e a realidade durante o processo de aprendizagem e pensamento (Papert, 1971, p. 3).

Ao observarmos ao nosso redor, é evidente que o mundo em que vivemos está saturado de tecnologia. A cada momento, novos aplicativos e plataformas digitais são desenvolvidos com o propósito de simplificar as atividades diárias das pessoas e também de oferecer soluções para problemas sociais, baseadas em ações colaborativas e na velocidade proporcionada pelas máquinas. Na contemporaneidade, para que um indivíduo possa ser autônomo e capaz de ativamente transformar o ambiente em que está inserido, não basta apenas saber utilizar aplicativos e ferramentas digitais; é imperativo compreender e "falar" a linguagem que permeia e viabiliza o desenvolvimento tecnológico de maneira constante e

ágil, considerando o contexto do mundo em que vivemos. Essa nova linguagem é a programação de computadores, e nesse contexto, Mitchel Resnick (2020) enfatiza que aprender a programar no século XXI é tão importante quanto foi aprender a ler, escrever e falar no século XX, reforçando que

mesmo que a maioria das pessoas não se torne um jornalista ou escritor profissional, é importante que todos aprendam a escrever. Isso também se aplica à programação, por razões semelhantes. A maioria das pessoas não será um programador profissional ou cientista da computação, mas aprender a programar com fluência é uma habilidade valiosa para todos. Tornar-se fluente, seja na escrita ou na programação, ajuda a desenvolver seu pensamento, desenvolver sua voz e desenvolver sua identidade. (RESNICK, 2020, p. 44 - 45).

Considerando as informações apresentadas, o pensamento computacional emerge como uma habilidade essencial que todos os indivíduos deveriam cultivar. Sua abrangência transcende conceitos meramente computacionais, abordando aspectos como compreensão do comportamento humano, design de sistemas e pensamento crítico, conforme destacado por Wing (2010). O pensamento computacional incorpora uma série de características fundamentais, conforme delineado por Brackman (2017), que incluem a capacidade de identificar, analisar e implementar soluções eficientes e eficazes para problemas, reformular problemas complexos, escolher representações adequadas, interpretar código e dados, utilizar abstração e decomposição, avaliar a simplicidade e elegância de sistemas, pensar de forma recursiva, verificar padrões, prevenir, detectar e recuperar de situações adversas, modularizar antecipadamente, entre outras. Brackman (2017) destaca que o pensamento computacional promove uma aprendizagem interdisciplinar, sendo avaliado por meio de nove itens, tais como abstração e reconhecimento de padrões, processamento sistemático da informação, sistemas de símbolos e representações, noções de controle de fluxo em algoritmos, decomposição de problemas, pensamento iterativo, recursivo e paralelo, lógica condicional, eficiência e restrições de desempenho, e depuração e detecção de erros sistemáticos. Essa abordagem ampla e interdisciplinar do pensamento computacional revela sua relevância para o desenvolvimento cognitivo abrangente dos aprendizes, proporcionando-lhes habilidades fundamentais para lidar com uma variedade de problemas em diferentes contextos.

A contextualização do pensamento computacional em ambiente escolar ganha relevância à medida que vivemos em uma era cada vez mais digital e interconectada. A rápida evolução da tecnologia influencia significativamente a forma como enfrentamos desafios e oportunidades no século XXI. Nesse contexto, a integração do pensamento computacional nas

práticas educacionais busca preparar os alunos para uma sociedade em constante transformação, para tanto Robinson nos coloca que,

no século XXI, a humanidade enfrenta alguns de seus maiores desafios. Nossa melhor solução é cultivar nossos talentos de imaginação, criatividade e inovação. Esta deveria ser uma das maiores prioridades da educação no mundo todo. A educação é a solução para o futuro e os riscos dificilmente poderiam ser maiores. (ROBINSON, 2019, p. 54).

A contextualização do pensamento computacional no ambiente escolar também busca promover a equidade no acesso à tecnologia, preparando os alunos para serem cidadãos ativos em uma sociedade cada vez mais digitalizada. Isso pode envolver o uso de ferramentas educacionais, jogos e projetos que incentivam a aplicação prática do pensamento computacional em contextos do dia a dia. Neste sentido Resnick nos coloca que

as crianças do mundo todo estão passando cada vez mais tempo diante de telas: brincando com *videogames*, enviando mensagens aos amigos, assistindo vídeos e pesquisando informações. As tecnologias envolvidas nessas atividades, e aquelas encontradas na loja de brinquedos, são incrivelmente criativas. Contudo, na maior parte dessas atividades, as crianças apenas *interagem* com as tecnologias, e não *criam* com elas. Se queremos que as crianças cresçam como pensadoras criativas, precisamos proporcionar a elas diferentes maneiras de envolvimento com as telas, oferecendo mais oportunidades de criarem os próprios projetos e expressarem as próprias ideias. (RESNICK, 2020, p. 41).

Além disso, ao integrar essa abordagem, as escolas incentivam a criatividade, o raciocínio crítico e a colaboração, habilidades essenciais para a adaptação em um mundo dinâmico. Vale ressaltar que “todo mundo tem um talento criativo. O desafio é desenvolvê-lo” (ROBINSON, 2019, p. 19). A contextualização do pensamento computacional, portanto, não apenas reflete a demanda da sociedade contemporânea, mas também visa proporcionar uma educação mais abrangente e alinhada com as exigências do século XXI.

No contexto educacional, o pensamento computacional foi incorporado à Base Nacional Comum Curricular (BNCC), que orienta os currículos escolares em todo o país. A BNCC define as aprendizagens essenciais que devem ser promovidas pelas escolas públicas e privadas. Entre as 10 competências gerais da educação básica, está a capacidade de

compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva" (BRASIL, 2017).

O pensamento computacional é destacado na BNCC como uma habilidade específica na área de Matemática. O documento estipula que, no Ensino Médio, os alunos devem ser capazes de "utilizar os conceitos básicos de uma linguagem de programação na implementação de algoritmos escritos em linguagem corrente e/ou matemática" (BRASIL, 2017, p. 531). No entanto, seu impacto vai além da Matemática. Segundo Mestre et al. (2017), o pensamento computacional também ajuda na resolução de problemas em diversos contextos da sociedade, permitindo que as pessoas apliquem conceitos de computação em suas atividades diárias.

Trabalhar com o pensamento computacional na escola apresenta diversos benefícios e razões que refletem a necessidade de preparar os alunos para um mundo cada vez mais digital e orientado pela tecnologia. Aqui estão algumas razões significativas: habilidades cognitivas, preparação para o futuro, estímulo à criatividade e inovação, fomento à lógica e raciocínio crítico, equidade no acesso à tecnologia, aprendizagem ativa e colaborativa e adaptação à mudança. Robinson define “a *imaginação* é o processo de usar a mente para pensar em coisas que não estão presente aos nossos sentidos; a *criatividade*, que é o processo de desenvolver ideias originais que tenham valor; e a *inovação*, que é o processo de colocar as novas ideias em prática” (2019, p. 18). Papert exemplifica seu modo de pensar sobre a criatividade da seguinte forma,

[...] ao invés de sufocar a criatividade da criança, a solução é criar um ambiente intelectual menos dominado pelo critério de falso e verdadeiro, como acontece na escola. [...] Elas aprendem matemática e ciência em um ambiente onde falso e verdadeiro, certo ou errado não os critérios decisivos [...] (1985, p. 163).

Além disso, a preparação para o futuro profissional é uma consideração significativa. Em um mundo cada vez mais tecnológico, o pensamento computacional não apenas prepara os alunos para carreiras diretamente relacionadas à tecnologia, mas também os desenvolve habilidades valiosas em praticamente todos os setores. A capacidade de entender e aplicar princípios computacionais torna-se uma vantagem competitiva no mercado de trabalho.

Outro ponto relevante é o estímulo à criatividade e inovação. Ao encorajar os alunos a encontrar soluções únicas para problemas, o pensamento computacional cultiva uma mentalidade inovadora. A capacidade de pensar de maneira criativa e adaptativa é essencial em um ambiente que valoriza a originalidade e a capacidade de encontrar abordagens únicas para desafios.

O pensamento computacional também contribui para o desenvolvimento de habilidades de resolução de problemas. Ao proporcionar uma abordagem estruturada e sistemática para enfrentar desafios, os alunos aprendem a decompor problemas complexos em partes gerenciáveis e a abordá-los de maneira organizada. Essa competência é valiosa não apenas no contexto tecnológico, mas em situações cotidianas diversas.

Além disso, o pensamento computacional incentiva a aprendizagem ativa e colaborativa. Ao envolver os alunos em projetos práticos, resolver problemas em equipe e compartilhar ideias, promove-se um ambiente educacional dinâmico e participativo, visto que “o pensamento é integrado ao fazer no contexto de interagir, brincar, criar coisas, e a maioria dos pensamentos é feita em conexão com outras pessoas, compartilhando ideias, obtemos reações, completamos as ideias delas” (RESNICK, 2020, p. 86). Essa abordagem não apenas fortalece o aprendizado, mas também prepara os alunos para colaborar efetivamente em ambientes profissionais e sociais.

Outra razão essencial é a capacidade de adaptação a mudanças tecnológicas. O pensamento computacional proporciona uma compreensão sólida dos princípios computacionais, preparando os alunos para se adaptarem rapidamente a mudanças no cenário tecnológico. Essa mentalidade flexível e aberta à inovação é importante em um mundo em constante evolução.

Por fim, o pensamento computacional contribui para a alfabetização digital. Em um contexto digital, não se trata apenas de usar tecnologia, mas de compreender os fundamentos por trás das ferramentas digitais. Isso capacita os alunos a participar plenamente na sociedade digital, sendo cidadãos digitalmente competentes e informados, uma vez que estamos formando crianças com habilidades de construir seus conhecimentos de forma ativa. (RESNICK, 2020).

A importância da criatividade e do raciocínio lógico no pensamento computacional é fundamental para promover uma abordagem integral e eficaz no desenvolvimento de habilidades computacionais. A criatividade desempenha um papel importante ao inspirar os alunos a abordar problemas de maneira original, fomentando a inovação na criação de algoritmos e na resolução de desafios computacionais. Essa habilidade não apenas torna a programação uma atividade dinâmica e expressiva, mas também instiga soluções únicas e eficientes.

Por outro lado, o raciocínio lógico se apresenta como a espinha dorsal do pensamento computacional, proporcionando a estrutura necessária para analisar situações, decompor problemas e construir algoritmos claros e precisos, como abordamos no capítulo anterior. O

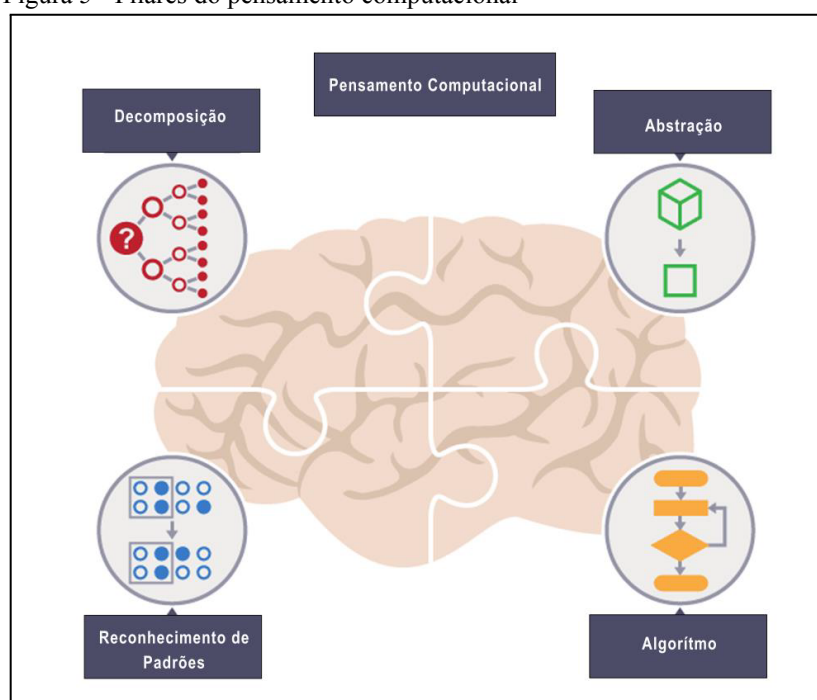
desenvolvimento do raciocínio lógico é essencial não apenas na programação, mas também em diversas áreas da vida, promovendo uma abordagem estruturada para a resolução de problemas.

A integração harmoniosa da criatividade com o raciocínio lógico resulta em soluções inovadoras e eficazes. Ao desafiar os alunos a pensar fora da caixa e, ao mesmo tempo, aplicar uma lógica rigorosa, essa combinação é particularmente valiosa na resolução de problemas complexos. Preparando os alunos para enfrentar desafios futuros, essa abordagem equilibrada no pensamento computacional não apenas os capacita na programação, mas também os instiga a adaptar-se a novas tecnologias e a abordar questões emergentes de maneira criativa e estruturada. Em última análise, a interação entre a criatividade e o raciocínio lógico proporciona uma base sólida para uma compreensão abrangente da computação e suas aplicações.

4.1.1 Pilares do pensamento computacional

O pensamento computacional está estruturado em quatro pilares: decomposição, reconhecimento de padrões, abstração e algoritmo - transcendem a mera resolução de problemas técnicos, transformando-se em competências fundamentais para a compreensão e navegação no mundo moderno (BBC Learning, 2015), conforme representado na figura 5.

Figura 5 - Pilares do pensamento computacional



Fonte: "Introduction to Computational Thinking". BBC Bitesize, 2023.

De acordo com Humphreys (2015), o pensamento computacional representa um processo cognitivo que engloba raciocínio lógico, permitindo a resolução de problemas e uma compreensão aprimorada de procedimentos e sistemas. O autor destaca que essas habilidades de pensamento e resolução de problemas são cruciais e devem estar incorporadas em todos os currículos e na vida em geral. O pensamento computacional, segundo Humphreys, abrange a capacidade de pensar em termos de algoritmos, a habilidade de realizar a decomposição de problemas complexos em partes menores, a capacidade de pensar em generalizações identificando e utilizando padrões, bem como a capacidade de pensar em abstrações, fazendo escolhas de boas representações e a capacidade de pensar em termos de avaliação. Essa abordagem abrangente do pensamento computacional proposta por Humphreys destaca sua importância como uma ferramenta fundamental para o desenvolvimento do pensamento lógico e para a solução eficaz de problemas em diversas esferas da vida.

A decomposição não apenas facilita a resolução de problemas, mas também a capacidade de gerenciar tarefas complexas, dividindo-as em etapas realizáveis. Isso não só promove uma abordagem mais eficiente na resolução de problemas, mas também aprimora a capacidade de gerenciar projetos e tarefas do dia a dia. Exemplo deste pilar é imaginar a tarefa complexa de planejar uma festa. Ao decompor esse desafio, você identifica etapas como escolher o tema, fazer a lista de convidados, planejar o menu, organizar a decoração e coordenar as atividades. Cada uma dessas etapas pode ser gerenciada separadamente, facilitando a execução eficiente do planejamento como um todo. (WING, 2006) e (BBC Learning, 2023).

O reconhecimento de padrões, por sua vez, não se limita à análise de dados, mas se estende à interpretação de situações e ao entendimento de dinâmicas complexas. Ao identificar padrões em contextos diversos, os indivíduos desenvolvem uma visão analítica que vai além da tecnologia, permeando a compreensão de fenômenos sociais e naturais. Exemplo deste pilar é a análise de dados, identificar padrões pode ser determinante. Ao examinar o histórico de vendas de uma empresa, você pode reconhecer padrões sazonais, ajudando na previsão de demanda e ajuste de estratégias de marketing para períodos específicos do ano. (WING, 2006) e (BBC Learning, 2023).

A abstração, terceiro pilar, não apenas simplifica problemas complexos, mas fomenta a capacidade de generalização. Ao criar modelos simplificados, os indivíduos aprendem a aplicar conceitos aprendidos em um contexto específico a situações mais amplas, promovendo

uma compreensão profunda e flexível. Exemplo deste pilar é na aprendizagem de programação, abstrair um problema pode envolver a criação de uma função que execute uma tarefa específica. Por exemplo, ao desenvolver um programa de gerenciamento de contatos, abstrair a funcionalidade de adicionar um novo contato simplifica o código e torna-o mais compreensível. (WING, 2006) e (BBC Learning, 2023).

No que diz respeito aos algoritmos, o quarto pilar, sua aplicação vai além da programação. Pensa-se em termos de algoritmos não apenas para resolver problemas técnicos, mas para planejar estratégias, tomar decisões e otimizar processos em diversas áreas da vida. Exemplo deste pilar é considerar a tarefa de preparar uma receita de bolo. O conjunto de instruções sequenciais, desde a preparação dos ingredientes até a etapa de assar, representa um algoritmo. Seguir essas instruções garante a replicação bem-sucedida da receita. (WING, 2006) e (BBC Learning, 2023)

Esses pilares, quando integrados, não apenas capacitam a resolução de desafios computacionais, mas moldam uma mentalidade que se traduz em habilidades transversais. A capacidade de decompor, reconhecer padrões, abstrair e pensar algoritmicamente torna-se um arsenal poderoso para enfrentar a complexidade da sociedade moderna, contribuindo para a formação de indivíduos adaptáveis, analíticos e inovadores. Esses não são apenas princípios para entender a tecnologia; são ferramentas para compreender e moldar ativamente o mundo ao nosso redor.

4.2 Considerações parciais

O pensamento computacional apresenta-se como uma realidade a ser considerada pelas escolas, acerca deste tema buscou-se ao longo do capítulo investigar os desdobramentos do tema no processo de complexificação da capacidade de argumentação, através do questionamento: Quais as conexões do pensamento computacional na formação de um ser argumentativo que resolve problemas quando programa computadores?

O pensamento computacional desempenha um papel importante na formação de indivíduos capazes de resolver problemas e argumentar de maneira eficaz ao programar computadores. Essa ligação entre habilidades cognitivas e técnicas abre diversas conexões que contribuem para um desenvolvimento integral.

Em primeiro lugar, destaca-se o raciocínio lógico exigido na programação de computadores e na construção de argumentos sólidos. A capacidade de criar algoritmos

eficientes reflete-se na organização lógica de ideias ao formular argumentos em diferentes contextos.

A resolução de problemas é outra área onde o pensamento computacional e a programação se encontram. A habilidade de identificar e solucionar possíveis erros na programação traduz-se em uma abordagem sistemática e analítica para resolver problemas em diversas situações do dia a dia.

A aplicação do pensamento computacional e da argumentação dedutiva revela uma conexão profunda entre habilidades aparentemente distintas. A abstração, uma competência essencial na programação para simplificar conceitos complexos em blocos de construção mais simples, demonstra ser transferível para a construção de argumentos claros e objetivos. Essa capacidade de destacar ideias chave e simplificar informações não apenas contribui para a eficácia da comunicação, mas também permite uma compreensão mais clara e acessível dos argumentos apresentados. Além disso, a decomposição, prática comum na programação ao dividir problemas complexos em partes menores, revela-se valiosa ao enfrentar argumentos entrelaçados. Essa habilidade permite uma análise mais detalhada e abrangente, pois os indivíduos podem examinar cada componente do argumento separadamente, identificando suas relações e avaliando sua validade individualmente. Por fim, o reconhecimento de padrões, fundamental na identificação de padrões de dados na programação, estende-se à capacidade de reconhecer padrões em informações e argumentos. Esse discernimento contribui para uma visão mais integral, permitindo uma compreensão mais profunda das variantes envolvidas e facilitando a identificação de falácias lógicas. Em suma, a aplicação desses princípios do pensamento computacional na argumentação dedutiva não apenas fortalece a resolução de problemas, mas também promove uma análise crítica mais precisa e uma comunicação mais eficaz em uma variedade de contextos.

Ao integrar o pensamento computacional na formação de indivíduos argumentativos que programam computadores, cria-se uma sinergia de habilidades que não apenas promove a proficiência técnica, mas também aprimora competências cognitivas fundamentais para enfrentar os desafios complexos do mundo contemporâneo. Esse conjunto de habilidades é básico em um cenário onde a tecnologia desempenha um papel cada vez mais central, conectando o universo da programação com a capacidade de pensar de maneira crítica e construtiva.

5 METODOLOGIA

A criação de um projeto de pesquisa abrange elementos teóricos e metodológicos científicos indispensáveis tanto para a organização do conhecimento quanto para a prática em campo. Na literatura, há diversos componentes que ajudam na consolidação do universo da pesquisa. Contudo, é necessário que todos os procedimentos sejam descritos com precisão, deixando claro o processo de interpretação que o pesquisador utilizará (DESLANDES, 2018, p. 45).

Segundo Minayo (2002), esse processo de estruturação se apoia em três componentes principais: a teoria, o método e a criatividade. Esses elementos são geradores de conhecimento e ajudam a desvendar a dinâmica de entender a realidade. Portanto, é fundamental delinear e explicar com precisão a estrutura fundamentada nos objetivos, nos métodos, na natureza das fontes utilizadas para abordar e analisar o objeto de estudo, e nas técnicas aplicadas. Complementando, Minayo (2002) afirma que, para que o pensamento possa orientar a prática em uma abordagem realista, é necessário considerar simultaneamente o método, as técnicas e a experiência. Dessa forma, combinam-se a teoria do conhecimento, sua aplicação prática e a criatividade, ou seja, o pensamento crítico e a sensibilidade do pesquisador.

Segundo Deslandes (2018), ao elaborar um trabalho científico, enfrentam-se três aspectos: técnico, ideológico e científico. O aspecto técnico envolve a escolha dos instrumentos e a definição do objeto e da abordagem da pesquisa, seguindo regras científicas. O aspecto ideológico refere-se às escolhas do pesquisador sobre temas, bases teóricas e metodologias, com base em suas convicções pessoais e contextos históricos e sociais. O aspecto científico busca integrar os dois anteriores para ir além do senso comum. Lakatos e Marconi (2003) afirmam que a metodologia científica conecta o pesquisador com práticas e ideias, consolidando a metodologia científica, as técnicas de pesquisa e a metodologia do trabalho.

Neste contexto, descreve-se a trajetória da pesquisa, que combina uma investigação bibliográfica com um estudo de campo. Utilizam-se técnicas qualitativas, como revisão de literatura, observação, questionário e teste. A análise e o tratamento dos dados são realizados através do método indutivo, utilizando uma abordagem descritiva com elementos exploratórios. Isso fornece a base para o relatório, que inclui a organização dos dados em categorias, bem como as considerações e resultados obtidos.

5.1 Categorias de análise

Para Gomes (2009), as categorias de análise devem ser homogêneas, ou seja, baseadas em princípios consistentes. Em geral, uma categoria abrange elementos ou aspectos com características comuns, permitindo a organização de ideias ou expressões em torno de um conceito abrangente. Marconi e Lakatos (2018) definem categorias como grupos ou classes dentro de uma série classificada. Ainda na perspectiva dos autores, as categorias devem derivar de um único princípio de classificação e ser completas, permitindo que cada resposta seja alocada em uma dessas categorias.

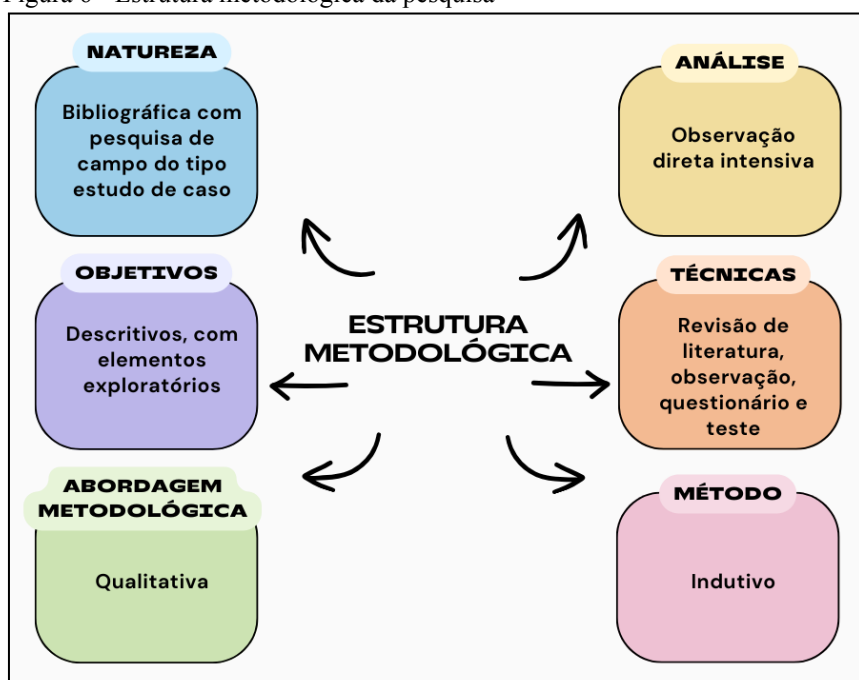
Para o estudo que articula o pensamento computacional e argumento dedutivo, as categorias de análise foram fundamentadas na abstração e generalização. A **abstração** envolve a identificação e o foco nas características essenciais dos conceitos, como os princípios do pensamento computacional e os elementos chave do argumento dedutivo, sem se fixar em detalhes específicos, o que facilita uma compreensão mais clara e estruturada desses elementos. Já a **generalização** consiste em aplicar os princípios abstraídos a diversas situações problemas e exemplos, abrangendo diferentes situações. Isso enriquece a análise ao demonstrar como os conceitos de pensamento computacional e argumento dedutivo podem se manifestar e interagir em variadas circunstâncias. A combinação de abstração e generalização permite organizar e interpretar os dados de forma a revelar padrões e relações significativas entre os temas investigados.

5.2 Estrutura metodológica

Desenvolver a metodologia da pesquisa social é uma fase desafiadora que requer um alto nível de envolvimento do pesquisador. Para assegurar uma base ideológica robusta e coerente com os objetivos do estudo, escolheu-se fundamentar a realidade empírica qualitativa na experiência de campo, seguindo as etapas do estudo de caso conforme Gil (2002).

A figura a seguir resume a proposta desta pesquisa, considerando que "a definição da metodologia requer dedicação e cuidado do pesquisador [...]. Indica as conexões e a leitura operacional que o pesquisador fez do quadro teórico e de seus objetivos de estudo" (DESLANDES, 2018, p. 42).

Figura 6 - Estrutura metodológica da pesquisa



Fonte: Construção da autora com base na teoria, 2024.

Para Minayo (2002), a pesquisa qualitativa se concentra no universo das relações humanas, significados, representações, crenças, valores e atitudes, que não podem ser traduzidos em números e indicadores. Gil (2019) destaca que a principal diferença entre pesquisas quantitativas e qualitativas é que a primeira apresenta resultados em termos numéricos, enquanto a segunda utiliza descrições verbais. A abordagem qualitativa é justificada porque, conforme Bodgan e Biklen (1994), "o comportamento humano é significativamente influenciado pelo contexto em que ocorre, sendo essencial deslocar-se ao local de estudo" (p. 48).

Para tanto, a pesquisadora se dedicará a analisar os dados em sua totalidade, mantendo constante atenção ao campo de pesquisa. Qualquer situação pode oferecer pistas valiosas que ajudem a compreender melhor o objeto de estudo (Bodgan e Biklen, 1994). Visto que, segundo Gil (2019), a pesquisa descritiva se caracteriza pelo uso de técnicas de coleta de dados em campo, analisando as características da população, fenômeno ou relações entre variáveis. Ela não apenas identifica a existência de relações entre variáveis, mas também busca determinar a natureza dessas relações, aproximando-se da pesquisa explicativa. Por outro lado, algumas pesquisas, embora definidas como descritivas, acabam oferecendo uma nova perspectiva sobre o problema, assemelhando-se às pesquisas exploratórias.

Baseia-se no método indutivo, que vai do particular ao geral. Gil (2019) explica que a generalização não deve ser buscada previamente, mas deve surgir a partir da observação de

casos concretos que confirmem a realidade estudada, ocorrendo após a coleta de dados. Lakatos e Marconi (2003) complementam que é necessário observar atentamente os fatos ou fenômenos, classificá-los e agrupá-los com base em suas semelhanças, para então chegar à generalização e classificar cada relação observada.

A observação direta intensiva exige uma análise detalhada dos fatos ou fenômenos em estudo, indo além de simplesmente ver e ouvir. Nesta tese, a coleta de dados é realizada por meio da observação participante, onde a pesquisadora se integra ao grupo durante as atividades, utilizando questionamentos e incentivando os participantes a refletirem para obter as suas próprias informações (Lakatos e Marconi, 1992). Esta abordagem é uma forma de observação participante, na qual o pesquisador se une ao grupo de maneira estruturada para coletar dados específicos, em vez de se tornar um membro natural e espontâneo do grupo. Marconi e Lakatos (2018) explicam que, nesse tipo de observação, "o observador integra-se ao grupo com a finalidade de obter informações".

Para estruturar a metodologia, foram escolhidas as concepções que melhor conectam a teoria com a realidade empírica, considerando métodos, técnicas, e a experiência da pesquisadora. A decisão sobre cada escolha foi feita com cuidado para garantir que teoria e metodologia estejam alinhadas e integrem-se de forma harmoniosa.

5.3 Metodologia da pesquisa empírica

De acordo com Minayo (2010), em um ciclo de pesquisa, após a fase exploratória e a elaboração do projeto de investigação, ocorre o trabalho de campo. Nesta etapa, realiza-se o recorte empírico da construção teórica por meio de entrevistas, observações, entre outros métodos, com o objetivo de confirmar ou refutar as hipóteses e a construção teórica.

A abordagem metodológica proposta em campo incorpora elementos qualitativos, integrando métodos de pesquisa de observação e estudo de caso, com o objetivo analisar os desdobramentos do pensamento computacional no processo de complexificação da capacidade de argumentação. Segundo Bogdan e Biklen (1994), o principal objetivo dos pesquisadores qualitativo, é

[...] melhor compreender o comportamento e experiência humana. Tentam compreender o processo mediante o qual as pessoas constroem significados e descrever em que consistem estes mesmos significados. Recorrem à observação empírica por considerarem que é em função de instâncias concretas do comportamento humano que se pode refletir com maior clareza e profundidade sobre a condição humana (p. 70).

A pesquisa de observação foi conduzida em tempo real na sala de aula, permitindo uma avaliação direta do desempenho dos alunos no desenvolvimento de argumentos. Esse método qualitativo fornecerá dados detalhados sobre os comportamentos, interações e dinâmicas presentes durante o processo de aprendizagem. As técnicas utilizadas podem ser consideradas “dados de campo, ou seja, obtidos no local em que os fenômenos ocorrem espontaneamente, mediante procedimentos como observação, aplicação de questionário e entrevistas” (GIL, 2019, n.p). Segundo Gil (2009) e Lakatos e Marconi (1992), essas técnicas constituem a parte prática da pesquisa de campo, englobando métodos de coleta de dados que foram posteriormente analisados.

De acordo com Lakatos e Marconi (1992), o estudo de caso é uma abordagem de pesquisa que se dedica a investigar minuciosamente um caso específico, seja ele um indivíduo, grupo, evento, organização ou fenômeno, que possua uma característica comum. Durante a fase exploratória e na seleção da amostra, é fundamental especificar detalhes como o sexo, a faixa etária, a organização à qual pertencem, entre outros aspectos relevantes para a pesquisa. É essencial detalhar cada etapa do trabalho de “campo (escolha do espaço de pesquisa, critérios e estratégias para escolha do grupo/sujeitos de pesquisa, a definição de métodos, técnicas e instrumentos para a construção de dados e os mecanismos para entrada em campo)” (DESLANDES, 2018, p. 43), além dos procedimentos para análise.

Paralelamente, foi elaborado um questionário estruturado³ (ANEXO A), com perguntas específicas que visam validar a capacidade argumentativa dos alunos. Este instrumento possibilitará a coleta de dados qualitativa e descritiva, complementando a análise qualitativa obtida pela pesquisa de observação e estudo de caso. Adicionalmente, um teste de pensamento computacional⁴ (ANEXO B) foi aplicado para avaliar as competências dos participantes neste domínio específico. Esse teste incluirá questões práticas que exigirão a aplicação de conceitos relacionados a algoritmos, decomposição de problemas e outras habilidades inerentes ao pensamento computacional, contribuindo para a análise quantitativa.

O teste de pensamento computacional foi empregado na pesquisa por sua abordagem quantitativa e eficácia, bem como por ter passado por um processo rigoroso de validação. Este processo confirmou a validade do conteúdo (GONZÁLEZ, 2015), dos critérios (GONZÁLEZ, 2015) e a validade convergente (GONZÁLES et al., 2016).

³ O questionário dividido em cada uma das etapas, pode ser acessado em cada fase, através dos seguintes links: etapa 1: <https://abre.ai/qcep>; etapa 2: <https://abre.ai/qcep2>; etapa 3: <https://abre.ai/qcep3>; etapa 4: <https://abre.ai/qcep4>; etapa 5: <https://abre.ai/qcep5>.

⁴ O teste de pensamento computacional pode ser acessado através do seguinte link: <https://abre.ai/testepcep>.

Originalmente o teste era composto por 28 questões, e buscava identificar a habilidade de formação e solução de problemas, utilizando os princípios básicos da computação e incorporando sintaxes lógicas presentes em linguagens de programação. Ele também abrange os conceitos dos quatro pilares do pensamento computacional: abstração, decomposição, reconhecimento de padrões e algoritmos. A pesquisadora adaptou o teste para 16 questões, mantendo sua essência, mas destacando o pilar do pensamento computacional: abstração. Esta escolha é justificada pela categoria de análise do estudo. Uma análise detalhada da relação entre esses pilares e as questões é fornecida no quadro 22.

Quadro 22 - Pilares do pensamento computacional abordados no teste

QUESTÃO	ABSTRAÇÃO	DECOMPOSIÇÃO	RECONHECIMENTO DE PADRÕES	ALGORITMO
1	x			x
2	x			x
3	x			x
4	x	x		x
5	x	x	x	x
6	x	x	x	x
7	x	x		x
8	x		x	x
9	x	x	x	x
10	x	x		x
11	x	x	x	x
12	x	x	x	x
13	x	x	x	x
14	x	x	x	x
15	x	x	x	x
16	x	x	x	x

Fonte: Autora, 2024.

No contexto da pesquisa, o teste de pensamento computacional foi administrado tanto no início quanto no final das atividades, permitindo a coleta de dados comparativos sobre o progresso dos participantes ao longo do período de intervenção. Esse procedimento

possibilitou uma análise mais precisa das mudanças na habilidade de formação, solução de problemas e raciocínio lógico dos alunos. Além disso, durante as aulas de programação de computadores utilizando-se do *software* Scratch, foi aplicado o questionário elaborado para validar a capacidade argumentativa dos estudantes. Esta aplicação ocorreu em conjunto com a observação direta e o estudo de caso, ou seja, os alunos responderam a quatro questões voltadas as quatro formas válidas de argumento dedutivo por aula, totalizando 20 questões. Esta divisão se justifica para que os indivíduos tenham contato com as quatro formas válidas de argumento dedutivo: *Modus ponens*, *modus tollens*, silogismo hipotético e silogismo disjuntivo em todos os encontros. Ao final, permitirá uma avaliação mais abrangente das habilidades cognitivas e comunicativas dos participantes no contexto específico do aprendizado da programação.

O público alvo é composto por dois alunos do sexo masculino, frequentando o 7º ano do Ensino Fundamental na Escola Estadual de Ensino Fundamental Mathias Lorenzon, localizada no município de Getúlio Vargas, RS. Esses alunos foram escolhidos aleatoriamente e de forma voluntária dentro do projeto que ocorre na escola, considerando o conhecimento prévio de programação de computadores. A seleção considerou a participação dos alunos no projeto Conect +, que ocorre na escola e aborda a programação de computadores através da resolução de problemas, realizado semanalmente no turno inverso às aulas.

Um programa de parceria público-privada, o Conect +, é estabelecido entre a 15ª Coordenadoria Regional de Educação do Rio Grande do Sul (CRE-RS) e a ARAPL Intelligent Manufacturing, com o objetivo de introduzir aulas de pensamento lógico computacional do 6º ao 9º ano do ensino fundamental. O contrato definiu compromissos mútuos: o Poder Público ofereceu um ambiente regulatório favorável (espaço escolar) e recursos técnicos (computadores e internet), enquanto o Setor Privado forneceria recursos humanos, técnicos e treinamento para os professores. O projeto foi avaliado por meio de testes padronizados, visando melhorar a capacidade dos alunos em resolver problemas de forma lógica e eficiente. Um grupo de análise de dados é formado para direcionar ações futuras, e atividades extras são realizadas para integrar as turmas com foco em tecnologias para cidades sustentáveis. O contrato teve uma duração inicial de 2 meses, com disposições sobre proteção de dados, propriedade intelectual e resolução de conflitos, mas com os avanços apresentados pelos estudantes, o contrato foi estendido e após um ano de trabalho conjunto, o programa Conect + continua vigorando e segue com o propósito de ensinar para o ano de 2024.

O projeto Conect + visa ensinar pensamento computacional e raciocínio lógico, promovendo competências que vão desde a habilidade de abordar problemas complexos de maneira estruturada e analítica até trabalhar com algoritmos e blocos de construção. Além disso, o projeto aprimora habilidades de comunicação, criatividade e pensamento crítico. As atividades ocorrem no ambiente escolar, no turno inverso ao horário regular de aulas, de forma voluntária, através de oficinas semanais de 2 horas, ministradas por um monitor. Durante as aulas, as atividades são organizadas de modo a criar situações problema que não apenas incentivam a reflexão sobre soluções eficazes, mas também proporcionam oportunidades para aprender e exercitar competências na programação de computadores.

A coleta de dados desta pesquisa foi realizada por meio de várias ferramentas para obter uma visão abrangente das capacidades dos alunos. Foi elaborado um questionário estruturado com perguntas específicas que visam validar a capacidade argumentativa dos alunos. Este instrumento permitiu a coleta de dados qualitativos, enriquecendo a análise qualitativa obtida pela pesquisa de observação e estudo de caso. Esta aplicação ocorreu em conjunto com a observação direta e o estudo de caso, permitindo uma avaliação mais abrangente das habilidades cognitivas e comunicativas dos participantes no contexto específico do aprendizado da programação. No contexto da pesquisa, o teste de pensamento computacional foi administrado tanto no início quanto no final das atividades, permitindo a coleta de dados comparativos sobre o progresso dos participantes ao longo do período de intervenção. Esse procedimento possibilitou uma análise mais precisa das mudanças na habilidade de formação, solução de problemas e raciocínio lógico dos alunos.

A intervenção educativa consistiu em seis aulas de duas horas cada, focadas na utilização do *software* Scratch para a resolução de problemas. Durante essas aulas, os alunos foram orientados a construir os passos necessários para a tomada de decisões, promovendo o desenvolvimento de habilidades tanto em programação quanto em pensamento crítico e argumentativo. Esta estrutura permitiu uma abordagem abrangente e progressiva, abordando diferentes aspectos do raciocínio lógico ao longo do programa, enquanto mantém o foco na execução bem sucedida das atividades planejadas.

Os dados qualitativos foram analisados utilizando técnicas de análise de conteúdo, permitindo identificar padrões e categorias emergentes das respostas dos alunos. Os dados quantitativos, oriundos dos testes de pensamento computacional, foram analisados estatisticamente para identificar possíveis correlações entre as variáveis estudadas. Para tanto, utilizaremos uma categoria de análise: abstração e generalização de conclusões, que tem como objetivo investigar como a capacidade de abstração (pilar do pensamento computacional) no

pensamento computacional pode facilitar a generalização de conclusões a partir de premissas específicas em um argumento dedutivo.

Em termos de considerações éticas, os responsáveis pelos alunos assinaram um termo de consentimento livre e esclarecido, garantindo a participação voluntária e consciente dos participantes. Os dados pessoais dos alunos foram mantidos em sigilo, utilizando códigos de identificação para garantir a privacidade dos participantes.

Ao final, a pesquisa visa responder ao questionamento central: Qual o papel do pensamento computacional na capacidade de produzir argumentos válidos? Com o objetivo de estabelecer relações entre o pensamento computacional e o processo de desenvolvimento da argumentação dedutiva. Dessa forma, a metodologia proposta busca não apenas explorar as interações entre esses conceitos, mas também fornecer uma compreensão mais profunda do impacto do pensamento computacional na formação de habilidades argumentativas em alunos do ensino fundamental II.

5.4 Scratch

O Scratch⁵ é um ambiente de programação visual criado pelo *Media Lab do Massachusetts Institute of Technology* (MIT), de Boston, Estados Unidos da América, inspirado nas linguagens Logo e *Squeak*, mas com a proposta de ser mais simples, acessível e intuitivo, além de permitir que os usuários criem, compartilhem e remixem projetos livremente, promovendo um ambiente colaborativo e educativo (RESNICK, 2020). A ferramenta foi projetada para que crianças possam criar programas sem precisar escrever códigos complexos. Conforme destaca Resnick (2020, p. 15), “criar projetos é a atividade básica da comunidade Scratch”. Em vez de codificar, elas utilizam blocos de comandos que se encaixam entre si para formar um conjunto de instruções, conforme a figura 7. Esses blocos facilitam a elaboração das instruções para o computador, já que a sintaxe é determinada pelo encaixe dos blocos, o que ajuda a reduzir erros comuns encontrados em linguagens de programação que requerem codificação manual, baseada em texto.

⁵ Scratch é uma plataforma de programação gratuita e de código aberto, desenvolvida e mantida pelo MIT Media Lab. Ele pode ser acessado e utilizado por qualquer pessoa através do site oficial (<https://scratch.mit.edu/>), e não há custos associados ao seu uso.

Figura 7 - Blocos de comandos utilizados para a programação no Scratch



Fonte: Autora, 2024.

A criação se baseia em quatro pilares conceituais: projetos, paixão, cooperação e aprender brincando⁶. Essa filosofia foi desenvolvida a partir da experiência de Mitchel Resnick (1998) no *Lifelong Kindergarten Group* do MIT, e inspirada no funcionamento do jardim de infância (RESNICK, 2020) e também influenciada pelas ideias de Seymour Papert (1993) sobre a relação entre educação e tecnologia.

Para Resnick (2020), o Scratch é fundamentado da seguinte maneira: **Projetos** (*Projects*) envolvem o planejamento cuidadoso da estrutura visual e da programação com blocos, considerando a experiência do usuário para otimizar a interface e o funcionamento do projeto. **Paixão** (*Passion*) incentiva a criação baseada nos interesses pessoais do usuário, garantindo que o projeto seja significativo para ele, o que promove maior engajamento e criatividade. **Cooperação** (*Peers*) estimula a troca de ideias e sugestões, com a plataforma permitindo que as criações sejam compartilhadas e remixadas por outros usuários. Por fim, **Pensar brincando** (*Play*) valoriza o processo de criação e aprendizado através do jogo e da diversão, mostrando que brincar pode ser uma forma séria e produtiva de trabalhar, uma abordagem cada vez mais adotada no mundo corporativo e em empresas de tecnologia.

Segundo Resnick (2020, p. 16), o Scratch foi criado para apoiar explorações lúdicas como um caminho para a criatividade, encorajando os jovens a assumir riscos e experimentar coisas novas, visto que o Scratch é uma plataforma de programação visual que permite criar aplicativos multimídia de maneira intuitiva. Seu objetivo é tornar a introdução aos conceitos

⁶ Originalmente em inglês: *projects, passion, peers e play*.

de matemática e computação mais acessível, promovendo também o pensamento criativo, o raciocínio lógico e a colaboração entre os usuários. De acordo com EDUScratch, “O Scratch foi projetado e desenvolvido para enfrentar o desafio do crescente descompasso entre os avanços tecnológicos globais e a fluência tecnológica das pessoas.”

A ferramenta Scratch utilizada neste trabalho visa apoiar o ensino de programação ao introduzir conceitos básicos de linguagem de programação para os alunos. Com seu layout simples, que utiliza apenas uma janela, a ferramenta torna mais fácil a manipulação e a execução de mídias e comandos (SCRATCH, 2017). Além disso, o Scratch busca proporcionar um ambiente onde “as crianças constroem as próprias casas e castelos, em vez de apenas brincar com casas e castelos prontos. Com o Scratch, elas programam as próprias histórias e jogos, em vez de simplesmente interagir com histórias e jogos já criados” (RESNICK, 2020, p. 43).

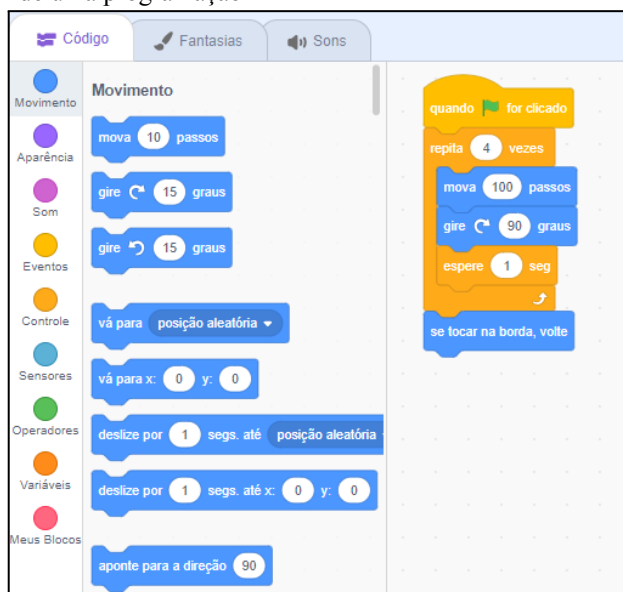
O Scratch oferece diversas potencialidades, como liberdade de criação, comunicação, criatividade e compartilhamento, funcionando como um parquinho virtual, uma vez que

[...] o Scratch é associado à programação, mas imaginação e compartilhamento também são importantes para a experiência Scratch. Assim como as crianças em um parquinho estão constantemente criando jogos para brincarem umas com as outras, as crianças do site do Scratch estão constantemente imaginando novos tipos de projetos e compartilhando suas criações. Outros sites de programação são projetados como cercadinhos, oferecendo um conjunto restrito de atividades para ajudar as crianças a aprenderem conceitos específicos de programação; para nós, a abordagem no estilo parquinho do Scratch é tão importante quanto as ideias de computação que fazem parte dos blocos de computação (RESNICK, 2020, p. 123).

Além disso, ele facilita a aprendizagem de conceitos escolares, manipulação de mídias, troca de projetos via internet, permitindo que os usuários compartilhem, reutilizem e adaptem projetos já existentes, e a integração de objetos do mundo físico (SCRATCH, 2017), onde “as crianças estão sempre compartilhando ideias e contribuindo para o trabalho umas das outras. Estão aprendendo como a comunidade científica funciona” (RESNICK, 2020, p. 97).

A programação no Scratch ocorre de maneira ordenada, através de blocos que são arrastados para o palco principal, respeitando uma ordem lógica e formando pilhas de códigos organizados. Esses blocos são projetados para se encaixar entre si, eliminando erros de sintaxe (SCRATCH, 2017), conforme ilustrado na figura 8.

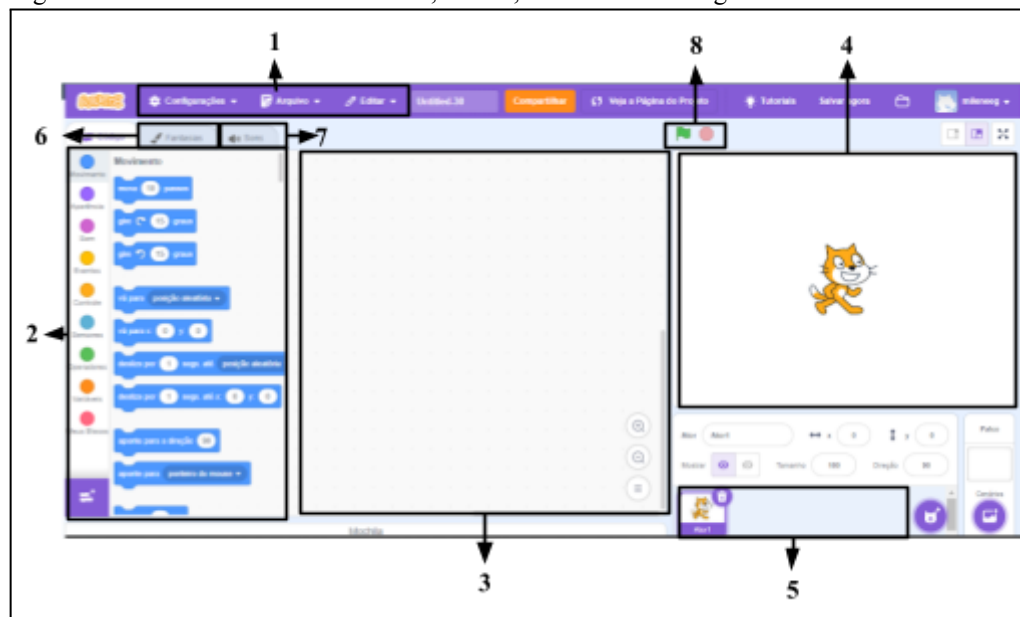
Figura 8 - Blocos de comandos do Scratch e ordem lógica de uma programação



Fonte: Autora, 2024.

A Figura 9 demonstra que uma sequência lógica de passos está sendo seguida. Nesse exemplo, é possível identificar a aplicação de vários conceitos, como estruturas de repetição (loops), movimento, tempo e condições. Esses conceitos são executados à medida que os blocos de comandos são encaixados, criando uma sequência lógica de código.

Figura 9 - Tela inicial do Scratch: áreas, botões, abas e elementos gráficos



Fonte: Autora, 2024.

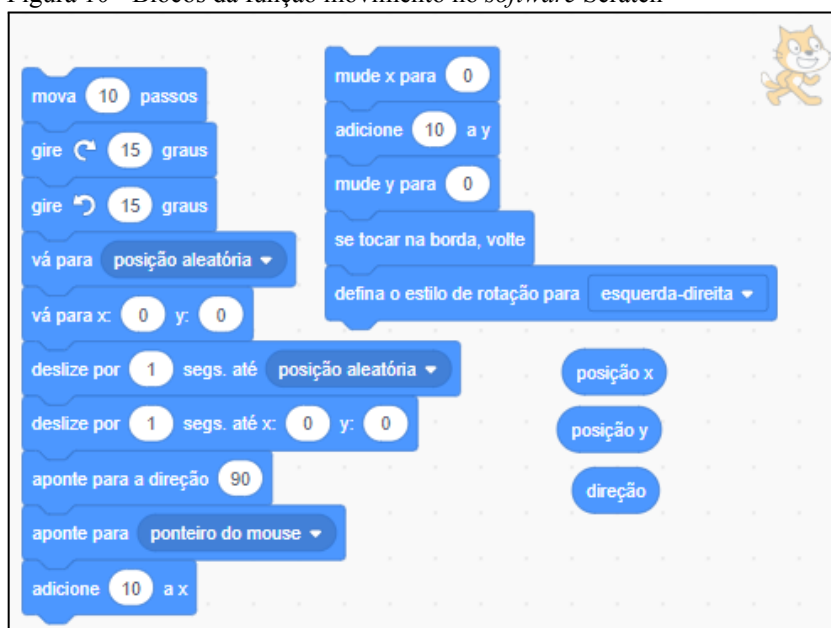
A tela mostrada na figura 9 é formada pelos controles lógicos e pelo espaço onde os comandos são manipulados e exibidos para o usuário. Uma das vantagens dessa ferramenta é

que todo o desenvolvimento ocorre em uma única tela. Essa tela contém várias funcionalidades e comandos que permitem a criação de jogos, animações e outras produções por meio de comandos de blocos lógicos.

Na tela inicial do Scratch mostrada na Figura 6, encontram-se diversos elementos essenciais para a criação e manipulação de projetos de programação. No topo, a barra de menus (1) oferece opções para criar, salvar e carregar projetos, além de acessar configurações e ajuda. À esquerda, a área de blocos (2) contém categorias de comandos, como movimento, aparência, som, eventos, controle, sensores, operadores, variáveis e meus blocos, que podem ser arrastados e soltos para formar scripts na área de scripts (3), localizada no centro da tela. À direita, o palco (4) é onde os sprites executam as ações programadas, visualizando animações, jogos e histórias. Abaixo do palco, a lista de sprites (5) permite adicionar, excluir e selecionar sprites para editar seus scripts, trajes e sons. Quando um sprite é selecionado, o editor de trajes (6) possibilita desenhar, importar e modificar suas aparências, enquanto o editor de sons (7) permite gravar, importar e editar sons para uso nos projetos. Além disso, botões de controle de projetos (8), como a bandeira verde para iniciar e o botão vermelho para parar a execução dos scripts, facilitam o teste e a visualização do projeto em tempo real. Esses componentes juntos proporcionam uma interface intuitiva e eficiente para programação visual.

A ferramenta Scratch conta com nove comandos: movimento, aparência, som, eventos, controle, sensores, operadores, variáveis e meus blocos, e descritos a seguir. A função **movimento** permite que o personagem se desloque para qualquer lugar no palco, totalizando dezoito blocos. Usando esses blocos, podemos determinar a velocidade do movimento, a direção em que o componente deve se deslocar, fazer o componente girar e definir o que acontece quando ele toca na borda da tela. Essas informações são essenciais ao programar um objeto virtual que requer movimentos.

Figura 10 - Blocos da função movimento no *software* Scratch



Fonte: Autora, 2024.

Os blocos de **aparência** são vinte, e responsáveis por alterar os trajés e criar efeitos como aparecer, desaparecer, mudar cor e tamanho, além de exibir diálogos. Com esses blocos, podemos definir se um ator fala ou pensa algo de forma escrita e por quanto tempo o texto permanece na tela. Também é possível alterar a fantasia do ator ou a imagem do cenário, além de ajustar o tamanho ou a cor do personagem. Enfim, há uma variedade de ações possíveis, sempre em conjunto com os demais blocos disponíveis.

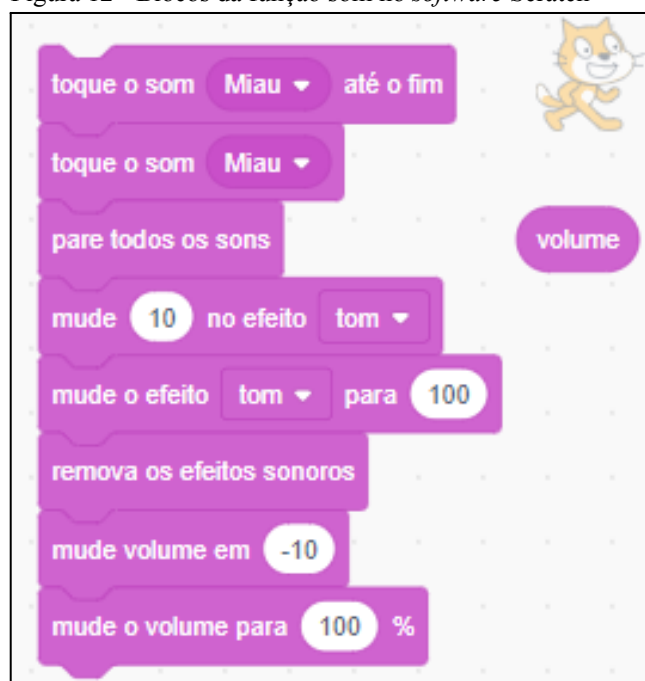
Figura 11 - Blocos da função aparência no *software* Scratch



Fonte: Autora, 2024.

A função **som** permite reproduzir e importar sons, totalizando nove blocos. Com esses blocos, podemos fazer com que um ator ou cenário execute um som. O Scratch oferece alguns efeitos sonoros integrados, mas também é possível importar sons externos, como gravações de voz. Com os blocos apropriados, é possível ajustar o volume e o tom dos efeitos sonoros. Além disso, a plataforma possui um editor de som que, embora simples, permite modificar os sons, como fazer o *upload* de uma música e editar para usar apenas um trecho específico.

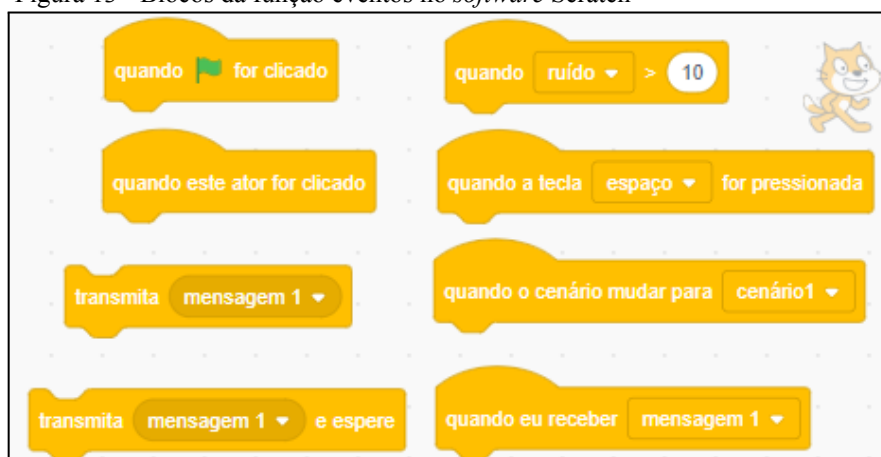
Figura 12 - Blocos da função som no *software* Scratch



Fonte: Autora, 2024.

A função **eventos** é responsável por iniciar ações, sendo essencial para a interação e coordenação entre diferentes partes do projeto, totalizando oito blocos. Se não utilizarmos um bloco de evento, uma ação pode ocorrer indefinidamente ou nem acontecer. Portanto, é fundamental usar blocos de evento para definir em quais situações e interações entre componentes, ou entre estes e o usuário, determinadas ações dos atores ou dos cenários serão conduzidas.

Figura 13 - Blocos da função eventos no *software* Scratch



Fonte: Autora, 2024.

Os onze blocos que integram a categoria **controle** contém comandos pré definidos que organizam a estrutura lógica de conexão dos blocos de comando. Esses blocos organizam a execução das ações dos componentes, como determinar quantas vezes uma ação será realizada. Por exemplo, o bloco "repita ___ vezes" faz com que todas as ações englobadas por ele se repitam o número de vezes especificado. Esses blocos, em conjunto com outras categorias, também estabelecem as condições para que a ação ocorra. Embora se assemelhem aos blocos de eventos, eles estão agrupados na categoria de controle por decisão dos desenvolvedores do Scratch.

Figura 14 - Blocos da função controle no *software* Scratch



Fonte: Autora, 2024.

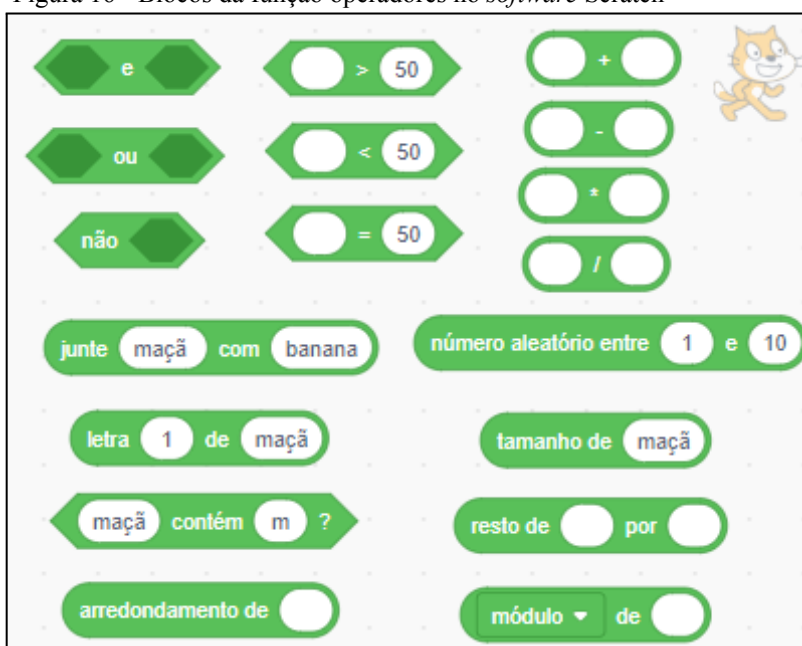
Os dezoito blocos que compõem a categoria **sensores** detectam cores, distâncias e posições, podendo ser combinados com outros comandos. Bem como definir o tipo de interação que condiciona as ações dos atores e cenários, associando essa interação com outras categorias de blocos. Por exemplo, um ator em um jogo pode se aproximar de outro ou tocá-lo. Outra interação pode ser o usuário clicando com o mouse ou pressionando uma tecla, o que desencadeia uma ação específica de um componente. Os blocos de sensores possuem formatos que indicam onde se encaixam em outros blocos. Por exemplo, o bloco "tocando na cor..." deve ser associado a blocos de controle para definir o que acontece quando o ator toca na cor escolhida. Alguns blocos de controle com formato arredondado podem ser acoplados a blocos de outras categorias para determinar um valor numérico específico, como o ano atual.

Figura 15 - Blocos da função sensores no *software* Scratch



Fonte: Autora, 2024.

A categoria **operadores** também é composta por dezoito blocos, e fornecem operadores relacionais e lógicos para realizar operações com os blocos de comando disponíveis na ferramenta. A utilização combinada dos grupos de blocos é importante para definir números que dependem de outros componentes, sua posição na tela, e outros fatores.

Figura 16 - Blocos da função operadores no *software* Scratch

Fonte: Autora, 2024.

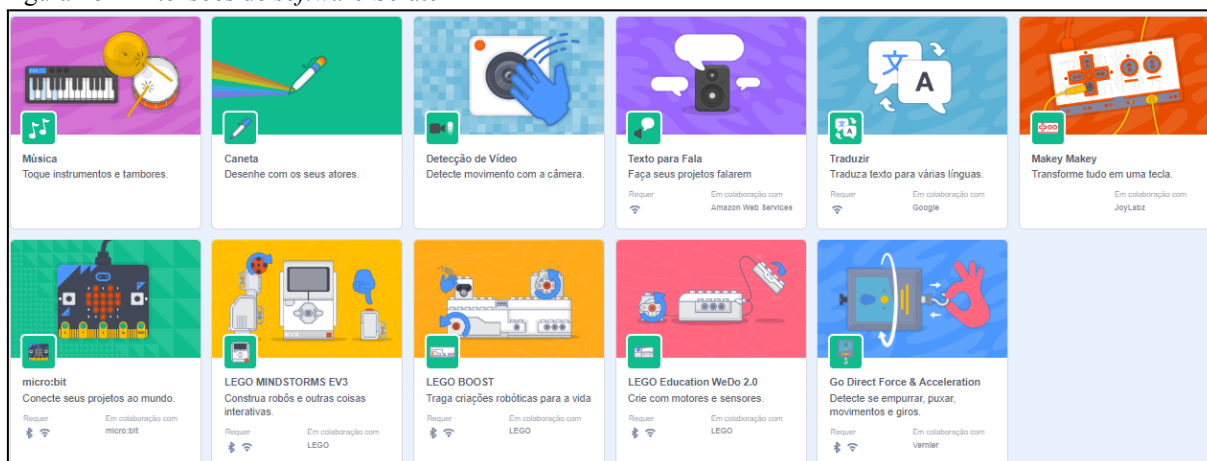
As **variáveis**, que podem ser criadas pelo usuário, armazenam valores para uso posterior e permitem a criação de listas. A função **meus blocos** permite aos usuários criar seus próprios blocos de comando personalizados no Scratch. Esses blocos personalizados podem ser utilizados para simplificar e organizar o código, facilitando a reutilização de comandos que se repetem. Isso contribui para tornar o código mais limpo e fácil de entender, além de promover a prática da programação modular.

Figura 17 - Blocos da função variáveis no *software* Scratch

Fonte: Autora, 2024.

Existem ainda as extensões do Scratch, que são adicionais e ampliam as funcionalidades da ferramenta, permitindo que os usuários integrem recursos e tecnologias externas aos seus projetos. Elas oferecem novos blocos de comando que podem ser usados para controlar *hardware*, interagir com interfaces de programação de aplicações (APIs)⁷, ou adicionar características específicas aos projetos. Entre as principais extensões estão as de **Hardware**, que permitem a integração com dispositivos externos como o micro e LEGO *Mindstorms*, possibilitando a criação de projetos interativos que envolvem elementos físicos. Extensão de **sensores e dispositivos**, que incluem funcionalidades para utilizar sensores de som, luz, movimento e outros, permitindo a criação de projetos que reagem a estímulos do ambiente. Extensão de **APIs e serviços online**, que facilitam a conexão com serviços online e APIs, permitindo a criação de projetos que utilizam dados da *web* ou interagem com outros aplicativos. A extensão **mídia**, que adiciona blocos para manipular e criar conteúdo multimídia, como gráficos e sons. Além disso, a extensão **caneta** permite desenhar no palco usando blocos específicos, oferecendo mais opções criativas para projetos que envolvem gráficos e ilustrações. Essas extensões ajudam a expandir as possibilidades criativas e técnicas do Scratch, permitindo aos usuários explorar novas áreas e realizar projetos mais complexos e variados.

Figura 18 - Extensões do *software* Scratch



Fonte: Autora, 2024.

O Scratch é composto por um palco e diversos objetos. O palco serve como o plano de fundo onde são executados os comandos em conjunto com as ações dos objetos, que podem estar relacionados a sons, imagens, variáveis, estruturas de decisão, repetição e lógica

⁷ É a sigla para *Application Programming Interfaces*, que em português significa Interfaces de Programação de Aplicações.

matemática. Para programar na ferramenta, é necessário arrastar e encaixar os blocos de comando de forma adequada, seguindo os encaixes específicos, para desenvolver o projeto desejado (MALAN; LEITNER, 2007).

Com base no exposto anteriormente, no capítulo 3, a argumentação dedutiva e o raciocínio lógico desempenham papéis importantes no processo de programar no Scratch. A argumentação dedutiva permite que os alunos construam e testem hipóteses baseadas em regras e informações fornecidas, ajudando a estruturar seu pensamento de forma sistemática. O raciocínio lógico é basilar para seguir a sequência correta de operações e decisões, garantindo que o programa resolva problemas de maneira eficiente e eficaz.

Considerando o que foi apresentado no Capítulo 4, é relevante estimular o pensamento computacional durante o processo de programação no Scratch, pois ele não só apoia a decomposição de problemas complexos em partes menores e mais manejáveis, mas também promove a identificação de padrões e a abstração de informações pertinentes. Isso permite o desenvolvimento de algoritmos mais eficazes e soluções inovadoras.

5.5 Organização das atividades propostas

As atividades têm como objetivo introduzir a programação utilizando o Scratch. Elas exigem que os alunos aprendam conceitos básicos, como movimentação, respeito aos limites, e o uso de estruturas simples de condição e repetição. Apesar da simplicidade, essas atividades demandam que os alunos sigam uma ordem lógica e apliquem o pensamento computacional, o raciocínio lógico e a argumentação dedutiva para alcançar os objetivos propostos. Além disso, utilizou-se da observação participativa para enriquecer a coleta de dados, proporcionando uma compreensão mais profunda do processo de aprendizagem dos alunos.

A atividade 1 tem o intuito de iniciar a programação no Scratch e para isso o aluno foi direcionado a utilizar algumas funções contidas no Scratch, como o movimento, o controle e a aparência, as quais são necessárias para que a atividade seja solucionada.

Quadro 23 - Proposta da atividade 1

ATIVIDADE 1
Desenvolva um programa que faça o personagem mover 10 passos, esperar um segundo e trocar de fantasia, quando o personagem tocar na borda ele vira e continua andando. Utilizar uma estrutura de repetição para que o personagem não pare de andar.

CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE

MOVIMENTO	X		CONTROLE	X
APARÊNCIA	X		SENSORES	
SOM			OPERADORES	
CANETA			VARIÁVEIS	

PILARES DO PENSAMENTO COMPUTACIONAL

ABSTRAÇÃO

- Quantos passos precisa andar?
- Precisa esperar?
- Quantos segundos?
- É necessário trocar de fantasia?
- O que devo fazer se tocar na borda?
- Posso parar de andar?

RECONHECIMENTO DE PADRÕES

- Mover 10 passos;
- Esperar 1 segundo;
- Tocar na borda e voltar.

DECOMPOSIÇÃO

- Estrutura de repetição.

ALGORITMO



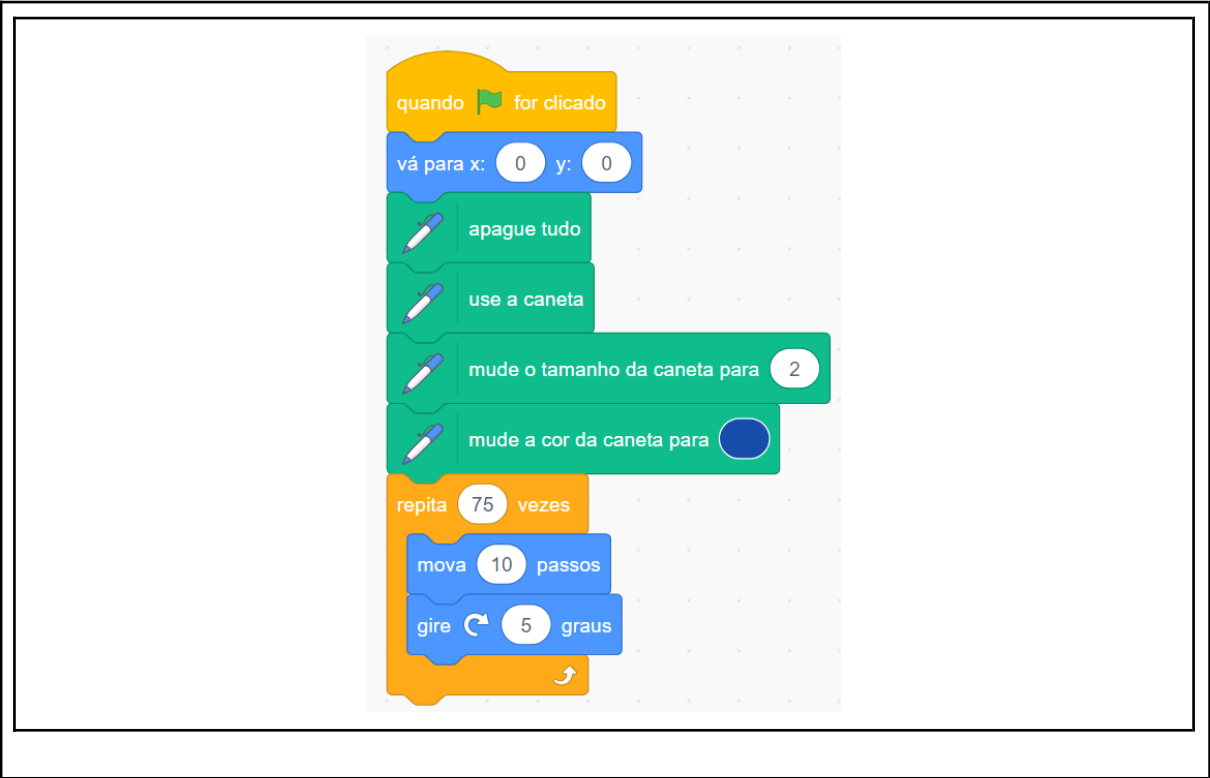
Fonte: Autora, 2024.

A atividade 2 busca explorar a lógica de programação, com o intuito de trabalhar com a parte gráfica da ferramenta. Por mais que não seja o foco desse trabalho, esses exercícios

demonstram que o Scratch permite trabalhar não apenas com os algoritmos, mas também com desenhos, gráficos e atividades lúdicas, almejando chamar a atenção do aluno. O exercício abaixo exige que o aluno consiga utilizar um algoritmo a fim de desenhar um círculo, e para isso o mesmo precisa ter algumas noções básicas não só de programação para construir o algoritmo, mas sim noções geométricas e utilizar o pensamento computacional a fim de solucionar a problemática.

Quadro 24 - Proposta da atividade 2

ATIVIDADE 2				
Desenvolva um algoritmo que mova o personagem e desenhe um círculo.				
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE				
MOVIMENTO	X		CONTROLE	X
APARÊNCIA			SENSORES	
SOM			OPERADORES	
CANETA	X		VARIÁVEIS	
PILARES DO PENSAMENTO COMPUTACIONAL				
ABSTRAÇÃO • Que posição o personagem precisa estar? • Quantos passos é necessário? • Quantos graus meu personagem vai girar? • Posso repetir minhas ações?				
RECONHECIMENTO DE PADRÕES • Mover 10 passos; • Girar 5°; • Utilizar a caneta.				
DECOMPOSIÇÃO • Estrutura de repetição				
ALGORITMO				



Fonte: Autora, 2024.

A atividade 3 aborda os mesmos princípios da atividade 2, visando que os alunos aprendam sobre figuras geométricas e suas características, além de aplicar seus conhecimentos aritméticos para desenhar um quadrado utilizando o Scratch por meio dos blocos de programação.

Quadro 25 - Proposta da atividade 3

ATIVIDADE 3				
Desenvolva um algoritmo que mova o personagem e desenhe um quadrado.				
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE				
MOVIMENTO	X		CONTROLE	X
APARÊNCIA			SENSORES	X
SOM			OPERADORES	
CANETA	X		VARIÁVEIS	
PILARES DO PENSAMENTO COMPUTACIONAL				
ABSTRAÇÃO				

- Que ângulo tem um quadrado?
- Esse ângulo se repete? Quantas vezes?
- Quantos lados tem um quadrado?
- A medida dos lados de um quadro é igual?
- Posso repetir meus movimentos?

RECONHECIMENTO DE PADRÕES

- Mover 100 passos;
- Girar 90°;
- Utilizar a caneta.

DECOMPOSIÇÃO

- Estrutura de repetição

ALGORITMO



Fonte: Autora, 2024.

A atividade 4 consiste em uma problemática comum, a qual se faz presente no dia a dia dos alunos, que é a multiplicação de numerais. Nessa perspectiva a atividade 4 exige que o aluno desenvolva um algoritmo para solucionar a multiplicação de dois numerais, e que o mesmo desenvolva uma condição para validar a resposta inserida pelo usuário.

Quadro 26 - Proposta da atividade 4

ATIVIDADE 4				
Desenvolva um programa que pergunte ao usuário: Quanto é 5 x 5? Em seguida o usuário digita a resposta e o programa faz a análise exibindo ao final, se a resposta está certa ou errada.				
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE				
MOVIMENTO	X		CONTROLE	X
APARÊNCIA	X		SENSORES	
SOM			OPERADORES	X
CANETA			VARIÁVEIS	
PILARES DO PENSAMENTO COMPUTACIONAL				
ABSTRAÇÃO • O personagem precisa perguntar? O que? • É necessário utilizar a resposta do usuário? • Eu preciso comparar valores? • O valor inserido precisa ser comparado a quem? • O personagem precisa dizer alguma coisa? O que?				
RECONHECIMENTO DE PADRÕES • Numerais; • Validação de dados.				
DECOMPOSIÇÃO • Estrutura de condição; • Comparação de valores.				
ALGORITMO 				

Fonte: Autora, 2024.

A atividade 5 além de trabalhar com a lógica de programação, também exige que o aluno trabalhe com o plano cartesiano, utilize uma estrutura de condição e repetição, além de criar variáveis para fazer a validação da idade do aluno. A atividade é considerada simples, de acordo com o público que está sendo trabalhada, porém é uma atividade rica em detalhes, ou seja, para que ela seja solucionada é necessário que o aluno saiba utilizar a lógica de programação e pensar computacionalmente, a fim de fazer a abstração, criar padrões, decomposição e por fim desenvolver um algoritmo válido.

Quadro 27 - Proposta da atividade 5

ATIVIDADE 5				
Crie um programa que ao ser executado pelo usuário, o personagem se move até a posição x: -200 e y: 0. Troque de fantasia, mova 10 passos e espere um segundo, repita essa função 5 vezes. Em seguida, o personagem pergunta ao usuário qual sua idade, o usuário insere o valor e o programa exibe uma mensagem informando se o mesmo é maior ou menor de idade.				
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE				
MOVIMENTO	X		CONTROLE	X
APARÊNCIA	X		SENSORES	X
SOM			OPERADORES	X
CANETA			VARIÁVEIS	X
PILARES DO PENSAMENTO COMPUTACIONAL				
ABSTRAÇÃO • Que posição o personagem precisa estar? • É necessário trocar o traje? Vai repetir esta ação? • É necessário mover o personagem? Quantos passos? Vai repetir esta ação? • O personagem precisa perguntar? O que? • Eu preciso criar variáveis? Quantas? Porque? • Eu preciso comparar valores? • O valor inserido precisa ser comparado a quem? • O personagem precisa dizer alguma coisa? O que?				
RECONHECIMENTO DE PADRÕES • Numerais; • Validação de dados.				
DECOMPOSIÇÃO • Estrutura de condição; • Estrutura de repetição; • Armazenamento de dados; • Comparação de valores.				

ALGORITMO

quando  for clicado

vá para x: -200 y: 0

repita 5 vezes

próxima fantasia

mova 10 passos

espere 1 seg

pergunte Qual a sua idade? e espere

mude idade ▼ para resposta

se idade > 17 então

diga Você é maior de idade por 2 segundos

senão

diga Você é menor de idade por 2 segundos

apague tudo

Fonte: Autora, 2024.

A atividade 6 tem como principal objetivo trabalhar com estrutura de condição, a fim de verificar se o número inserido pelo usuário é par ou ímpar, entretanto é necessário que o aluno desenvolva um algoritmo capaz de validar esses resultados.

Quadro 28 - Proposta da atividade 6

ATIVIDADE 6			
Desenvolva um algoritmo onde o usuário insere um valor inteiro e o programa aponta se esse valor é um número par ou ímpar.			
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE			
MOVIMENTO		CONTROLE	X

APARÊNCIA	X		SENSORES	X
SOM			OPERADORES	X
CANETA			VARIÁVEIS	X

PILARES DO PENSAMENTO COMPUTACIONAL

ABSTRAÇÃO

- O personagem precisa perguntar? O que?
- Eu preciso criar variáveis? Quantas? Porque?
- Como eu identifico um número par? E um ímpar?
- Eu preciso comparar valores?
- O valor inserido precisa ser comparado a quem?
- O personagem precisa dizer alguma coisa? O que?

RECONHECIMENTO DE PADRÕES

- Inserção de numerais;
- Validação de dados.

DECOMPOSIÇÃO

- Estrutura de condição;
- Armazenamento de dados;
- Comparação de valores.

ALGORITMO

```
quando for clicado
  pergunte Digite um número e espere
  mude numero para resposta
  se resto de numero por 2 = 0 então
    diga o número é par por 2 segundos
  senão
    diga o número é ímpar por 2 segundos
```

Fonte: Autora, 2024.

A atividade 7 objetiva trabalhar com a média de numerais, exigindo do aluno a criação de variáveis para armazenar e tratar os dados inseridos pelos usuários, além de que essas variáveis também são responsáveis por exibir informações ao usuário. Para isso é necessário que o aluno tenha noções básicas de como calcular a média dos numerais e saber como transpor o pensamento computacional por meio de um algoritmo que seja válido.

Quadro 29 - Proposta da atividade 7

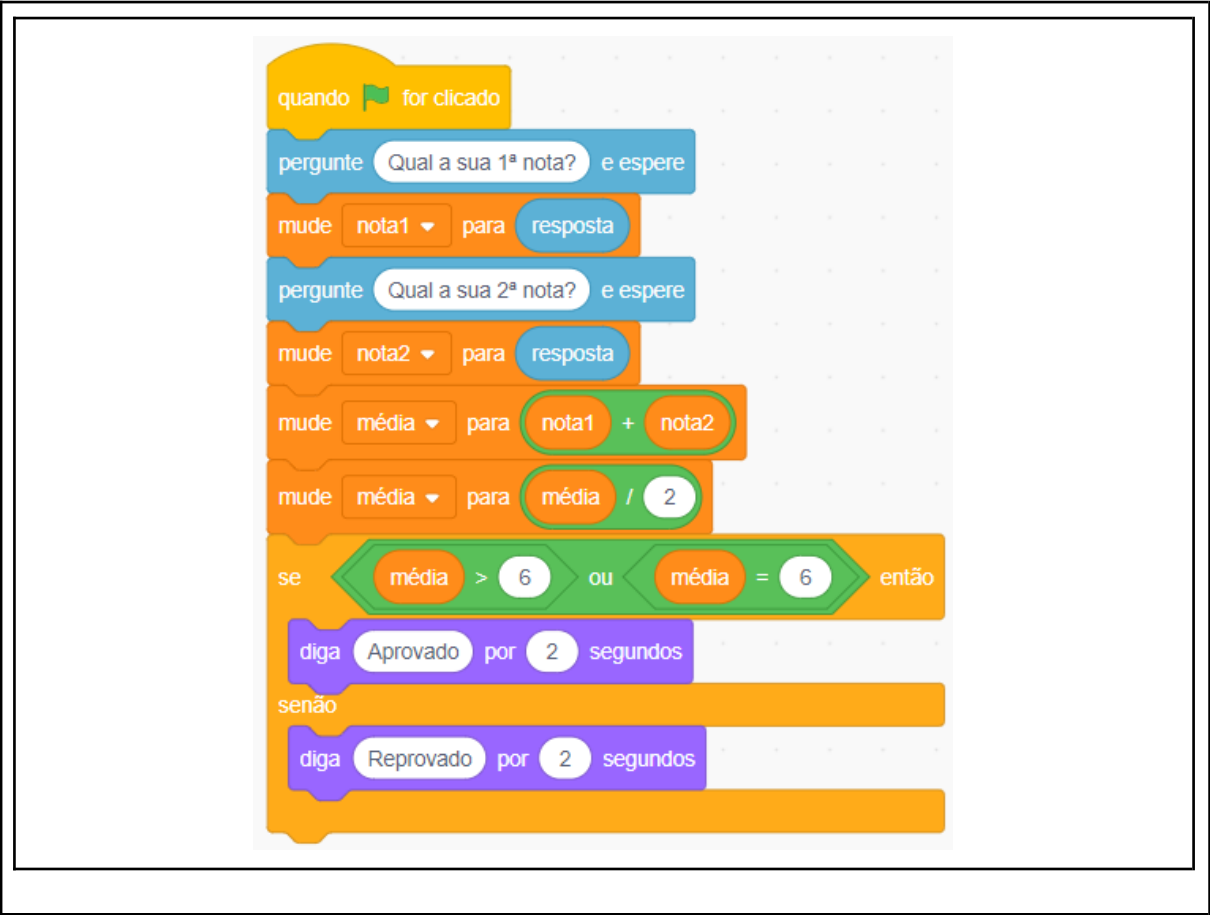
ATIVIDADE 7				
Crie um programa que calcula a média de duas notas inseridas pelo usuário e informe o valor da média				
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE				
MOVIMENTO			CONTROLE	X
APARÊNCIA	X		SENSORES	X
SOM			OPERADORES	X
CANETA			VARIÁVEIS	X
PILARES DO PENSAMENTO COMPUTACIONAL				
ABSTRAÇÃO • O personagem precisa perguntar? O que? • Eu preciso criar variáveis? Quantas? Porque? • Como eu faço o cálculo de uma média? • Que operações são necessárias? • O personagem precisa dizer alguma coisa? O que?				
RECONHECIMENTO DE PADRÕES • Inserção de numerais; • Validação de dados.				
DECOMPOSIÇÃO • Armazenamento de dados.				
ALGORITMO 				

Fonte: Autora, 2024.

A atividade 8 tem como objetivo avaliar a capacidade do aluno em criar variável, efetuar cálculos, inserir e exibir valores, além de construir o código. A atividade exige que o aluno utilize uma estrutura de condição para verificar a média, e notificar o usuário se o mesmo está aprovado ou reprovado. Com a utilização da ferramenta Scratch, a mesma exige que o aluno utilize as seguintes funções: controle, variáveis, operadores, sensores e aparência.

Quadro 30 - Proposta da atividade 8

ATIVIDADE 8				
Crie um programa que calcula a média de duas notas inseridas pelo usuário. Em seguida, caso o valor gerado pela média das notas for maior ou igual a 6 exiba que o aluno está aprovado, caso contrário, o aluno está reprovado.				
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE				
MOVIMENTO			CONTROLE	X
APARÊNCIA	X		SENSORES	X
SOM			OPERADORES	X
CANETA			VARIÁVEIS	X
PILARES DO PENSAMENTO COMPUTACIONAL				
ABSTRAÇÃO • O personagem precisa perguntar? O que? • Eu preciso criar variáveis? Quantas? Porque? • Como eu faço o cálculo de uma média? • Que operações são necessárias? • Eu preciso comparar valores? • O valor inserido precisa ser comparado a quem? • O personagem precisa dizer alguma coisa? O que?				
RECONHECIMENTO DE PADRÕES • Inserção de numerais; • Validação de dados; • Agrupamento de valores.				
DECOMPOSIÇÃO • Estrutura de condição; • Armazenamento de dados; • Comparação de valores.				
ALGORITMO				



Fonte: Autora, 2024.

A atividade 9 tem como objetivo avaliar a capacidade do aluno em criar variáveis, efetuar cálculos, inserir e exibir valores, além de construir o código. A atividade exige que o aluno utilize uma estrutura de condição para verificar a idade, e notificar o usuário sobre a sua idade.

Quadro 31 - Proposta da atividade 9

ATIVIDADE 9				
Crie um algoritmo que pergunte ao usuário seu ano de nascimento e o ano atual. Armazene as duas respostas. Calcule e diga a idade do usuário.				
CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE				
MOVIMENTO			CONTROLE	X
APARÊNCIA	X		SENSORES	X
SOM			OPERADORES	X
CANETA			VARIÁVEIS	X

PILARES DO PENSAMENTO COMPUTACIONAL	
ABSTRAÇÃO	• O personagem precisa perguntar? O que? • Eu preciso criar variáveis? Quantas? Porque? • Como eu faço o cálculo da idade de uma pessoa com base no ano? • Que operações são necessárias? • O personagem precisa dizer alguma coisa? O que?
RECONHECIMENTO DE PADRÕES	• Inserção de numerais; • Validação de dados.
DECOMPOSIÇÃO	• Armazenamento de dados.
ALGORITMO	 The image shows a Scratch script designed to calculate a person's age. It begins with a 'when green flag is clicked' event block. The first sequence consists of a 'ask' block asking 'Qual o seu ano de nascimento?' followed by a 'wait' block. This is followed by a 'set' block for the variable 'nascimento' to the 'response' value. The second sequence is another 'ask' block asking 'Qual o ano atual?' followed by a 'wait' block, then a 'set' block for the variable 'ano' to the 'response' value. The third sequence is a 'set' block for the variable 'idade' to the calculation 'ano - nascimento'. Finally, a 'say' block displays the text 'Você tem' followed by the 'idade' variable, then 'com' followed by the 'anos' variable, and ends with 'por 2 segundos'.

Fonte: Autora, 2024.

O objetivo da atividade 10 é avaliar se os alunos conseguem pensar computacionalmente e utilizar estruturas lógicas a fim de desenvolverem um algoritmo para efetuar operações simples com dois dígitos. Aparentemente parece um simples problema, porém o aluno necessita usar raciocínios lógicos e estruturas específicas para decompor o problema em problemas menores visto que estamos trabalhando com quatro operações distintas.

Quadro 32 - Proposta da atividade 10

ATIVIDADE 10
Desenvolva um algoritmo cujo o usuário escolha qual operação deve ser realizada: adição, subtração, divisão

e multiplicação, em seguida são inseridos dois valores pelo usuário, a operação é realizada e o resultado é exibido.

CATEGORIAS DE COMANDOS UTILIZADOS NO SCRATCH PARA PROGRAMAR A ATIVIDADE

MOVIMENTO			CONTROLE	X
APARÊNCIA	X		SENSORES	X
SOM			OPERADORES	X
CANETA			VARIÁVEIS	X

PILARES DO PENSAMENTO COMPUTACIONAL

ABSTRAÇÃO

- O personagem precisa perguntar? O que?
- Eu preciso criar variáveis? Quantas? Porque?
- Que operações são necessárias?
- O que é necessário para realizar cada uma das operações?
- O personagem precisa dizer alguma coisa? O que?

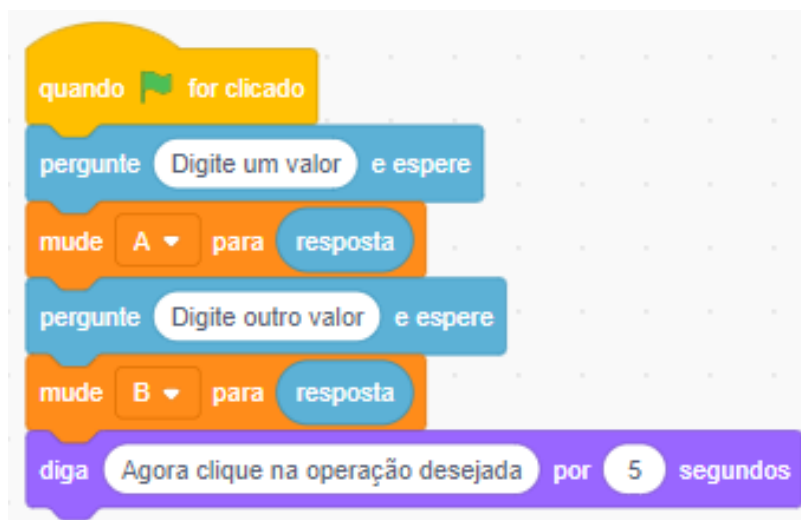
RECONHECIMENTO DE PADRÕES

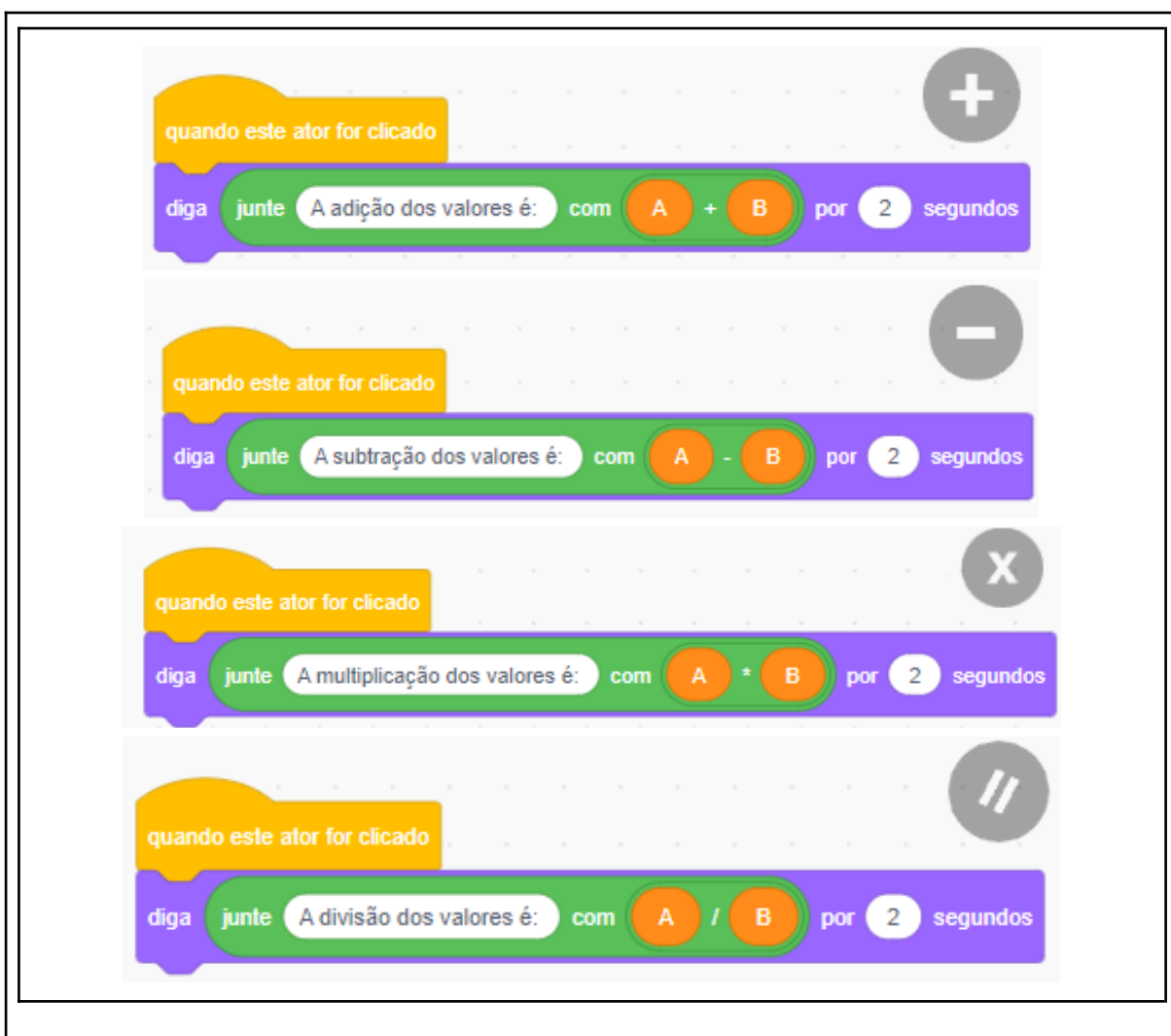
- Inserção de numerais;
- Operações matemáticas;
- Validação de dados.

DECOMPOSIÇÃO

- Estrutura de condição;
- Armazenamento de dados.

ALGORITMO





Fonte: Autora, 2024.

Ao integrar esses elementos relevantes para esta tese, como conceitos básicos de programação, pensamento computacional, raciocínio lógico e argumentação dedutiva, espera-se que as atividades não apenas introduzam aos alunos à programação de forma acessível, mas também fomentem habilidades críticas. A observação participativa permitirá uma análise detalhada e contextualizada do processo de aprendizagem, oferecendo percepções valiosas para aprimorar a leitura dos dados coletados.

Os conceitos básicos de programação, como movimentação, respeito aos limites e uso de estruturas de condição e repetição, são essenciais para estabelecer uma base sólida. Esses conceitos introduzem os alunos às práticas fundamentais da codificação e desenvolvem suas habilidades técnicas.

O pensamento computacional apresenta-se de forma relevante nesta estrutura para a resolução de problemas de forma estruturada e eficiente, envolvendo a decomposição de problemas complexos em partes mais simples, a identificação de padrões, a abstração de

detalhes irrelevantes e o desenvolvimento de algoritmos. Ao aplicar o pensamento computacional, os alunos aprendem a abordar problemas de maneira lógica e organizada, como abordado anteriormente no capítulo 4 deste estudo.

O raciocínio lógico é uma habilidade necessária que permite aos alunos seguirem uma sequência lógica de passos para resolver problemas, sendo importante para entender como diferentes partes de um programa interagem e para prever os resultados das instruções dos blocos do programa. O desenvolvimento do raciocínio lógico ajuda os alunos a construir soluções eficazes e a identificar e corrigir erros no código. A argumentação dedutiva, por sua vez, capacita os alunos a justificar e explicar o raciocínio por trás de suas soluções. Essa habilidade é importante não apenas para a programação, mas também para a comunicação clara de ideias e para a capacidade de pensar criticamente sobre problemas e soluções. Temas abordados anteriormente no capítulo 3 deste estudo.

A observação participativa enriquece a coleta de dados ao permitir uma compreensão mais profunda do processo de aprendizagem dos alunos. Ao observar como os alunos interagem com as atividades e aplicam os conceitos aprendidos, os educadores podem obter percepções valiosas sobre as dificuldades enfrentadas e as estratégias utilizadas pelos alunos.

5.6 Considerações parciais

A estrutura metodológica da pesquisa sobre a relação entre pensamento computacional e argumentação dedutiva serão baseadas em uma abordagem qualitativa, permitindo uma exploração aprofundada das experiências dos alunos e das dinâmicas de aprendizado. O estudo foi realizado com alunos do Ensino Fundamental II da Escola Estadual de Ensino Fundamental Mathias Lorenzon, localizada em Getúlio Vargas-RS, especificamente com os participantes do projeto Conect +. Essa escolha de contexto possibilitou observar diretamente como as habilidades de programação e resolução de problemas influenciam a construção de argumentos. A coleta de dados incluiu observação direta, com registros sistemáticos das interações em sala de aula e atividades do projeto, e um estudo de caso que analisou os dois alunos, focando em suas abordagens para resolver problemas e formular argumentos. Foram utilizados um teste para avaliar a compreensão dos alunos sobre os conceitos de pensamento computacional e argumentação, além de um questionário estruturado para explorar suas percepções sobre como a programação e o pensamento lógico impactam suas habilidades argumentativas. A análise dos dados foi realizada por meio de análise de conteúdo, permitindo identificar temas recorrentes e evidências das conexões entre as habilidades

estudadas, e análise de caso, que proporcionou um entendimento mais profundo das interações e desempenhos dos alunos. Considerações éticas foram observadas em todas as etapas da pesquisa, assegurando a privacidade e o consentimento informado dos participantes. Espera-se que os resultados contribuam para a compreensão teórica das relações entre pensamento computacional e argumentação dedutiva, além de fornecer compreensão práticos sobre como integrar essas competências no currículo escolar, potencializando o aprendizado dos alunos.

6 RESULTADOS DA PESQUISA EMPÍRICA

O desenvolvimento de habilidades cognitivas complexas, como o pensamento computacional e a argumentação dedutiva, é fundamental no contexto educacional contemporâneo. O pensamento computacional, que envolve resolver problemas de forma estruturada e lógica, é cada vez mais valorizado, especialmente com a integração das tecnologias digitais no currículo escolar. Da mesma forma, a argumentação dedutiva, que permite aos alunos construir e avaliar argumentos com base em premissas e conclusões, desempenha um papel importante na formação do pensamento crítico. Essa habilidade permite aos estudantes não apenas analisar informações de maneira rigorosa, mas também questionar, interpretar e tomar decisões de forma fundamentada, essencial em um mundo onde a capacidade de filtrar e avaliar dados é mais necessária do que nunca.

Este estudo visou explorar a relação entre o pensamento computacional e a argumentação dedutiva por meio de uma pesquisa empírica qualitativa. A pesquisa foi conduzida em três etapas principais: observação participante, aplicação de um teste de pensamento computacional e a administração de um questionário focado na argumentação dedutiva.

A observação participante foi utilizada para obter uma compreensão aprofundada das práticas e comportamentos dos alunos durante atividades relacionadas ao pensamento computacional. Essa abordagem permitiu a coleta de dados contextuais sobre como os alunos interagem com tarefas de programação e como eles aplicam estratégias de resolução de problemas em situações reais.

O teste de pensamento computacional foi aplicado para avaliar as habilidades dos alunos em resolver problemas de programação e em usar algoritmos e lógica de maneira eficiente. Este teste proporcionou uma medida quantitativa das competências dos alunos em relação ao pensamento computacional.

Finalmente, um questionário voltado à argumentação dedutiva foi desenvolvido para avaliar a capacidade dos alunos de construir e analisar argumentos lógicos. O questionário incluiu perguntas que desafiaram os alunos a aplicar princípios de argumentação dedutiva em diferentes contextos.

Ao integrar esses métodos, o estudo buscou identificar padrões e correlações entre o desenvolvimento do pensamento computacional e a argumentação dedutiva, fornecendo observações valiosas para a prática educacional e para futuras pesquisas na área.

6.1 Observação participativa dos encontros

Guiando-se pelos questionamentos planejados durante o processo de resolver problemas com o Scratch, observa-se uma diferença na externalização de ideias entre o aluno A e o aluno B. O aluno A procura argumentar e justificar suas decisões, enquanto o aluno B, além de argumentar e justificar suas decisões, demonstra através de esquemas que nortearam suas programações no Scratch, conforme demonstrado a seguir.

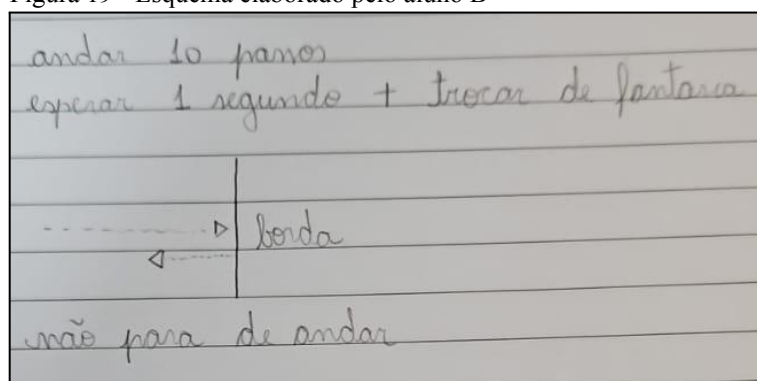
O aluno A afirmou durante a atividade 1,

*o meu personagem precisa se mover 10 passos, esperar 1 segundo, trocar de fantasia e repetir isso infinitamente, **para resolver o problema**. Se em algum momento ele tocar na borda, vai girar, voltar e continuar andando, **porque** eu vou estar usando o bloco de sempre e meu personagem não para nunca enquanto o programa estiver rodando, **assim** eu resolvi tudo que está pedindo. (A).*

Já o aluno B expôs seu raciocínio através da fala e de uma representação, pontuando o seguinte,

*prof, em cada momento o personagem vai andar 10 passos, esperar 1 segundo, trocar a fantasia, se ele tocar na borda vai voltar e continuar andando **por causa** do bloco que podemos usar, e o personagem não vai parar de andar; **porque** o próprio problema tá dizendo pra nós usar o bloco de repetição, que quer dizer que não para nunca, e **assim** eu encontrei os meios para resolver o problema. [...] Eu construí um passo a passo para programar e representar o que eu disse. (B).*

Figura 19 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Considerando os indicadores de inferência de Fávero et al. (2017), os alunos A e B conseguiram elaborar um argumento dedutivo válido. Eles indicaram seus passos e premissas usando a palavra “porque” e concluíram com “assim”. Isso evidencia que suas premissas são verdadeiras e foram testadas pelos respectivos programas Scratch, resultando em uma conclusão verdadeira e diretamente ligada à premissa.

O aluno A apresentou uma explicação clara e direta, afirmando que para resolver o problema, o personagem deveria se mover 10 passos, esperar 1 segundo, trocar de fantasia e repetir infinitamente. Ele argumentou que, ao usar o bloco de repetição, o personagem não pararia enquanto o programa estivesse rodando e que, ao tocar na borda, o personagem giraria e continuaria andando, conforme o bloco utilizado. Sua solução foi baseada em uma compreensão direta do funcionamento dos blocos e na aplicação dos conceitos básicos do Scratch.

Por outro lado, o aluno B expôs seu raciocínio de maneira detalhada tanto verbalmente quanto através de uma representação visual. Ele explicou que o personagem deveria andar 10 passos, esperar 1 segundo, trocar a fantasia, e, ao tocar na borda, voltar e continuar andando. O aluno B fez uso do bloco de repetição para garantir que o personagem não parasse, reforçando que a natureza contínua do bloco de repetição estava alinhada com as exigências do problema. Adicionalmente, ele elaborou um passo a passo para organizar sua programação e garantir que a solução fosse bem representada e compreendida.

A principal diferença entre os alunos reside na abordagem: o aluno A focou na aplicação direta dos blocos, enquanto o aluno B adotou uma estratégia mais sistemática, utilizando representações visuais e um planejamento detalhado. Ambos os métodos refletem uma compreensão válida do problema e da solução, mas com níveis variados de detalhamento e planejamento, evidenciando a importância de uma abordagem estruturada para resolver problemas complexos no Scratch.

No decorrer da atividade 2, o aluno A demonstrou uma certa insegurança sobre como tomar suas decisões, mas após um curto período de tempo passou a refletir e ao final expôs que,

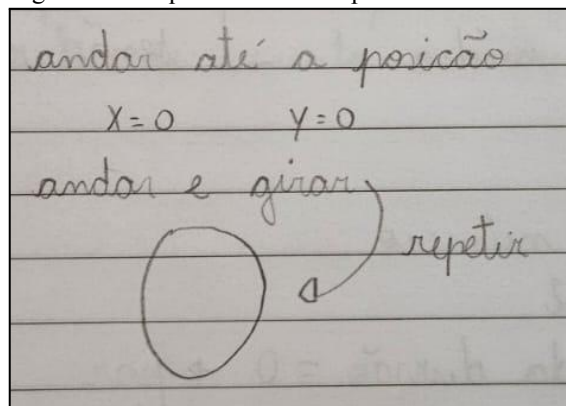
*demorei para perceber que precisava definir uma posição inicial para o meu personagem, **assim** que percebi e fiz essa definição, pude determinar se o personagem iria girar no sentido horário ou anti-horário, o que não iria alterar a resposta final neste caso. Precisei lembrar também que um círculo tem 360 graus, mas me questionei durante um tempo quantos graus e quantos passos meu personagem iria andar cada vez para completar o círculo. A partir daí, **observei** que se meu personagem fizesse rotações pequenas e andasse poucos passos por vez era simples de desenhar o círculo, **então** adicionei o bloco de repetição e fui testando a quantidade de repetições necessárias **para que** o círculo se fechasse. Foi **assim** que eu resolvi esse problema (A).*

Novamente o aluno B expôs seu raciocínio através da fala e de uma representação, pontuando o seguinte,

dessa vez eu achei melhor criar um guia, o passo a passo antes de qualquer coisa, só com as informações que tinha no problema, sem pensar muito nos blocos que tem dentro do Scratch, e

*quando parei para pensar precisei apagar, **porque** se eu não definir uma posição inicial, qualquer que seja, o meu personagem pode sair da tela de exibição e desenhar o círculo onde eu não esteja vendo. Depois que defini a posição era só girar e andar, e **assim** que eu parei pra pensar em ângulos notei que quanto maior o ângulo no giro, mais distante de um círculo ficaria minha figura, **então** com a repetição de um giro de 5 graus e 10 passos cada vez, consegui fechar o círculo, resolvendo o problema (B).*

Figura 20 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Com base nos indicadores de inferência de Fávero et al. (2017), conclui-se que os alunos A e B demonstraram a elaboração de um argumento dedutivo válido ao programar no Scratch, embora com abordagens diferentes. O aluno A, inicialmente inseguro sobre como tomar suas decisões, refletiu ao longo do processo e percebeu a necessidade de definir uma posição inicial para o personagem. Ele explicou que, ao definir a posição e determinar a direção de rotação, conseguiu compreender que a soma dos ângulos deveria completar 360 graus para formar um círculo. Ele notou que pequenas rotações e passos curtos facilitavam o desenho do círculo, utilizando um bloco de repetição para testar até que o círculo estivesse completo.

O aluno B, por outro lado, começou com a criação de um guia passo a passo baseado nas informações do problema, sem se preocupar inicialmente com os blocos do Scratch. Ele destacou a importância de definir uma posição inicial para evitar que o personagem saísse da tela. Ao refletir sobre os ângulos, ele notou que menores ângulos resultariam em uma figura mais próxima de um círculo. Com isso, ele programou o personagem para girar 5 graus e andar 10 passos repetidamente até fechar o círculo.

Ambos os alunos utilizaram o raciocínio lógico e a argumentação dedutiva para justificar suas decisões e validar suas premissas, resultando em conclusões verdadeiras e coerentes com os resultados obtidos no Scratch. O aluno A demonstrou uma progressão na compreensão e confiança ao longo do processo, enquanto o aluno B mostrou uma abordagem

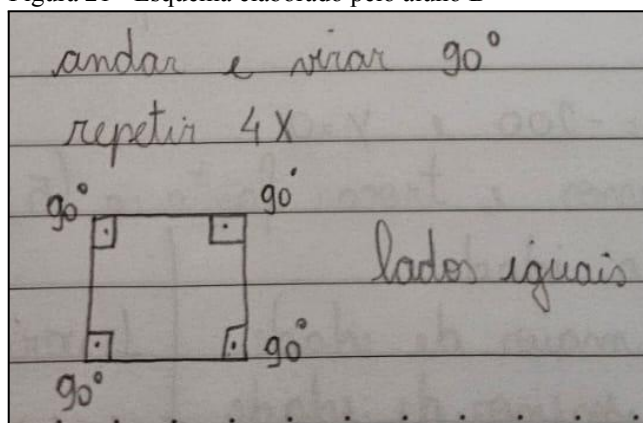
sistemática e organizada desde o início. Esses métodos diferentes evidenciam como a reflexão e a preparação podem influenciar na resolução de problemas e na construção de soluções eficazes.

Durante a atividade 3, tanto o aluno A quanto o aluno B demonstraram facilidade para resolver o problema. Eles conseguiram expor e justificar seus caminhos de forma clara, apresentando argumentos detalhados que evidenciaram seu entendimento dos conceitos envolvidos e a lógica por trás de suas decisões no Scratch.

*um quadrado tem quatro lados iguais e os ângulos do quadrado também são iguais, que é de 90 graus, **portanto** precisei definir esta medida em passos do meu personagem, girar 90 graus e repetir quatro vezes. A quantidade de passos do personagem que definiu a medida do lado do quadrado, que poderia ser maior ou menor, depende da decisão de quem está programando e eu defini 50 passos como medida do lado. [...] Como meu personagem desenhou um quadrado, eu acredito que tomei as decisões corretas enquanto programava. (A).*

O aluno B descreveu que “um quadrado tem quatro lados de medidas iguais e quatro ângulos de 90 graus, e isso me permitiu programar um único lado, como andar 60 passos e girar 90 graus dentro de um bloco de repetição, **então** coloquei para repetir 4x e estava resolvido.” Muito seguro de si, afirma que só “consegui ter essas percepções rápidas e programar com facilidade, **porque** construir o passo a passo antes de programar me ajuda a pensar e tomar as decisões corretas.”

Figura 21 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Ancorando-se nos indicadores de inferência de Fávero et al. (2017), conclui-se que ambos os alunos A e B demonstraram a elaboração de um argumento dedutivo válido ao programar no Scratch. O aluno A relatou de forma detalhada o processo de construção de um

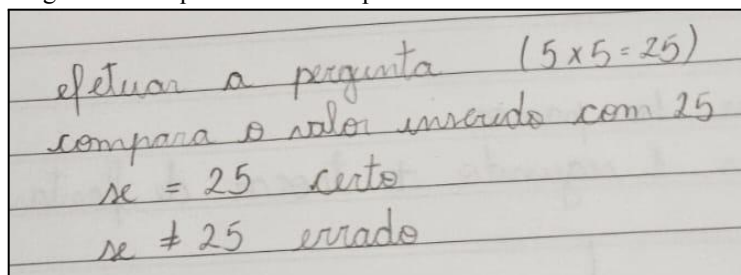
quadrado, destacando a importância de definir os ângulos e os passos necessários, além de justificar suas decisões baseadas na medida do lado do quadrado. Essa abordagem mostra que ele seguiu uma sequência lógica e verificou a precisão de suas decisões ao final. Por sua vez, o aluno B também apresentou um entendimento claro dos conceitos, mencionando a importância dos lados e ângulos iguais. Ele destacou a eficiência de programar dentro de um bloco de repetição, justificando que seu planejamento prévio foi crucial para a rápida percepção e fácil execução da tarefa. Ambas as descrições evidenciam que os alunos utilizaram o raciocínio lógico e a argumentação dedutiva para validar suas premissas e assegurar que suas conclusões fossem verdadeiras e coerentes com os resultados obtidos no programa Scratch.

Na atividade 4, tanto o aluno A quanto o aluno B reconheceram que era necessário definir o resultado da operação para estabelecer as condições adequadas na programação.

sempre que se faz uma pergunta, a gente espera uma resposta e é como se a gente armazenasse ela. Foi assim que eu comecei pensando, porque eu precisava perguntar “quanto é 5×5 ?”, e dependendo da resposta inserida, o programa teria que dizer se estava certo ou errado, então eu precisava comparar o número inserido com o resultado correto da conta que é 25. Se o número inserido for igual a 25 a resposta está correta e se o número inserido for diferente de 25 está errado, para isso eu usei a condicional “se, então, senão”. Assim consegui resolver o problema. (A).

O aluno B destaca que “após fazer a pergunta, é só comparar o valor inserido pelo usuário com 25, que é o resultado de 5×5 . Utilizando a condicional “se, então, senão”, se a resposta for igual a 25 está certo e se for diferente de 25 está errado.” Cita ainda a importância do seu processo de abstração, denominado de “passo a passo” por ele, porque “quando eu leio o problema, eu já penso em como eu vou desenhar ele, assim fica mais fácil de pensar e depois programar.”

Figura 22 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Considerando a análise dos indicadores de inferência de Fávero et al. (2017), observamos que tanto o aluno A quanto o aluno B aplicaram conceitos adequados na atividade 4, mas com abordagens ligeiramente distintas. Ambos os alunos demonstraram a capacidade de formular um argumento dedutivo válido. O aluno A articulou seu raciocínio ao identificar a necessidade de armazenar e comparar respostas. Ele explicou que, para verificar a correção da resposta para a operação 5×5 , utilizou uma comparação entre o número inserido e o resultado esperado (25), empregando a condicional "se, então, senão" para validar a resposta. O aluno A fez uso do argumento dedutivo ao justificar suas decisões, indicando a premissa (a necessidade de comparação), a razão (o valor correto), e a conclusão (a verificação da resposta).

Por outro lado, o aluno B também usou o argumento dedutivo ao abordar o problema de maneira direta e prática. Ele identificou a comparação necessária e aplicou a condicional "se, então, senão" para determinar se a resposta estava correta. O aluno B destacou a importância de seu planejamento prévio, ou "passo a passo", como uma forma de abstração que facilita a programação. Sua abordagem, que incluiu uma reflexão sobre a visualização do problema antes de codificar, também demonstra um uso eficaz do argumento dedutivo ao conectar a premissa (a comparação necessária) com a conclusão (a validação da resposta).

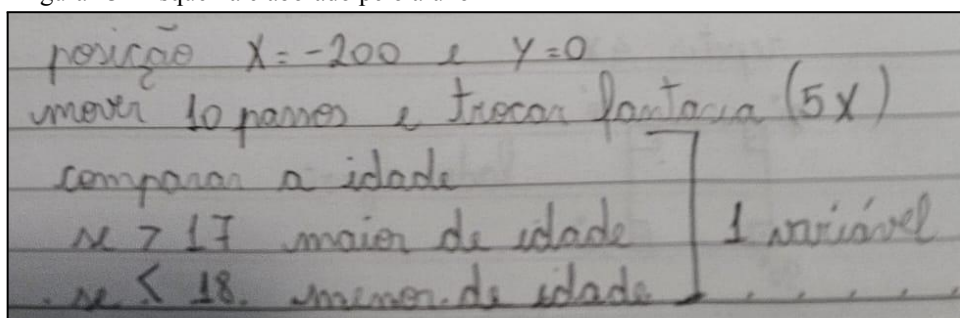
Ambos os alunos, portanto, não apenas aplicaram corretamente os conceitos de programação, mas também utilizaram o argumento dedutivo para estruturar suas soluções e justificar suas decisões. Isso revela um domínio significativo dos conceitos envolvidos e uma habilidade avançada em aplicar lógica e abstração para resolver problemas de programação de maneira eficaz.

A atividade 5 exigiu um tempo considerável para que os alunos compreendessem bem o problema e encontrassem uma solução. Inicialmente, parece que o aluno A teve dificuldades de interpretação, ao contrário do aluno B, que se destacou devido aos seus guias de programação elaborados anteriormente.

no primeiro momento eu não entendi que o personagem precisava se mover para a posição $x = -200$ e $y = 0$, para depois trocar a fantasia, mover 10 passos, esperar um segundo e repetir só essas três coisas por cinco vezes. Depois eu vi que tinha a pergunta da idade, e eu precisava programar para dizer se o valor inserido era de uma pessoa maior ou menor de idade, logo pensei no problema anterior, que eu ia precisar da mesma condicional do "se, então, senão". Defini que qualquer valor digitado menor que 17, diria que a pessoa é menor de idade, logo, todo valor maior que isso significa que a pessoa é maior de idade, mas não funcionou. O esquema estava certo, mas eu não tinha criado uma variável, e posso afirmar que o esquema estava certo, porque quando inclui a variável funcionou (A).

O aluno B relata que “percebi a necessidade de uma variável quando escrevi no papel o passo a passo necessário para resolver o problema, porque eu iria usar só uma condicional de “se, então, senão””, e quando questionado sobre o porquê da variável, coloca que “eu posso dizer que a minha variável idade é o valor inserido como resposta a pergunta, então qualquer resposta maior que 17, representa um indivíduo maior de idade.” Na sequência, foi indagado do porquê utilizar 17 e não 18, então explicou que “quando uma pessoa completa 18 anos, já é maior de idade, então só vai ser menor de idade com 17 anos ou menos”.

Figura 23 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Baseando-se na análise dos indicadores de inferência de Fávero et al. (2017), podemos concluir, que a atividade 5 exigiu um esforço significativo dos alunos para entender o problema e desenvolver uma solução adequada. A abordagem dos alunos A e B revelou diferenças notáveis na forma como enfrentaram o desafio e aplicaram o raciocínio dedutivo.

O aluno A inicialmente encontrou dificuldades na interpretação do problema, o que atrasou seu progresso. Sua abordagem revelou uma compreensão inicial limitada dos requisitos, especialmente na necessidade de definir a posição do personagem e usar uma variável para a idade. Embora tenha reconhecido a importância da condicional “se, então, senão” e tenha identificado a necessidade de correção ao incluir a variável, o aluno A demonstrou uma certa dificuldade em traduzir a lógica teórica para a implementação prática. Isso indica que, embora tenha compreendido a lógica necessária para resolver o problema, houve uma falha na execução inicial, que foi corrigida posteriormente com a inclusão da variável. A dificuldade inicial em interpretar e aplicar a solução é um indicativo da importância da clareza na formulação de problemas. Esse processo de identificação e correção de erros ilustra a importância de habilidades de depuração no pensamento computacional, pois permite que os alunos ajustem suas abordagens e melhorem suas soluções.

Em contraste, o aluno B se destacou pela organização e pelo uso eficaz de guias de programação. A elaboração prévia de um passo a passo permitiu ao aluno B abordar o problema com maior clareza e precisão. Ao reconhecer a necessidade de uma variável desde o início, o aluno B aplicou de forma mais eficiente o raciocínio dedutivo ao identificar que a variável “idade” deveria armazenar o valor inserido e que a condição para determinar maioria deveria ser baseada na comparação com 17 anos. A decisão de usar 17 anos em vez de 18 anos também foi justificada com um raciocínio claro, refletindo uma compreensão mais profunda das condições e do contexto do problema.

Portanto, a análise demonstra que a preparação antecipada e o uso de estratégias de planejamento, como a elaboração de guias e o entendimento antecipado dos requisitos do problema, foram fatores fundamentais para o sucesso na solução. O aluno B evidenciou uma aplicação mais eficaz do raciocínio dedutivo e da abstração, resultando em uma resolução mais eficiente do problema. Já o aluno A, ao enfrentar e corrigir seus erros, exemplifica como o pensamento computacional não apenas envolve a aplicação de conceitos teóricos, mas também a habilidade de identificar problemas e adaptar soluções.

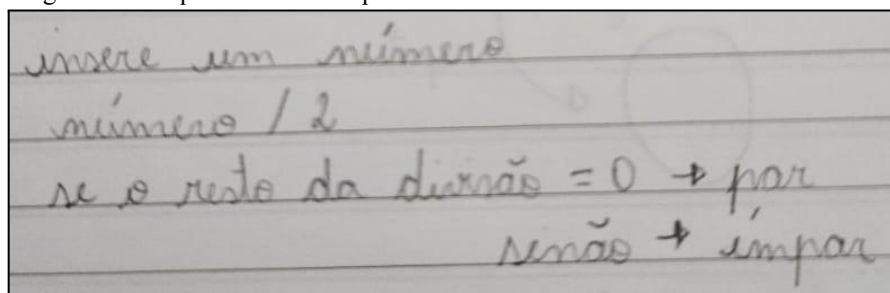
A atividade 6 exigiu uma pesquisa para identificar números pares e ímpares, pois ambos os alunos não se lembravam desse conceito. Após essa etapa, a atividade fluiu bem e os resultados foram conforme o esperado.

depois que a gente pesquisou como identificar um número par ou ímpar, tudo ficou mais fácil e claro. Eu precisava perguntar um número para o usuário, e o meu programa faria a análise dele para identificar se é par ou ímpar. Então com base no problema anterior, vi que precisava de uma variável, para que meu programa pudesse ler o número inserido, avaliar e classificar ele, para isso precisei criar uma condicional de “se, então, senão”, e uma operação que pegasse o número e dividisse por 2, o resto da operação tinha que ser igual a zero e isso caracteriza um número par, caso contrário, o número é ímpar (A).

Já o aluno B relata que “*depois da pesquisa de como identificar um número par ou ímpar, pude criar meu passo a passo e tudo ficou claro para poder programar.*” Quando questionado sobre suas decisões, coloca que

seguindo o modelo anterior, quando eu insiro um número, preciso armazenar ele na variável e validar ele. Para validar eu precisei da fórmula que a gente pesquisou, então eu disse que o resto da divisão de qualquer número por dois precisa ser zero, para o número ser par, senão ele é ímpar (B).

Figura 24 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Com base na análise dos indicadores de inferência de Fávero et al. (2017), pode-se concluir que no processo de resolução da atividade 6, ambos alunos demonstraram capacidade argumentativa dedutiva, justificativas coerentes quanto as suas tomadas de decisões, bem como envolveu a identificação de números pares e ímpares, revelou uma evolução no processo de resolução de problemas por parte dos alunos. Inicialmente, ambos os alunos necessitaram realizar uma pesquisa para relembrar o conceito fundamental de paridade, demonstrando que a recuperação de conhecimento prévio é uma etapa importante para a aplicação de conceitos em novas situações.

O aluno A relatou que, após a pesquisa, a compreensão do problema se tornou mais clara. Ele conseguiu aplicar o conceito de paridade criando uma variável para armazenar o número inserido pelo usuário e utilizando uma condicional para avaliar se o número era par ou ímpar. Sua abordagem envolveu uma operação matemática para verificar o resto da divisão por 2, que é uma técnica fundamental na programação para classificar números. Essa prática de aplicar conhecimento teórico à programação prática exemplifica a importância da abstração e da aplicação de fórmulas matemáticas na resolução de problemas. Mostrou uma capacidade argumentativa ao explicar sua abordagem e decisões. Ele não apenas implementou a lógica necessária para identificar números pares e ímpares, mas também justificou seu processo ao relacionar a criação de variáveis e o uso de condicionais com a necessidade de verificar o resto da divisão por 2. Sua argumentação foi sustentada por uma compreensão clara do conceito e pela aplicação prática da fórmula matemática. Ele conseguiu demonstrar como a aplicação de uma condição específica pode levar a uma solução correta, sublinhando a importância de uma estrutura lógica na programação.

O aluno B, por sua vez, utilizou um método similar, destacando a eficácia de criar um passo a passo para programar de forma estruturada. Ao inserir o número na variável e aplicar a fórmula para verificar a paridade, ele demonstrou um entendimento sólido da necessidade de validar a entrada do usuário. Sua abordagem confirma a importância da pesquisa prévia e

da criação de um plano detalhado para enfrentar desafios da programação. Apresentou uma argumentação que focou na importância do planejamento prévio e na aplicação prática do conceito de paridade. Ele explicou seu raciocínio ao armazenar o número em uma variável e aplicar a fórmula de divisão para verificar a paridade. Sua argumentação também incluiu uma explicação clara do processo de validação dos dados, destacando a importância de uma abordagem sistemática para resolver problemas. A capacidade de argumentar sobre a aplicação da fórmula e a validação correta demonstrou um entendimento aprofundado do conceito e uma habilidade em articular suas decisões.

Ambos os alunos conseguiram superar a dificuldade inicial e aplicar o conceito de forma adequada, mas suas experiências evidenciam a importância da pesquisa, da aplicação prática do conhecimento e da elaboração de um planejamento eficaz. Enquanto o aluno A focou na aplicação direta do conceito com o uso de condicionais e operações matemáticas, o aluno B destacou a importância do planejamento e da validação correta dos dados. Em ambos os casos, a capacidade argumentativa foi relevante para a resolução eficaz do problema. O aluno A e o aluno B não apenas resolveram o problema proposto, mas também foram capazes de explicar e justificar suas abordagens de maneira detalhada e lógica. Isso evidencia que uma argumentação sólida é fundamental para a compreensão e aplicação dos conceitos de programação, permitindo que os alunos não apenas resolvam problemas, mas também expliquem suas soluções de forma clara e convincente.

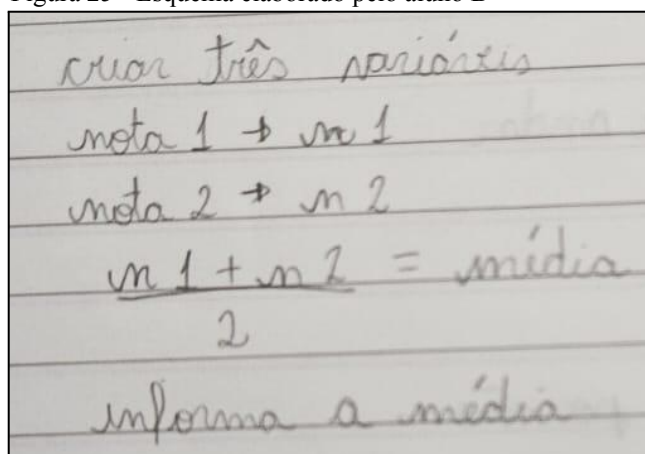
A atividade 7 foi relacionada à realidade escolar tanto pelo aluno A quanto pelo aluno B. Assim que leram o problema, rapidamente concluíram que, para calcular a média ao inserir duas notas, precisariam de variáveis.

precisei pensar quantas variáveis eu precisava, porque imaginei que duas seriam suficientes, uma para a nota 1 e outra para a nota 2. Ao iniciar a programação percebi que precisava de uma terceira variável para representar a média, porque eu precisava calcular a média daquelas notas inseridas e já armazenadas, para depois dizer para o usuário o resultado. Então fiz a primeira pergunta e armazenei a primeira resposta na variável nota 1, fiz a segunda pergunta e armazenei a segunda resposta na variável nota 2. Depois eu criei uma operação para definir a variável média, que é a soma das duas notas dividido por 2, porque eu tenho duas notas. O programa está pronto para dizer a média das notas. (A).

O aluno B seguiu argumentando baseado em seus esquemas que norteiam as programações, e colocou que,

quando eu tenho duas notas e quero obter a média, eu preciso somar as duas e dividir por dois e foi essa operação que eu fiz para definir a variável média. A primeira nota inserida eu armazenei na variável n1 e a segunda nota inserida eu armazenei na variável n2. Assim, meu programa estava pronto para dizer a média das notas. (B).

Figura 25 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Com base na análise dos indicadores de inferência de Fávero et al. (2017), pode-se concluir que a capacidade argumentativa e a aplicação de conceitos dedutivos foram fundamentais para que os alunos A e B resolvessem a atividade 7. A atividade, que envolvia calcular a média de duas notas inseridas pelo usuário, foi rapidamente relacionada à realidade escolar por ambos os alunos, demonstrando sua capacidade de abstração e aplicação prática dos conceitos.

O aluno A demonstrou uma capacidade argumentativa sólida ao explicar detalhadamente seu processo de raciocínio. Ele identificou a necessidade de criar três variáveis: uma para cada nota e uma terceira para armazenar a média. Sua explicação incluiu a sequência lógica de perguntas e armazenamento das respostas, seguida pela operação necessária para calcular a média. A clareza em sua argumentação e a justificação de cada passo evidenciam uma compreensão profunda do problema e a capacidade de estruturar uma solução eficaz. Ele concluiu que, ao armazenar as notas e realizar a operação de soma e divisão, o programa estaria pronto para fornecer a média ao usuário, demonstrando uma abordagem metódica e dedutiva.

O aluno B, por sua vez, seguiu uma abordagem semelhante, mas sua argumentação foi enriquecida por esquemas que nortearam sua programação. Ele explicou que, ao ter duas notas, a operação necessária para calcular a média era somar as duas notas e dividir por dois. Ele destacou a importância de armazenar as notas em variáveis específicas (n1 e n2) antes de realizar a operação. A utilização de esquemas para planejar sua programação demonstra uma habilidade organizacional e a capacidade de visualizar a solução antes de implementá-la,

facilitando a compreensão e execução do problema. Sua argumentação clara e estruturada confirma a eficácia de seu método e a validade de sua solução.

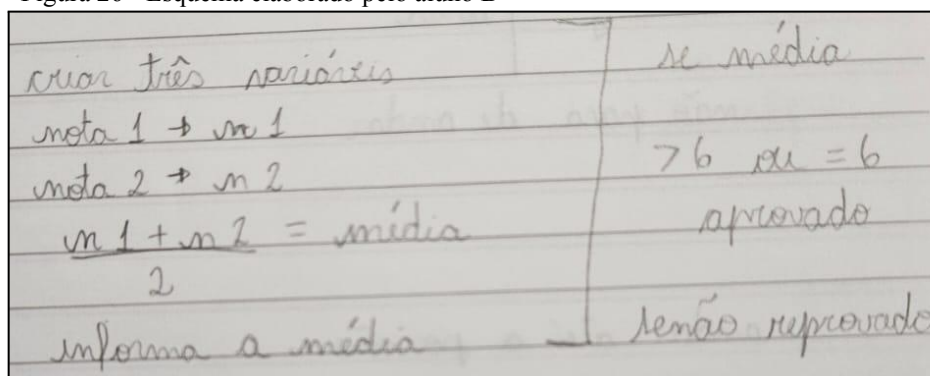
Ambos os alunos, ao relacionarem o problema à sua realidade escolar e utilizarem a abstração e a argumentação dedutiva, foram capazes de resolver a atividade de maneira eficaz. Suas explicações detalhadas e justificativas lógicas demonstram não apenas a compreensão do problema, mas também a habilidade de comunicar claramente suas soluções. Essa capacidade argumentativa é base para o desenvolvimento do pensamento computacional, permitindo que os alunos não apenas resolvam problemas, mas também expliquem e defendam suas soluções de forma convincente.

Ao apresentar a atividade 8, os alunos A e B perceberam que podiam utilizar a estrutura do problema anterior e adicionar uma condicional que determinasse que, se a média das duas notas fosse maior ou igual a 6, o aluno estaria aprovado; caso contrário, estaria reprovado.

eu só preciso pegar o problema anterior e adicionar uma condicional “se, então, senão” e indicar a operação. Então eu usei um bloco “ou” e disse que se a média fosse maior ou igual a 6 o aluno estaria aprovado, senão estaria reprovado. (A).

se eu continuar do programa de antes, posso usar a condicional “se, então, senão”, e dizer que se a média for maior ou igual a 6, o aluno está aprovado, senão está reprovado. Precisei inserir três blocos de operadores, um para dizer “ou”, outro para representar a média maior que 6 e outro para representar a média igual a 6. [...] Cheguei a essa conclusão de condicional quando escrevi, porque quando só olhei e pensei, não tive ideia de como fazer, mas logo que escrevi parece que vi os blocos verdes de operadores na minha frente. (B).

Figura 26 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Com base na análise dos indicadores de inferência de Fávero et al. (2017), pode-se concluir que a atividade 8 reforçou a capacidade argumentativa e dedutiva dos alunos A e B, permitindo-lhes aplicar conceitos previamente aprendidos para resolver um problema

adicional. Ao perceberem que podiam utilizar a estrutura do problema anterior e adicionar uma condicional para determinar a aprovação ou reprovação com base na média das notas, ambos os alunos demonstraram uma clara compreensão dos conceitos de programação e lógica.

O aluno A justificou sua abordagem de forma concisa e direta. Ele explicou que bastava pegar o problema anterior, que já calculava a média das notas, e adicionar uma condicional “se, então, senão” para verificar se a média era maior ou igual a 6. Sua argumentação simples e eficaz indicou a operação necessária para determinar a aprovação ou reprovação, evidenciando uma aplicação prática e lógica dos conceitos aprendidos. Utilizou um argumento dedutivo claro ao identificar que bastava reutilizar o cálculo da média das notas e adicionar uma condicional para determinar a aprovação ou reprovação. Ele justificou suas decisões de forma lógica, indicando que se a média fosse maior ou igual a 6, o aluno estaria aprovado; caso contrário, estaria reprovado. Esse raciocínio dedutivo mostra que ele partiu de premissas verdadeiras (o cálculo da média e a comparação com o valor 6) para chegar a uma conclusão lógica e válida.

O aluno B, por sua vez, destacou a importância de visualizar e escrever os passos antes de programar. Ele mencionou que ao continuar do programa anterior, poderia usar a mesma condicional “se, então, senão” e especificar que, se a média fosse maior ou igual a 6, o aluno estaria aprovado; caso contrário, estaria reprovado. Sua argumentação detalhada sobre a inclusão dos operadores “ou” e as comparações com o valor 6 demonstra uma abordagem dedutiva robusta. Ele partiu das premissas conhecidas (a necessidade de calcular a média e comparar com 6 e concluiu logicamente que a estrutura condicional resolveria o problema, assegurando que a solução fosse correta e eficiente. Sua argumentação incluiu detalhes sobre a inserção de blocos de operadores para representar as condições necessárias. Ele enfatizou que a clareza de seu raciocínio surgiu ao escrever os passos, o que lhe permitiu visualizar os blocos de operadores necessários, mostrando a importância da preparação e organização no processo de programação.

Ambos os alunos, ao reutilizarem a estrutura do problema anterior e adicionarem uma condicional, demonstraram não apenas a capacidade de resolver problemas de maneira eficiente, mas também a habilidade de justificar e argumentar suas soluções com clareza. Essa prática de argumentação dedutiva e a capacidade de adaptar soluções anteriores a novos problemas são essenciais para o desenvolvimento do pensamento computacional. A abordagem metódica e detalhada de ambos os alunos reflete um entendimento profundo dos

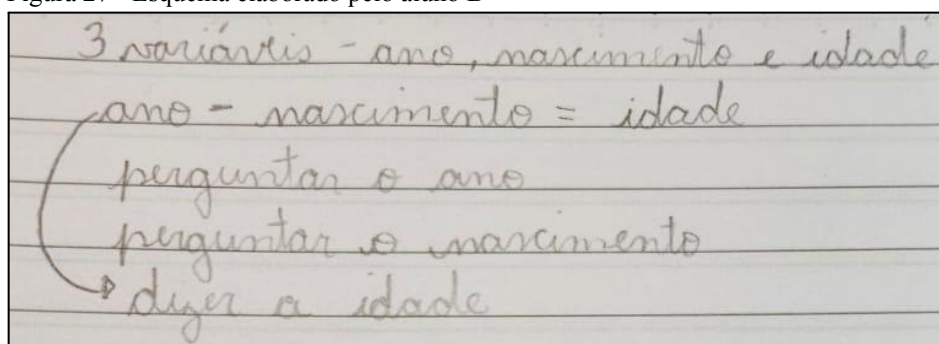
conceitos e uma aplicação prática eficaz, evidenciando o sucesso de suas estratégias de resolução de problemas.

Quando os alunos A e B se depararam com a atividade 9, rapidamente a relacionaram com as anteriores, dizendo que precisariam de três variáveis. Desta vez, estariam calculando uma idade através do ano atual e do ano de nascimento, mas seguindo o mesmo esquema lógico do problema de cálculo de média.

primeiro pensei no cálculo que eu preciso para saber a idade, que é o ano atual menos o ano do nascimento, então o resultado é a idade da pessoa. Usei uma variável para o ano do nascimento, outras variável para o ano atual e mais uma variável para poder calcular a idade. Usei um bloco de operador para fazer o cálculo e o meu personagem passou a dizer a idade da pessoa, então o problema estava resolvido. (A).

primeiro perguntava o ano do nascimento e armazenei o dado na variável que eu chamei de nascimento, depois perguntava o ano atual e armazenei o dado na variável que eu chamei de ano. A variável que eu chamei idade ficou definida como um cálculo, que seria o ano atual menos o ano de nascimento e então o programa passou a dizer a idade do usuário. (B).

Figura 27 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Considerando a análise dos indicadores de inferência de Fávero et al. (2017), pode-se concluir que na atividade 9, tanto o aluno A quanto o aluno B demonstraram um entendimento sólido do uso de variáveis e operadores para resolver problemas matemáticos através da programação. Ambos aplicaram um argumento dedutivo claro e eficaz, partindo de premissas conhecidas para chegar a uma solução lógica.

O aluno A seguiu um raciocínio dedutivo simples e direto, identificando que a idade poderia ser calculada subtraindo o ano de nascimento do ano atual. Ele definiu três variáveis e utilizou um bloco de operador para realizar o cálculo, demonstrando uma abordagem lógica e estruturada. A conclusão de que o problema estava resolvido ao obter a idade correta reflete uma aplicação bem sucedida de um argumento dedutivo.

O aluno B, por sua vez, também aplicou um argumento dedutivo ao definir variáveis para o ano de nascimento, o ano atual e a idade. Ele seguiu uma sequência lógica de perguntas e armazenamento de dados, utilizando um cálculo simples para definir a variável idade. A representação da figura 27, a explicação detalhada de seu raciocínio e a identificação correta das variáveis necessárias reforçam a eficácia de seu argumento dedutivo.

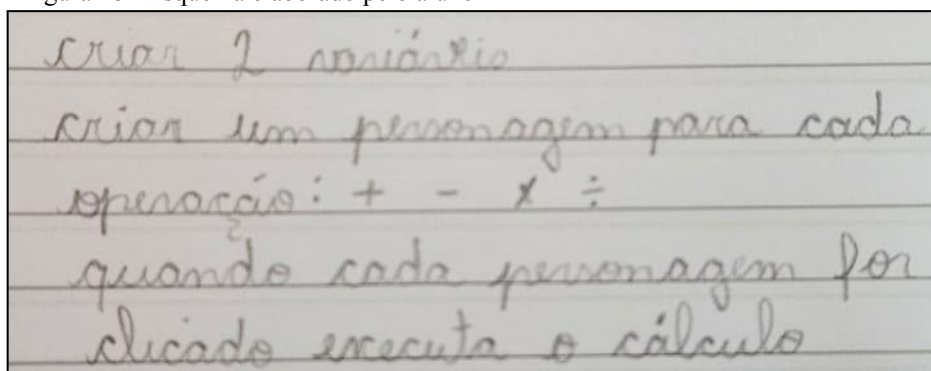
A atividade 9 também evidenciou a capacidade argumentativa dos alunos A e B. Ambos foram capazes de explicar detalhadamente o processo seguido para resolver o problema, justificando suas escolhas e demonstrando a lógica por trás de cada passo. A clareza na argumentação e a habilidade de relacionar a atividade atual com problemas anteriores mostram um progresso significativo na capacidade de construir e apresentar argumentos dedutivos válidos e coerentes, o que é base para o desenvolvimento do pensamento crítico e computacional.

A atividade 10 evidenciou a compreensão lógica dos alunos A e B, que dialogaram entre si e concluíram que a alternativa mais simples, rápida e efetiva seria criar um personagem para cada operação: adição, subtração, multiplicação e divisão. Dessa forma, precisariam de apenas duas variáveis. Os caminhos e decisões foram justificados da seguinte maneira,

depois de conversar com B, vimos que podemos criar duas variáveis, uma para cada valor inserido. Depois criamos um personagem pra conta de mais, outro pra conta de menos, outro pra conta de vezes e outro pra conta de dividir. Assim, quando o usuário clica em cima desses personagens ele vai falar o resultado da operação. (A).

quando conversei com A, tivemos a ideia de programar através de personagens, assim, quando o usuário clicar em cima da operação desejada, o programa vai executar esta operação e vai dizer o resultado. É só usar o bloco de operação no personagem correspondente e o programa funciona perfeitamente, como se fosse uma calculadora. (B).

Figura 28 - Esquema elaborado pelo aluno B



Fonte: Registro fotográfico da autora, 2024.

Considerando a análise dos indicadores de inferência de Fávero et al. (2017), pode-se concluir que na atividade 10, os alunos A e B aplicaram um argumento dedutivo claro e estruturado. Ao identificar que a criação de personagens para cada operação permitiria uma solução eficiente e intuitiva, eles demonstraram a habilidade de decompor o problema em componentes menores e manejáveis, facilitando a implementação da solução. A justificativa das decisões tomadas, desde a criação das variáveis até a associação de operações com personagens, evidenciou um raciocínio lógico coerente.

A abordagem adotada pelos alunos na atividade 10 reflete um entendimento profundo do pensamento computacional. Eles demonstraram a capacidade de abstrair o problema, identificar as operações necessárias e criar uma solução modular que facilita a interação do usuário. A decisão de usar personagens para representar operações é um exemplo de como a abstração pode simplificar a resolução de problemas complexos, tornando o processo mais acessível e direto.

A atividade 10 também destacou a capacidade argumentativa dos alunos A e B. Através do diálogo e da colaboração, eles foram capazes de justificar suas escolhas e explicar detalhadamente o raciocínio por trás de cada passo do processo. A clareza na comunicação e a habilidade de construir argumentos dedutivos sólidos mostram um progresso significativo na capacidade de apresentar e defender suas ideias, o que é essencial para o desenvolvimento do pensamento crítico e computacional.

Em resumo, a atividade 10 demonstrou que os alunos A e B não apenas compreenderam os conceitos matemáticos e computacionais necessários para resolver o problema, mas também foram capazes de aplicar esses conceitos de maneira lógica e estruturada, utilizando um argumento dedutivo válido para chegar a uma solução eficiente e funcional.

A observação participativa revelou-se fundamental para a compreensão da capacidade de argumentar no processo de programar computadores, destacando como a prática da programação contribui para o desenvolvimento de habilidades argumentativas. Ao se envolverem ativamente nas atividades de programação, os alunos foram frequentemente confrontados com desafios que exigiam a construção e justificação de argumentos lógicos para suas soluções. Esse processo não só demandou que os alunos formassem argumentos sólidos baseados em premissas e inferências lógicas, mas também proporcionou um ambiente rico para o desenvolvimento dessas habilidades.

A análise dos dados coletados mostrou que, ao enfrentar desafios de programação, os alunos foram incentivados a articular claramente suas justificativas para as escolhas feitas em

seus códigos. Eles tiveram que explicar suas abordagens, defendendo suas soluções perante colegas e instrutores, o que fortaleceu significativamente sua capacidade de argumentação. Essa prática constante de justificar suas decisões promoveu uma maior conscientização sobre a importância de bases sólidas para suas argumentações, levando-os a desenvolver um pensamento mais crítico e estruturado.

Além disso, a necessidade de depurar e corrigir erros durante a programação exigiu dos alunos uma abordagem sistemática e crítica. Eles precisaram identificar falhas lógicas em seus códigos, rastrear a origem dos problemas e formular soluções alternativas. Esse processo de identificação e correção de erros reforçou o raciocínio lógico, uma vez que os alunos precisaram aplicar princípios de lógica dedutiva para resolver problemas complexos. A capacidade de analisar sistematicamente um problema e desenvolver uma solução lógica é uma habilidade essencial tanto na programação quanto na construção de argumentos válidos.

Esse ambiente de resolução de problemas não apenas melhorou a habilidade dos alunos em construir argumentos coerentes, mas também reforçou seu raciocínio lógico ao exigir uma abordagem sistemática e crítica para identificar e corrigir erros. A prática de depuração, em particular, mostrou-se extremamente benéfica para o desenvolvimento de habilidades de pensamento crítico e analítico. Os alunos aprenderam a abordar problemas de maneira metódica, o que é crucial para a construção de argumentos dedutivos bem fundamentados.

A observação participativa também proporcionou percepções valiosas sobre como diferentes aspectos da programação influenciam o desenvolvimento da argumentação. Por exemplo, a decomposição de problemas complexos em partes menores e mais gerenciáveis, uma habilidade central no pensamento computacional, facilitou a estruturação de argumentos dedutivos. Os alunos puderam aplicar essas técnicas de decomposição para organizar suas premissas de maneira lógica e sequencial, o que é essencial para a construção de argumentos sólidos.

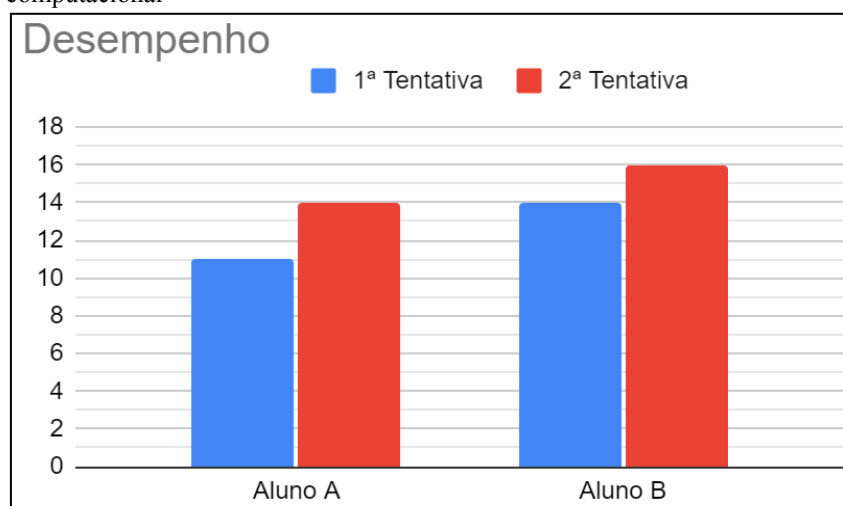
Em resumo, a observação participativa confirmou que a programação não só facilita a prática da argumentação dedutiva, mas também contribui significativamente para o aprimoramento do raciocínio lógico. A integração de atividades de programação no currículo escolar oferece uma base sólida para o desenvolvimento dessas habilidades essenciais, preparando os alunos para enfrentar desafios complexos com uma argumentação estruturada e lógica. Esses achados oferecem uma base sólida para futuras pesquisas, explorando como diferentes aspectos da programação influenciam o desenvolvimento da argumentação e como essas práticas podem ser aprimoradas para maximizar os benefícios educacionais.

A importância de integrar práticas de programação no ensino é destacada pela capacidade dessas atividades de promover o desenvolvimento de habilidades cognitivas cruciais. Ao envolver os alunos em processos que exigem tanto a criação quanto a justificção de soluções, a programação proporciona um contexto rico para o desenvolvimento de habilidades argumentativas e de raciocínio lógico. Essa abordagem prepara os alunos não apenas para o sucesso acadêmico, mas também para uma participação crítica e eficaz na sociedade, onde a capacidade de argumentar de forma lógica e coerente é cada vez mais valorizada.

6.2 Teste de pensamento computacional

Considerando o teste de pensamento computacional com 16 questões, que foi respondido no início dos encontros pelos alunos A e B e novamente ao final. O aluno A acertou 11 questões no início das atividades e, ao responder o teste novamente, acertou 14 questões. Já o aluno B acertou 14 questões no primeiro contato com o teste e, depois, acertou 16 questões, conforme apresentado no gráfico a seguir.

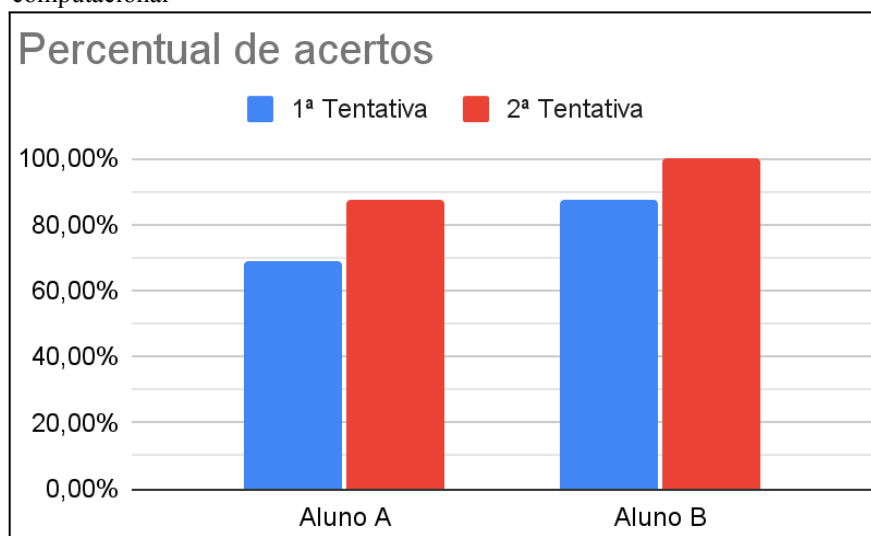
Figura 29 - Desempenho dos alunos ao responder o teste de pensamento computacional



Fonte: Autora, 2024.

Na primeira tentativa, o aluno A acertou cerca de 68,75% do teste e, na segunda tentativa, melhorou para cerca de 87,5%. O aluno B acertou aproximadamente 87,5% na primeira tentativa e, na segunda tentativa, acertou todas as questões, alcançando 100%, conforme demonstrado no gráfico a seguir.

Figura 30 - Percentual de acertos dos alunos ao responder o teste de pensamento computacional



Fonte: Autora, 2024.

Ainda sobre os acertos, realizamos um levantamento detalhado questão por questão para identificar os erros de cada aluno. Verificamos que o aluno A errou as questões 3, 9, 11, 12 e 13 na primeira tentativa, e voltou a errar as questões 9 e 11 na segunda tentativa. Por outro lado, o aluno B errou as questões 6 e 12 na primeira tentativa, mas acertou todas as questões na segunda tentativa, conforme quadro a seguir.

Com base na observação participativa, que incluiu a análise da capacidade dos alunos em resolver problemas, construir e revisar códigos, acredita-se que a repetição de erros pode ser um sinal de que o aluno não compreendeu completamente as instruções de uma questão ou que aplicou uma estratégia de resolução inadequada. A falta de clareza nas instruções pode levar a uma interpretação incorreta da questão, resultando em respostas erradas. Além disso, se o aluno usou uma abordagem inadequada ou incorreta na primeira tentativa, pode não ter ajustado sua estratégia na segunda tentativa, continuando a cometer os mesmos erros.

Quadro 33 - Erros e acertos de cada aluno no teste de pensamento computacional

	ALUNO A		ALUNO B	
QUESTÃO	1ª TENTATIVA	2ª TENTATIVA	1ª TENTATIVA	2ª TENTATIVA
1	v	v	v	v
2	v	v	v	v
3	x	v	v	v

4	v	v	v	v
5	v	v	v	v
6	v	v	x	v
7	v	v	v	v
8	v	v	v	v
9	x	x	v	v
10	v	v	v	v
11	x	x	v	v
12	x	v	x	v
13	x	v	v	v
14	v	v	v	v
15	v	v	v	v
16	v	v	v	v

Fonte: Autora, 2024.

Para analisar esses dados, consideramos uma abordagem estatística para entender se houve uma melhoria significativa no desempenho dos alunos na segunda tentativa em comparação com a primeira. A análise envolveu o cálculo da diferença de desempenho, que mede a mudança no número de questões corretas entre as duas tentativas.

O cálculo da diferença de desempenho é um passo essencial para identificar a eficácia das atividades realizadas entre as tentativas do teste. Ao calcular a diferença entre o número de questões corretas na segunda tentativa e na primeira tentativa para cada aluno, obtemos uma medida clara da mudança no desempenho, conforme o quadro a seguir.

Quadro 34 - Cálculos estatísticos

ALUNO A	ALUNO B
Primeira tentativa: 11 questões corretas Segunda tentativa: 14 questões corretas	Primeira tentativa: 14 questões corretas Segunda tentativa: 16 questões corretas
Diferença: $14 - 11 = 3$	Diferença: $16 - 14 = 2$
Percentual de melhoria: $\left(\frac{14-11}{16}\right) \times 100\% = \frac{3}{16} \times 100\% = 18,75\%$	Percentual de melhoria: $\left(\frac{16-14}{16}\right) \times 100\% = \frac{2}{16} \times 100\% = 12,5\%$

Fonte: Autora, 2024.

Os resultados indicam uma melhoria no desempenho dos alunos entre as duas tentativas. Essa melhoria pode ser atribuída a vários fatores, como o aumento da familiaridade com o formato do teste, maior confiança ao responder às questões, e o impacto positivo das atividades educacionais realizadas entre as tentativas, visto que em termos percentuais houve uma melhora de 18,75% no desempenho do aluno A. Já o aluno B melhorou 12,5% do desempenho, mas vale ressaltar que atingiu os acertos máximos na segunda tentativa.

Embora os dados de apenas dois alunos forneçam uma visão inicial, a análise estatística robusta geralmente requer um tamanho de amostra maior para detectar melhorias significativas com maior confiança. Uma amostra maior também permitiria a aplicação de análises estatísticas mais sofisticadas, que poderiam confirmar a significância dos resultados observados.

Além da análise quantitativa, é importante considerar as percepções dos alunos sobre as atividades realizadas. A observação participativa forneceu dados qualitativos suficientes para afirmar que os alunos estavam desenvolvendo o pensamento computacional, o que pode ter influenciado seu desempenho no teste de pensamento computacional.

Em resumo, a análise dos dados dos alunos A e B mostrou uma melhoria no desempenho nas questões do teste de pensamento computacional após a realização de atividades educativas. No entanto, para confirmar a significância estatística dessa melhoria, estudos futuros com amostras maiores são recomendados. Integrar análises quantitativas e qualitativas pode fornecer uma visão mais completa e ajudar a otimizar métodos de ensino que promovam o desenvolvimento eficaz do pensamento computacional.

6.3 Questionário de argumento dedutivo

A análise de dados qualitativos é uma ferramenta importante para a compreensão profunda das habilidades argumentativas dos alunos, especialmente quando se trata de argumentação dedutiva. Nesta pesquisa, a análise qualitativa é fundamentada nos conceitos propostos por Fávero et al. (2017), Weston (2009) e Walton (2012), que fornecem um quadro teórico robusto para a avaliação da construção e análise de argumentos lógicos.

Fávero et al. (2017) contribuem com uma abordagem prática para a identificação de indicadores de inferência, diferenciando claramente entre premissas e conclusões, e fornecendo ferramentas para a análise lógica dos argumentos. Weston (2009) aborda a importância da clareza e da estrutura lógica na argumentação, destacando a necessidade de uma análise detalhada das premissas e conclusões para garantir a validade e a coerência dos

argumentos. Walton (2012), por sua vez, explora as diferentes formas de argumentação e os mecanismos que sustentam a persuasão e a eficácia argumentativa. Com base nesses fundamentos teóricos, nosso estudo visa avaliar como os alunos aplicam princípios de argumentação dedutiva em contextos diversos, considerando o seguinte esquema.

Quadro 35 - Esquema das quatro formas válidas de argumento dedutivo

<i>Modus ponens (MP)</i> Se [proposição A], então [proposição B]. [Proposição A]. Portanto, [proposição B].	<i>Modus tollens (MT)</i> Se [proposição P], então [proposição Q]. [Proposição Q] é falso. Portanto, [proposição P] é falso
<i>Silogismo hipotético (SH)</i> Se [proposição R], então [proposição S]. Se [proposição S], então [proposição T]. Portanto, se [proposição R], então [proposição T]	<i>Silogismo disjuntivo (SD)</i> Ou [proposição M] ou [proposição N]. [Proposição M] é falso. Portanto, [proposição N].

Fonte: Construção da autora com base na teoria, 2023.

O questionário desenvolvido para esta pesquisa foi projetado para desafiar os alunos a utilizar e refletir sobre princípios de argumentação dedutiva, conforme descrito por Weston e Walton. As perguntas incluíram tanto questões abertas quanto fechadas, focadas em avaliar a capacidade dos alunos de identificar e formular premissas e conclusões, além de analisar a validade e a estrutura lógica dos argumentos apresentados, conforme os indicadores de inferência propostos por Fávero et al. (2017).

A análise qualitativa das respostas dos alunos segue os princípios delineados por Weston (2009) e Walton (2012), complementados pela abordagem prática de Fávero et al. (2017). Através de uma interpretação detalhada, buscamos identificar padrões de pensamento, estratégias argumentativas e áreas de dificuldade. Esta análise proporciona uma visão aprofundada das competências argumentativas dos alunos.

Ao analisar a primeira etapa do questionário de argumentação dedutiva, observou-se que o aluno A justificou corretamente as quatro perguntas propostas, utilizando duas vezes a estrutura de MP, uma vez a estrutura de MT e uma vez a estrutura de SD. O aluno B justificou apenas três perguntas de forma válida, empregando três vezes a estrutura de MP.

- (1) *Se o céu está claro portanto não vai chover. O céu estando claro significa que não vai chover.*
- (2) *O alarme dispara quando há intrusos. Se o alarme não disparou é porque não há intrusos.*
- (3) *Se chover amanhã levarei meu guarda-chuva. Amanhã choverá, então levarei o meu guarda-chuva.*
- (4) *Ou estudaremos ou falharemos na prova. Não estudamos então falharemos na prova. (A).*

- (1) *Concordo, porque se o céu está claro então não vai chover.*
- (2) *Concordo, pois o alarme disparou porque há intrusos, se não há intrusos não iria tocar.*

(3) *Concordo, porque amanhã irá chover então levarei o meu guarda-chuva.*

(4) *Eu concordo que se não estudarmos falharemos.*(B).

Ao analisar os argumentos do aluno A, percebe-se que o (1) é do tipo MP válido, onde a conclusão é uma consequência direta das premissas. (2) É do tipo MT válido e aplica a estrutura corretamente. (3) É do tipo MP válido e aplica a estrutura corretamente. (4) É do tipo SD válido e aplica a estrutura corretamente.

Ao analisar os argumentos do aluno B, percebe-se que o (1) é do tipo MP válido e aplica a estrutura corretamente. (2) É do tipo MP válido e aplica a estrutura corretamente. (3) É do tipo MP válido e aplica a estrutura corretamente. (4) Concorda com a conclusão do argumento válido, mas não fornece uma justificativa lógica adicional, ou seja, não apresenta uma argumentação lógica clara ou não segue uma estrutura de inferência específica.

Ao analisar a segunda etapa do questionário de argumentação dedutiva, observou-se que o aluno A justificou corretamente as quatro perguntas propostas, utilizando uma vez a estrutura de MP, uma vez a estrutura de MT, uma vez a estrutura de SH e uma vez a estrutura de SD. O aluno B também justificou as quatro perguntas de forma válida, empregando duas vezes a estrutura de MP, uma vez a estrutura de SH e uma vez a estrutura de SD.

(1) *Porque se estudarmos, então vamos obter boas notas e teremos a aprovação de ano.*

(2) *Se ela estivesse carregada, ela ligaria por ter energia o suficiente para ligar, mas ela não está carregada então ela não irá ligar.*

(3) *Se não estudarmos tiraremos notas ruins e reprovaremos de ano, e se estudarmos teremos notas boas e passaremos de ano.*

(4) *Ele vai ao cinema ou fica em casa; se ele não for ao cinema irá ficar em casa.*(A)

(1) *Se não estudarmos tiraremos notas ruins e reprovaremos de ano, e se estudarmos teremos notas boas e passaremos de ano*

(2) *Se o celular estiver carregado funcionará.*

(3) *Concordo, porque se estudarmos juntos entenderemos mais sobre o assunto.*

(4) *Concordo, pois ele não vai ao cinema então ele ficará em casa.* (B)

Ao analisar os argumentos do aluno A, percebe-se que o (1) é do tipo MP válido e aplica a estrutura corretamente. (2) É do tipo MT válido e aplica a estrutura corretamente. (3) É do tipo SH válido e aplica a estrutura corretamente. (4) É do tipo SD válido e aplica a estrutura corretamente.

Ao analisar os argumentos do aluno B, percebe-se que (1) é do tipo SH válido e aplica a estrutura corretamente. (2) É do tipo MP válido e aplica a estrutura corretamente, porém está implícito no argumento. (3) É do tipo MP válido e aplica a estrutura corretamente. (4) É do tipo SD válido e aplica a estrutura corretamente.

Ao analisar a terceira etapa do questionário de argumentação dedutiva, observou-se que tanto o aluno A, quanto o aluno B justificaram corretamente as quatro perguntas propostas, utilizando duas vezes a estrutura de MP e duas vezes a estrutura de MT.

- (1) *Concordo, ele disse que acordou cedo, e que não vai chegar atrasado na reunião.*
- (2) *A escola seria fechada caso houvesse neve, mas neste caso não teve neve, então a escola estará aberta.*
- (3) *Concordo, ele guardou dinheiro então ele vai viajar nas férias.*
- (4) *Concordo, pois eles não acordaram cedo, então perderam o voo. (A)*

- (1) *Concordo, pois se ele não acordasse ele não iria ao compromisso, assim ele acordou cedo então não perdeu seu compromisso.*
- (2) *Concordo, porque se houvesse neve não teria aula pois a escola estará fechada, então não nevou vai ter aula pois a escola estará aberta.*
- (3) *Concordo, porque ele economizou então ele poderá viajar, mas se ele não tivesse economizado ele não poderia viajar.*
- (4) *Se eles tivessem acordado cedo, não perderiam o voo, mas nesse caso, eles acordaram tarde e perderam o voo. (B)*

Ao analisar os argumentos dos alunos A e B, percebe-se que os argumentos (1) e (3) de ambos são do tipo MP válido e aplicam a estrutura corretamente. Já os argumentos (2) e (4) de ambos são do tipo MT válido e também aplicam a estrutura corretamente.

Ao analisar a quarta etapa do questionário de argumentação dedutiva, observou-se que o aluno A justificou corretamente as quatro perguntas propostas, utilizando duas vezes a estrutura de MP e duas vezes a estrutura de MT. O aluno B também justificou as quatro perguntas de forma válida, empregando uma vez a estrutura de MP e três vezes a estrutura de MT.

- (1) *Se a água não chegasse a zero graus a água não ficaria gelada e ela não ficaria congelada, mas chegou a zero graus então a água ficou gelada e ela congelará.*
- (2) *Se a Maria tivesse ganhado o concurso estaríamos sabendo, portanto se não estamos sabendo é porque a Maria não ganhou.*
- (3) *Se ele terminasse tarde ele não sairia cedo pois não terminou cedo, mas ele terminou cedo então ele pode sair mais cedo.*
- (4) *Concordo, ela não pagou a multa então ela terá pontos na carteira. (A)*

- (1) *Pois se a temperatura estiver abaixo de zero ela irá congelar e como ela ficou abaixo de zero a água congelou.*
- (2) *Concordo, porque se a Maria ganhasse eles iriam saber, portanto não receberam a notícia, então ela não ganhou.*
- (3) *Concordo, porque se ele não tivesse acabado o trabalho mais cedo, não iria sair mais cedo, portanto ele acabou mais cedo, então ele saiu mais cedo.*
- (4) *Pois ela não pagou e se ela não pagasse ela ganharia pontos na carteira. (B)*

Ao analisar os argumentos do aluno A, percebe-se que (1) e (3) são do tipo MP válido e aplicam a estrutura corretamente. (2) e (4) são do tipo MT válido e também aplicam a

estrutura corretamente. Já o aluno B utiliza-se de MP válido aplicando a estrutura corretamente em (1). Os argumentos (2), (3) e (4) são do tipo MT válido e também aplicam a estrutura corretamente.

Ao analisar a quinta e última etapa do questionário de argumentação dedutiva, observou-se que tanto o aluno A, quanto o aluno B justificaram corretamente as quatro perguntas propostas, utilizando duas vezes a estrutura de MP, uma vez a estrutura de MT e uma vez a estrutura mista de MP e MT.

- (1) *Se não alimentarmos o gato ele ficará com fome, mas eles trataram ele então o gato ficou satisfeito.*
 (2) *Se o avião se atrasasse eles perderiam o voo mas o avião chegou então iremos voar e aproveitaremos o voo.*
 (3) *Se a equipe não ganhasse, eles não iriam ter comemoração, portanto eles ganharam, então terá comemoração.*
 (4) *O projeto não foi concluído hoje, portanto foi adiado. Se o projeto tivesse sido acabado não teria sido adiado. (A)*

- (1) *Eu concordo que o gato está satisfeito, pois nas premissas está dizendo que alimentamos o gato, então ele está satisfeito.*
 (2) *Concordo, porque se ele (ela) tivesse se atrasado não teriam conexão, mas eles chegaram então não perderam a conexão.*
 (3) *A equipe ganhou o jogo, então teve comemoração. Se eles não tivessem ganhado não teria comemoração.*
 (4) *Concordo, se não tivessem adiado eles teriam acabado, mas o trabalho foi adiado então eles não tinham acabado. (B)*

Ao analisar os argumentos dos alunos A e B, percebe-se que os argumentos (1) e (2) de ambos são do tipo MP válido e aplicam a estrutura corretamente. Já os argumentos (3) de ambos é do tipo misto, onde aborda MP e MT de forma válida, aplicando as duas estruturas no mesmo argumento. (4) de ambos são do tipo MT válido e também aplicam a estrutura corretamente.

A análise dos argumentos dos alunos A e B revelou um entendimento sólido das estruturas lógicas do argumento dedutivo. Ambos os alunos demonstraram a capacidade de aplicar essas estruturas de forma válida e consistente, o que reflete uma boa compreensão dos princípios de argumentação dedutiva. Os resultados indicam que os alunos conseguem identificar e utilizar corretamente diferentes tipos de argumentos lógicos, o que é base para o desenvolvimento de habilidades críticas e analíticas. Esta compreensão é essencial para a construção de argumentos sólidos e bem fundamentados, sugerindo que os métodos pedagógicos empregados foram eficazes em transmitir esses conceitos. Futuras pesquisas poderiam explorar a aplicação dessas habilidades em contextos mais complexos e variados, fortalecendo ainda mais o domínio dos alunos na argumentação dedutiva.

Os argumentos analisados mostram a aplicação eficaz de *modus ponens*, onde a conclusão decorre diretamente de uma premissa condicional e uma afirmação dessa premissa. Por exemplo, quando um aluno afirma que “se o céu está claro, então não vai chover” e confirma que “o céu está claro”, a conclusão de que “não vai chover” segue logicamente e é válida. Similarmente, o uso de *modus tollens* foi evidenciado em exemplos como “se o alarme não disparou, então não há intrusos” e a conclusão de que “não há intrusos” quando o alarme não disparou. Esses exemplos demonstram uma compreensão sólida da estrutura lógica, reforçada pela abstração dos princípios lógicos e sua aplicação generalizada.

A aplicação do silogismo hipotético foi observada quando os alunos usaram condições para inferir resultados. Por exemplo, “se o avião se atrasasse, eles perderiam o voo; mas o avião chegou, então não perderam o voo”. Isso reflete a capacidade dos alunos de usar uma sequência lógica para prever o resultado, mostrando uma boa abstração e aplicação de princípios hipotéticos.

No caso do silogismo disjuntivo, que lida com alternativas, os alunos demonstraram compreensão ao aplicar a lógica de disjunção, como visto em “ou estudaremos ou falharemos na prova. Não estudamos, então falharemos na prova”. Esse tipo de raciocínio ilustra a habilidade dos alunos em lidar com situações que envolvem escolhas e consequências.

A combinação de abstração e generalização no ensino de pensamento computacional e argumentação dedutiva permite aos alunos uma compreensão profunda e flexível dos conceitos. A análise dos argumentos demonstra que, ao abstrair princípios lógicos e aplicá-los a uma variedade de problemas, os alunos não só internalizam as regras da lógica dedutiva, mas também são capazes de generalizar esses conceitos para resolver novos problemas de forma eficaz. Essa abordagem integrada é fundamental para desenvolver habilidades críticas de resolução de problemas e pensamento lógico.

6.4 Considerações parciais

A análise dos argumentos com base em *modus ponens*, *modus tollens*, silogismo hipotético e silogismo disjuntivo, juntamente com a perspectiva teórica de Fávero et al. (2017), Weston (2009) e Walton (2012), fornece uma compreensão aprofundada da articulação entre pensamento computacional e argumentação dedutiva. Através dessa análise, foi possível identificar como os diferentes tipos de raciocínio lógico são aplicados tanto na programação quanto na construção de argumentos válidos. A aplicação dessas estruturas lógicas evidenciou a interseção entre os processos de resolução de problemas e a formulação

de argumentos, demonstrando que a prática do pensamento computacional pode fortalecer significativamente a habilidade de argumentação dedutiva.

No contexto da pesquisa empírica, a categoria de análise que se destaca é a abstração e generalização, que desempenha um papel importante na interpretação e aplicação dos conceitos investigados. A abstração refere-se à capacidade de identificar e focar nas características essenciais dos conceitos, ignorando detalhes específicos que não são relevantes para a compreensão do problema ou argumento em questão. No contexto do pensamento computacional, isso envolve a compreensão dos princípios fundamentais que regem a lógica de programação, como a utilização de variáveis, condicionais e estruturas de repetição. A habilidade de abstrair permite que os alunos vejam além da superfície dos problemas de programação, identificando os elementos chave que precisam ser manipulados para alcançar uma solução eficaz.

Da mesma forma, na argumentação dedutiva, a abstração permite isolar e compreender os princípios lógicos que sustentam inferências válidas, como a relação entre premissas e conclusões. Isso facilita a construção de argumentos claros e precisos, baseados em uma compreensão sólida das regras lógicas. A abstração, nesse sentido, não só ajuda na construção de argumentos, mas também na avaliação crítica de argumentos apresentados por outros, uma habilidade essencial para o desenvolvimento do pensamento crítico.

A generalização, por outro lado, sugere aplicar os conceitos abstraídos a uma ampla gama de situações e exemplos. No estudo do pensamento computacional, a generalização permite que os princípios aprendidos sejam aplicados a diferentes problemas de programação, enriquecendo a capacidade dos alunos de transferir suas habilidades para novos desafios. Isso é particularmente importante em um mundo onde a tecnologia e os problemas que ela resolve estão em constante evolução. A capacidade de generalizar permite que os alunos adaptem seus conhecimentos a novos contextos, mantendo a relevância e eficácia de suas habilidades ao longo do tempo.

Na argumentação dedutiva, a generalização ajuda a entender como as estruturas lógicas podem ser usadas para resolver diferentes tipos de problemas argumentativos e para formular conclusões válidas em contextos variados. Isso não só reforça a aplicabilidade dos conceitos, mas também demonstra a flexibilidade e a universalidade das habilidades desenvolvidas. A habilidade de generalizar permite que os alunos vejam conexões entre diferentes áreas do conhecimento, aplicando princípios lógicos de forma transversal e integradora.

A combinação de abstração e generalização permitiu que a análise dos dados revelasse padrões e relações significativas entre pensamento computacional e argumentação dedutiva. Na prática pedagógica, isso resultou em uma abordagem mais estruturada e adaptativa para o ensino e a aprendizagem. Por exemplo, ao aplicar a abstração para identificar os princípios subjacentes ao raciocínio lógico e ao pensamento computacional, foram desenvolvidos questionamentos estratégicos de ensino que ajudaram os alunos a compreender e aplicar esses princípios de forma eficaz. Esses questionamentos incentivaram os alunos a identificar os elementos essenciais de um problema, tornando-os capazes de justificar suas escolhas e estratégias de resolução.

A generalização, por sua vez, permitiu que esses princípios fossem aplicados a uma variedade de problemas, preparando os alunos para enfrentar desafios diversos com uma abordagem lógica e sistemática. Isso é particularmente importante em um mundo cada vez mais interconectado e interdisciplinar, onde a capacidade de transferir habilidades e conhecimentos entre diferentes áreas é crucial para o sucesso. A prática de generalização também incentivou a criatividade e a inovação, pois permitiu que os alunos vissem novas possibilidades e soluções para problemas aparentemente não relacionados.

Portanto, a integração dessas categorias de análise na pesquisa empírica revela a importância de uma base sólida de abstração e generalização para o desenvolvimento de competências críticas e transferíveis. Essa abordagem não só facilita a compreensão e a aplicação dos conceitos investigados, mas também contribui para a formação de habilidades essenciais para a resolução de problemas complexos. Os resultados deste estudo sugerem que, ao promover essas habilidades, os educadores podem ajudar os alunos a desenvolver uma mentalidade mais analítica e adaptativa, capaz de enfrentar os desafios do século XXI de maneira eficaz e inovadora.

7 CONSIDERAÇÕES FINAIS

O problema de pesquisa proposto, que indaga sobre o papel do pensamento computacional na capacidade de produzir argumentos válidos em crianças do ensino fundamental II, estabelece um contexto enriquecedor para explorar interações entre pensamento computacional, argumentação dedutiva, programação de computadores e raciocínio lógico. Este estudo não apenas visou responder à questão central, mas também se propôs a abordar uma série de questões secundárias que aprofundam a compreensão dessas interações e seu impacto no desenvolvimento cognitivo dos alunos.

O primeiro questionamento sobre o vínculo entre raciocínio lógico e argumentação dedutiva revela a necessidade de compreender como o pensamento lógico, fundamental para a estruturação de argumentos dedutivos, influencia e é influenciado por esse processo. O vínculo entre raciocínio lógico e argumentação dedutiva revelou-se fundamental para a construção de argumentos sólidos. O raciocínio lógico proporciona a base para a estruturação e análise crítica de argumentos, permitindo que as crianças estabeleçam relações claras entre premissas e conclusões. Este estudo confirmou que uma sólida capacidade de raciocínio lógico é essencial para criar argumentos bem fundamentados e válidos, sendo um facilitador base para a argumentação dedutiva eficaz. A capacidade de desenvolver raciocínio lógico é um pré-requisito essencial para que os alunos consigam formular argumentos coerentes e justificados, demonstrando uma relação direta entre a lógica aplicada em atividades de programação e a lógica necessária para construir argumentos dedutivos.

O segundo questionamento sobre as articulações necessárias para conectar premissas e conclusão na construção de um argumento dedutivo direciona a atenção para os processos mentais envolvidos. Neste viés, o estudo demonstrou que a habilidade de identificar e aplicar regras lógicas é fundamental. A construção de um argumento dedutivo exige que as premissas sejam claramente relacionadas à conclusão através de uma cadeia lógica de inferências. O pensamento computacional, ao promover a prática de seguir algoritmos e decompor problemas complexos, fornece uma estrutura que ajuda a garantir que essas conexões sejam feitas de maneira sistemática e coerente. A aplicação de algoritmos na programação ensina os alunos a seguir sequências lógicas de passos, o que é essencial na construção de argumentos dedutivos. A prática de decomposição de problemas complexos em partes menores e gerenciáveis ajuda a estruturar as premissas de maneira que cada uma contribua diretamente para a conclusão, criando uma linha de raciocínio clara e convincente.

O terceiro questionamento, relacionado às conexões do pensamento computacional na formação de um ser argumentativo que resolve problemas ao programar computadores, destaca a importância de explorar a transferência de habilidades entre o pensamento computacional e o desenvolvimento da capacidade argumentativa. Analisar como as competências adquiridas ao programar computadores influenciam positivamente a resolução de problemas e a construção de argumentos oferece uma perspectiva prática sobre o impacto do pensamento computacional no pensamento argumentativo. A prática de programação não apenas desenvolve habilidades técnicas, mas também promove habilidades críticas de resolução de problemas e pensamento analítico, que são fundamentais para a argumentação eficaz. A capacidade de programar envolve não apenas seguir instruções, mas também adaptar, testar e refinar soluções, o que reflete diretamente na habilidade de construir e defender argumentos robustos e bem fundamentados.

Outro ponto a destacar são os pilares do pensamento computacional: decomposição, reconhecimento de padrões, abstração e algoritmo. A habilidade de decomposição do pensamento computacional auxilia os alunos na identificação e compreensão das premissas de um argumento dedutivo. O reconhecimento de padrões no pensamento computacional facilita a construção de argumentos coerentes, ao identificar relações lógicas entre premissas e conclusões. A capacidade de abstração no pensamento computacional contribui para a generalização de conclusões a partir de premissas específicas em um argumento dedutivo. Compreender algoritmos no pensamento computacional fortalece o raciocínio lógico dos alunos, auxiliando-os a seguir uma sequência de passos precisos para derivar conclusões válidas a partir de premissas. Além disso, a habilidade de resolver problemas no pensamento computacional pode ser aplicada na formulação e avaliação de argumentos dedutivos, especialmente ao testar a validade das conclusões derivadas das premissas. Essas habilidades permitem que as crianças abordem a resolução de problemas de maneira lógica e estruturada, aprimorando sua capacidade de construir e defender argumentos de forma eficaz. A prática de programação, ao envolver a resolução de problemas complexos e a criação de soluções, contribui para a formação de um indivíduo argumentativo que é capaz de aplicar o raciocínio lógico e a argumentação dedutiva em diversos contextos.

A decomposição, uma das competências centrais do pensamento computacional, permite que os alunos desmembrem um problema complexo em partes menores e mais manejáveis. No contexto de um argumento dedutivo, essa habilidade facilita a análise e o entendimento das premissas ao permitir que os alunos se concentrem em cada componente individualmente, compreendendo seu papel específico dentro da lógica do argumento. Assim,

a decomposição ajuda os alunos a perceberem como cada premissa contribui para a conclusão final, promovendo uma compreensão mais clara e detalhada do raciocínio dedutivo. Isso, por sua vez, melhora a capacidade dos alunos de construir e avaliar argumentos dedutivos de maneira lógica e estruturada.

O reconhecimento de padrões permite que os alunos percebam similaridades, diferenças, e repetições dentro das premissas e entre as premissas e a conclusão. Ao identificar essas relações lógicas, os alunos conseguem estruturar argumentos de maneira mais organizada e consistente, garantindo que a transição das premissas para a conclusão ocorra de forma lógica e fluida. Isso fortalece a coerência do argumento como um todo, permitindo que ele seja mais convincente e resistente a críticas. Em resumo, o reconhecimento de padrões no pensamento computacional é uma ferramenta para aprimorar a clareza e a lógica interna dos argumentos dedutivos.

A habilidade de abstrair permite aos indivíduos identificar os elementos essenciais de um problema ou situação, ignorando detalhes irrelevantes. Isso facilita a construção de argumentos dedutivos robustos, onde as premissas específicas podem ser aplicadas para derivar conclusões mais amplas e generalizáveis. Portanto, a abstração não apenas simplifica a compreensão dos componentes críticos de um argumento, mas também permite a aplicação desses componentes em diferentes contextos, promovendo a criação de argumentos dedutivos que são válidos em uma variedade de situações. Em resumo, a capacidade de abstração é fundamental para a generalização de conclusões, tornando o pensamento dedutivo mais versátil e aplicável a diferentes cenários.

A compreensão de algoritmos no pensamento computacional fortalece significativamente o raciocínio lógico dos alunos, pois os ensina a seguir uma sequência de passos precisos e sistemáticos na derivação de conclusões a partir de premissas. Algoritmos, por sua natureza, envolvem a execução ordenada de instruções que levam a um resultado específico. Ao internalizar essa lógica, os alunos aprendem a aplicar um processo similar na construção de argumentos dedutivos, onde cada passo deve ser cuidadosamente considerado para garantir que a conclusão seja válida e consistente com as premissas. Essa abordagem metódica ajuda a evitar saltos lógicos ou falácias, garantindo que cada conclusão seja solidamente baseada nos dados ou nas informações fornecidas pelas premissas. Portanto, compreender algoritmos não só aprimora a capacidade dos alunos de raciocinar de forma lógica e estruturada, mas também os capacita a elaborar argumentos dedutivos de maneira mais precisa e confiável. Em resumo, a integração do pensamento algorítmico no processo argumentativo contribui para um raciocínio mais rigoroso e fundamentado.

A análise dos dados obtidos ao longo da pesquisa permitiu identificar que o pensamento computacional desempenha um papel significativo na capacidade dos alunos de elaborar argumentos dedutivos. O processo de programação, que envolve a decomposição de problemas, a identificação de padrões, a abstração e a construção de algoritmos, contribui diretamente para o desenvolvimento do raciocínio lógico. Este, por sua vez, é fundamental na construção de argumentos dedutivos coerentes e bem fundamentados. Além disso, o pensamento computacional promove habilidades como a capacidade de depurar, ou seja, identificar e corrigir erros, o que reforça o pensamento crítico e a capacidade de revisão e refinamento de ideias.

Observou-se que o vínculo entre raciocínio lógico e argumentação dedutiva é estreito, pois ambos compartilham a necessidade de uma estrutura clara e sequencial de pensamento. As atividades de programação exigem que os alunos analisem problemas de forma lógica e sistemática, habilidades que são transferíveis para a construção de argumentos dedutivos. Ao programar, os alunos aprendem a articular premissas de maneira estruturada para chegar a uma conclusão válida, o que reflete diretamente nas suas capacidades argumentativas. Esse processo de articulação é essencial para a argumentação dedutiva, onde cada premissa deve ser claramente definida e conectada de forma lógica para sustentar a conclusão.

As articulações necessárias para conectar premissas e conclusões em um argumento dedutivo foram evidenciadas nas práticas de ensino que integraram o pensamento computacional. As tarefas de programação proporcionaram um ambiente onde os alunos puderam exercitar a justificativa de suas escolhas e decisões, aprimorando assim suas habilidades de argumentação. A prática constante de depuração de código, por exemplo, ensinou-os a revisar e refinar seus pensamentos, reforçando a conexão entre premissas e conclusões. Este processo de revisão e correção é importante para a construção de argumentos sólidos, pois permite que os alunos identifiquem falhas em seu raciocínio e façam ajustes necessários para fortalecer suas argumentações.

As conexões entre o pensamento computacional e a formação de habilidades argumentativas e de resolução de problemas foram claramente evidenciadas ao longo do estudo. Os alunos que cultivaram habilidades de pensamento computacional demonstraram uma maior capacidade de abordar problemas complexos com lógica e criatividade. Ao aplicar os conceitos aprendidos na programação, eles conseguiram encontrar soluções eficazes em uma variedade de contextos. Essa abordagem estruturada e metódica, característica do pensamento computacional, não apenas fortalece a capacidade dos alunos de construir

argumentos sólidos e bem fundamentados, mas também estimula um pensamento crítico mais profundo.

Além disso, a exposição a desafios de programação que são realistas e instigantes ofereceu aos alunos oportunidades valiosas para desenvolver resiliência e persistência. Essas habilidades são fundamentais para qualquer processo de resolução de problemas, pois encorajam os alunos a enfrentar obstáculos com confiança e a buscar soluções, mesmo diante de dificuldades. Dessa maneira, a integração do pensamento computacional na educação se revela não apenas como uma ferramenta para o aprendizado técnico, mas também como um suporte essencial para o desenvolvimento de competências críticas que são vitais no mundo contemporâneo.

Em conclusão, o estudo evidenciou que o pensamento computacional não só aprimora a capacidade dos alunos de produzir argumentos válidos, mas também fortalece o raciocínio lógico necessário para a argumentação dedutiva. A integração de práticas de programação no ensino fundamental é, portanto, altamente recomendada para desenvolver habilidades cognitivas essenciais em um mundo cada vez mais digital e complexo. Os resultados empíricos obtidos oferecem uma base para aprimorar as práticas pedagógicas, garantindo que os alunos estejam melhor preparados para os desafios do futuro. O desenvolvimento dessas competências não só facilita a navegação dos alunos pelo mundo digital, mas também os prepara para serem cidadãos críticos, capazes de pensar de forma lógica e criativa, e contribuir de maneira significativa para a sociedade.

As implicações para a prática pedagógica são significativas. Integrar o pensamento computacional no currículo escolar pode proporcionar benefícios substanciais para o desenvolvimento das habilidades argumentativas. Recomenda-se que educadores adotem práticas que promovam o pensamento computacional e a argumentação dedutiva, utilizando o ensino de programação como uma ferramenta para fortalecer essas habilidades. A abordagem sugerida pode preparar melhor os estudantes para enfrentar desafios acadêmicos e cotidianos de maneira mais estruturada e lógica.

Por fim, o estudo confirmou que o pensamento computacional é um recurso valioso para o desenvolvimento da argumentação dedutiva. A integração do ensino de programação e o desenvolvimento das habilidades associadas ao pensamento computacional têm o potencial de melhorar significativamente a capacidade dos alunos de construir argumentos válidos e de aplicar um raciocínio lógico e crítico.

REFERÊNCIAS

- ABBAGNANO, N. **Dicionário de filosofia**. Tradução: Alfredo Bosi e Ivone Castilho Benedetti. 5ª. ed. São Paulo: Martins Fontes, 2007.
- ALMEIDA, M.E. **Informática e formação de professores**. Secretaria de Educação a Distância. ProInfo- Brasília: Ministério da Educação, SEED. 2000.
- ARISTÓTELES. **Órganon**. Tradução: Edson Bini. 3ª. ed. [S. l.]: Edipro, 2016. 648 p.
- ARSLAN, C.; GOCMENCELEBI, S. I.; TAPAN, M. S. Learning and reasoning styles of pre service teachers: inductive or deductive reasoning on science and mathematics related to their learning style. **Procedia - Social and Behavioral Sciences**, v. 1, n. 1, p. 2460-2465, 2009.
- AYALON, M.; EVEN, R. Deductive reasoning: in the eye of the beholder. **Educational Studies in Mathematics**, v. 69, n. 3, p. 235-247, 2008.
- _____. Mathematics educators' views on the role of mathematics learning in developing deductive reasoning. **International Journal of Science and Mathematics Education**, v. 8, n. 6, p. 1131-1154, 2010.
- BBC LEARNING, Bitesize. **What is computational thinking?** s.d. Disponível em: <<http://www.bbc.co.uk/education/guides/zp92mp3/revision>>. Acesso em: 23 nov. 2023.
- BOGDAN, Robert; BIKLEN, Sarli Knopp. **Investigação qualitativa em educação: uma introdução à teoria e aos métodos**. Portugal: Porto Editora, 1994. p. 47-70.
- BOUCINHA, Rafael Marimon. **Aprendizagem do Pensamento Computacional e Desenvolvimento do Raciocínio**. 151 f. Tese (Doutorado) - Curso de Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BRACKMANN, Christian Puhlmann. **Desenvolvimento do Pensamento Computacional através de Atividades Desplugadas na Educação Básica**. 226 f. Tese (Doutorado) - Curso de Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.
- BRASIL. Ministério da Educação. **Parâmetros Curriculares Nacionais: Terceiro e quarto ciclos do Ensino Fundamental**. Brasília: Ministério da Educação, 1998. Disponível em: <<http://portal.mec.gov.br/seb/arquivos/pdf/introducao.pdf>>. Acesso em: 15/04/2021
- _____. Ministério da Educação. **Base Nacional Comum Curricular: Educação é a base**. Brasília: Ministério da Educação, 2017. Disponível em: <<http://basenacionalcomum.mec.gov.br/implementacao/praticas/caderno-de-praticas/aprofundamentos/193-tecnologias-digitais-da-informacao-e-comunicacao-no-contexto-escolar-possibilidades>> Acesso em 22 de outubro de 2020.
- BROWN, F. M.; TARNLUND, S. A. Inductive reasoning in Mathematics. In: **International Joint Conference On Artificial Intelligence**, 5., 1977, Cambridge, Massachusetts, USA. USA: Massachusetts Institute of Technology, 1977. v. 2. p. 844-850.

CORREIA, Luiz Henrique Andrade; SILVA, Alexandre José de Carvalho. **Computador Tutelado**. Lavras: Universidade Federal de Lavras - UFLA, 2005.

DESLANDES, Suely Ferreira. O projeto de pesquisa como exercício científico e artesanato intelectual. In: MINAYO, Maria Cecília de Souza (Org.). **Pesquisa Social: teoria, método e criatividade**, Petrópolis: Vozes, 2018. p. 29-55.

ECO, Umberto. **Como se faz uma tese**. 26 ed. São Paulo: Perspectiva, 2016.

EDUScratch. **Site do Scratch para Educadores**. Disponível em: <<http://eduscratch.dgidec.min-edu.pt>>. Acesso em: 10 jun. 2022.

FÁVERO, Alcemira Maria; FÁVERO, Altair Alberto; TONIETO, Carina; RAUBER, Jaime José; ROSSETO, Miguel da Silva. **Que tal um pouco de lógica?** 6. ed. Passo Fundo: Méritos, 2017. 128 p.

GIL, Antonio Carlos. **Como elaborar um projeto de pesquisa**. 4 ed. São Paulo: Atlas, 2002.

_____. **Métodos e técnicas de pesquisa social**. 7 ed. São Paulo: Atlas, 2019.

GOMES, Romeu. Análise e interpretação de dados de pesquisa qualitativa. In: MINAYO, Maria Cecília de Souza (Org.). **Pesquisa social: teoria, método e criatividade**. Petrópolis: Vozes, 2018. p. 72-94.

GONZÁLEZ, Marcos Román. Computational Thinking Test: Design Guidelines and Content Validation. **Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015)**. p.2436–2444, 2015. Barcelona, Spain: IATED.

GONZÁLEZ, Marcos Román. **Código alfabetización y Pensamiento Computacional en Educación Primaria y Secundaria**: Validación de un Instrumento y Evaluación de Programas, fev. 2016. Madrid, Spain: Universidad Nacional de Educación a Distancia.

GONZÁLEZ, Marcos Román; PÉREZ, J. C.; FERNÁNDEZ, Carmen Jiménez. **Test de Pensamiento Computacional**: diseño y psicometría general. 2015. Disponível em: <<https://doi.org/10.13140/RG.2.1.3056.5521>>. Acesso em: 16/04/2023.

HAVERTY, L. A. et al. **Solving inductive reasoning problems in Mathematics**: not-so-trivial pursuit. *Cognitive Science*, v. 24, n. 2, p. 249-298, 2000.

HUMPHREYS, Simon et alli. **Computational thinking: A guide for teachers 2015**. Disponível em: <<http://computingatschool.org.uk/computationalthinking>>

KENSKI, Vani Moreira. **Educação e tecnologias: o novo ritmo da informação**. 8ª. ed. São Paulo: Papirus, 2012. 141p.

LAKATOS, Eva Maria; MARCONI, Marina de Andrade. **Metodologia do trabalho científico: procedimentos básicos, pesquisa bibliográfica, projeto e relatório, publicações e trabalhos científicos**. 4 ed. São Paulo: Atlas, 1992.

LESSA, Valéria Espíndola. **A programação de computadores e a função afim**: um estudo sobre a representação e a compreensão de invariantes operatórios. 184 f. Tese (Doutorado) - Curso de Educação, Universidade de Passo Fundo, Passo Fundo, 2018.

LUNA, Sérgio Vasconcelos de. **Planejamento de pesquisa**: Uma introdução. 2 ed. São Paulo: EDUC, 2019.

MALAN, D. J.; LEITNER, H. H. **Scratch for budding computer scientists**. In: Technical Symposium On Computer Science Education, 38, 2007, Covington. Covington: SIGCSE, 2007. p. 223-227.

MARCONI, Marina de Andrade; LAKATOS, Eva Maria. **Fundamentos de metodologia científica**. 5 ed. São Paulo: Atlas, 2003.

_____. **Técnicas de pesquisa**. 8 ed. São Paulo: Atlas, 2018.

MESTRE, P. et al. **O uso do pensamento computacional como estratégia para resolução de problemas matemáticos**. 2017.

MINAYO, Maria Cecília de S. **O desafio do conhecimento**: Pesquisa qualitativa em saúde. 4 ed. São Paulo-Rio de Janeiro: Heucitec-Abrasco, 1996.

_____. **Pesquisa social: Teoria, método e criatividade**. 21 ed. Petrópolis: Vozes, 2002.

MIT LIBRARIES. **DSpace@MIT**. Cambridge, Massachusetts, 2016. Disponível em: <https://libraries.mit.edu/>. Acesso em: 30 de junho de 2024.

MORAN, José Manuel; MASETTO, Marcos T.; BEHRENS, Marilda Aparecida. **Novas tecnologias e mediação pedagógica**. Campinas : Papirus, 2000.

NEUBERT, G.; BINKO, J. B. **Inductive reasoning in the secondary classroom**. Washington, D.C.: National Education Association, 1992. 127 p. (Aspects of Learning Series).

PAPAGEORGIO, E. Towards a teaching approach for improving Mathematics inductive reasoning problem solving. In: **Conference Of The International Group For The Psychology Of Mathematics Education**, 33., 2009, Cidade: Thessaloniki, Grécia, 2009. p. 313-320. v. 4.

PAPERT, Seymour. **Teaching Children Thinking**, Logo Memo nº 2, 1971. Disponível em: <<https://goo.gl/9is34j>>. Acesso em: 28 de julho de 2020.

_____. **Logo**: Computadores e educação. São Paulo: Brasiliense, 1985.

_____. **A máquina das crianças**: repensando a escola na era da informática. Porto Alegre: Artmed, 2008.

PARK, J.; HAN, S. Using deductive reasoning to promote the change of students' conceptions about force and motion. **International Journal of Science Education**, v. 24, n. 6, p. 593-609, 2002.

PEIRCE, C. S. **Semiótica**. 3. ed. São Paulo: Perspectiva, 2003. 337 p.

PIAGET, Jean. **O nascimento da inteligência na criança**. 3.ed. Rio de Janeiro: Zahar, 1978.

PÓLYA, G. **A arte de resolver problemas**: Um novo aspecto do método matemático. Tradução: Heitor Lisboa de Araújo. Rio de Janeiro: Interciência, 2006.

REID, D.; KNIPPING, C. **Proof in mathematics education**: research, learning and teaching. Canada: Sense Publishers, 2010.

RAABE, André. **Pensamento Computacional para educadores**. 2016. Acesso em 12 de setembro de 2022.

RAABE, André. et al. **Um Instrumento para Diagnóstico do Pensamento Computacional**. Anais dos Workshops do Congresso Brasileiro de Informática na Educação, [S.l.], p. 1172, 2017. Disponível em: <<https://www.br-ie.org/pub/index.php/wcbie/article/view/7506/5301>>. Acesso em: 22 de dezembro de 2022.

RESNICK, Mitchel. Technologies for lifelong kindergarten. **Educational Technology Research & Development**, Boston, v. 46, n. 4, p. 43-55, December 1998. Disponível em: <http://web.media.mit.edu/~mres/papers.html>. Acesso em: 01/10/2022.

_____. **Jardim da infância para a vida toda**: por uma aprendizagem criativa, mão na massa e relevante para todos. Tradução: Mariana Casetto Cruz, Lívia Rulli Sobral. Porto Alegre: Penso, 2020.

ROBINSON, Ken. **Somos todos criativos**: Os desafios para desenvolver uma das principais habilidades do futuro. Tradução: Cristina Yamagami. São Paulo: Benvirá, 2019.

ROMANOWSKI, Joana Paulin; ENS, Romilda Teodora. **As pesquisas denominadas do tipo “estado da arte” em educação**. Revista Diálogo Educacional. Curitiba, v. 6, n. 19, p. 37-50, set-dez, 2006.

RUSHKOFF, Douglas. **As 10 questões essenciais da era digital**. São Paulo: Editora Saraiva, 2012.

SACRINI, Marcus. **Introdução à análise argumentativa**: teoria e prática. São Paulo: Paulus, 2016.

SANCHO, D. **Os professores e sua formação**. Lisboa: Nova Enciclopédia, 1998.

SCRATCH. **Imagine, program, share**. 2017. Disponível em: <http://Scratch.mit.edu>. Acesso em: 11 de junho de 2023.

SOUZA, J. S. **A abdução em Peirce**: um estudo hermenêutico. 2015. 92 f. Dissertação (Mestrado em Educação Matemática) – Universidade Estadual Paulista “Júlio de Mesquita Filho”, Rio Claro, 2015.

TEIXEIRA, Jaylon. **Contribuições para o Ensino de programação de computadores a futuros professores de Matemática**. 2017. Tese (Doutorado) - Universidade do Minho, Especialidade em Tecnologia Educativa, 2017.

TOFFLER, Alvin. **A terceira onda**. 32^a. ed. Rio de Janeiro: Record, 2014.

VALENTE, José Armando. **A Telepresença na Formação de Professores da Área de Informática em Educação**: implantando o construcionismo contextualizado. IV Congresso RIBIE, Brasília, 1998. Disponível em: <http://www.ufrgs.br/niece/eventos/RIBIE/1998/pdf/com_pos_dem/232.pdf>. Acesso em: 17 out. 2022.

WALTON, Douglas N. **Lógica informal**. Tradução: Ana Lúcia Franco e Carlos Salum. 2^a. ed. São Paulo: WMF, 2012. 410p.

WESTON, Anthony. **A construção do argumento**. São Paulo: WMF Martins Fontes, 2009. 111p. Tradução de: Alexandre Feitosa Rosas.

WING, Jeannette M. **Computational thinking**. In: Communications of The ACM, Março 2006. Vol. 49, nº 3. p. 33-35.

_____. **Computational Thinking: What and why?** 2010. Disponível em: <https://www.cs.cmu.edu/link/research-notebook-computational-thinking-what-and-why>. Acesso em: 26/05/2020.

_____. **Computational thinking benefits society**. 2014. Disponível em: <http://socialissues.cs.toronto.edu/index.html%3Fp=279.html>. Acesso em 26/05/2020.

ANEXO A - Questionário sobre argumento dedutivo

ARGUMENTO DEDUTIVO

Bem-vindo ao questionário sobre argumento dedutivo! Este questionário é composto por 20 questões que abordam diferentes tipos de argumentos dedutivos, incluindo *Modus Ponens*, *Modus Tollens*, Silogismo Hipotético e Silogismo Disjuntivo, conforme o modelo representado abaixo.

<i>Modus ponens (MP)</i> Se [proposição A], então [proposição B]. [Proposição A]. Portanto, [proposição B].	<i>Modus tollens (MT)</i> Se [proposição P], então [proposição Q]. [Proposição Q] é falso. Portanto, [proposição P] é falso
Silogismo hipotético (SH) Se [proposição R], então [proposição S]. Se [proposição S], então [proposição T]. Portanto, se [proposição R], então [proposição T]	Silogismo disjuntivo (SD) Ou [proposição M] ou [proposição N]. [Proposição M] é falso. Portanto, [proposição N].

O questionário está dividido em 5 etapas, e em cada etapa, você encontrará 4 questões. Cada questão apresentará um argumento e algumas perguntas relacionadas para avaliar sua compreensão e habilidade de raciocínio lógico.

QUESTÃO 1: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se o céu está claro, então não vai chover.
 Premissa 2: O céu está claro.
 Conclusão: Portanto, não vai chover.

Avaliação das Premissas:

a) Você concorda que se o céu está claro, então não vai chover?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que o céu está claro?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que não vai chover é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 2: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se o alarme não disparou, então não há intrusos.
Premissa 2: Há intrusos.
Conclusão: Portanto, o alarme disparou.

Avaliação das Premissas:

a) Você concorda que se o alarme não disparou, então não há intrusos?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que há intrusos?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que o alarme disparou é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 3: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se chover amanhã, levarei meu guarda-chuva.
Premissa 2: Choverá amanhã.

Conclusão: Portanto, levarei meu guarda-chuva.
--

Avaliação das Premissas:

a) Você concorda que se chover amanhã, então você levará seu guarda-chuva?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que choverá amanhã?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que você levará seu guarda-chuva é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 4: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Ou estudamos para a prova ou falharemos. Premissa 2: Não estudamos para a prova. Conclusão: Portanto, falharemos.

Avaliação das Premissas:

a) Você concorda que ou estudamos para a prova ou falharemos?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que não estudamos para a prova?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que falharemos é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 5: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se estudarmos com dedicação, obteremos boas notas.

Premissa 2: Estudamos com dedicação.

Conclusão: Portanto, obteremos boas notas.

Avaliação das Premissas:

a) Você concorda que se estudarmos com dedicação, então obteremos boas notas?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que estudaram com dedicação?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que obterão boas notas é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 6: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se a bateria estiver carregada, o dispositivo ligará.
Premissa 2: O dispositivo não ligou.
Conclusão: Portanto, a bateria não está carregada.

Avaliação das Premissas:

a) Você concorda que se a bateria estiver carregada, então o dispositivo ligará?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que o dispositivo não ligou?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que a bateria não está carregada é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 7: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se estudarmos juntos, entenderemos melhor o assunto.
Premissa 2: Estamos estudando juntos.
Conclusão: Portanto, entenderemos melhor o assunto.

Avaliação das Premissas:

a) Você concorda que se estudarmos juntos, então entenderemos melhor o assunto?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que estão estudando juntos?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que entenderão melhor o assunto é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 8: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Ou ele vai ao cinema ou fica em casa.

Premissa 2: Ele não vai ao cinema.

Conclusão: Portanto, ele fica em casa.

Avaliação das Premissas:

a) Você concorda que ou ele vai ao cinema ou fica em casa?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que ele não vai ao cinema?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que ele fica em casa é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 9: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se João acordar cedo, ele chegará a tempo para a reunião.
Premissa 2: João acordou cedo.
Conclusão: Portanto, ele chegará a tempo para a reunião.

Avaliação das Premissas:

a) Você concorda que se João acordar cedo, então ele chegará a tempo para a reunião?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que João acordou cedo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que ele chegará a tempo para a reunião é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 10: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se houver neve, a escola será fechada.
Premissa 2: A escola não foi fechada.
Conclusão: Portanto, não há neve.

Avaliação das Premissas:

a) Você concorda que se houver neve, então a escola será fechada?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que a escola não foi fechada?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que não há neve é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 11: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se Pedro economizar dinheiro, ele poderá viajar nas férias.
 Premissa 2: Pedro economizou dinheiro.
 Conclusão: Portanto, ele poderá viajar nas férias.

Avaliação das Premissas:

a) Você concorda que se Pedro economizar dinheiro, então ele poderá viajar nas férias?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que Pedro economizou dinheiro?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que ele poderá viajar nas férias é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 12: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Ou acordamos cedo ou perderemos o voo.
 Premissa 2: Não acordamos cedo.
 Conclusão: Portanto, perderemos o voo.

Avaliação das Premissas:

a) Você concorda que ou acordamos cedo ou perderemos o voo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que não acordamos cedo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que perderemos o voo é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 13: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se a temperatura cair abaixo de zero, a água congelará.
 Premissa 2: A temperatura caiu abaixo de zero.
 Conclusão: Portanto, a água congelará.

Avaliação das Premissas:

a) Você concorda que se a temperatura cair abaixo de zero, então a água congelará?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que a temperatura caiu abaixo de zero?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que a água congelará é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 14: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se Maria tivesse ganho o concurso, estaríamos sabendo.
 Premissa 2: Não estamos sabendo.
 Conclusão: Portanto, Maria não ganhou o concurso.

Avaliação das Premissas:

a) Você concorda que se Maria tivesse ganho o concurso, então estaríamos sabendo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que não estamos sabendo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que Maria não ganhou o concurso é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 15: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se eu terminar meu trabalho mais cedo, poderei sair mais cedo.

Premissa 2: Terminei meu trabalho mais cedo.
 Conclusão: Portanto, posso sair mais cedo.

Avaliação das Premissas:

a) Você concorda que se você terminar seu trabalho mais cedo, então poderá sair mais cedo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que você terminou seu trabalho mais cedo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que você pode sair mais cedo é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 16: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Ou ela pagará a multa ou terá pontos na carteira.
 Premissa 2: Ela não pagará a multa.
 Conclusão: Portanto, terá pontos na carteira.

Avaliação das Premissas:

a) Você concorda que ou ela pagará a multa ou terá pontos na carteira?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que ela não pagará a multa?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que ela terá pontos na carteira é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 17: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se alimentarmos o gato, ele ficará satisfeito.
Premissa 2: Alimentamos o gato.
Conclusão: Portanto, ele está satisfeito.

Avaliação das Premissas:

a) Você concorda que se alimentarmos o gato, então ele ficará satisfeito?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que alimentamos o gato?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que ele está satisfeito é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 18: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se o avião estiver atrasado, perderemos a conexão.
Premissa 2: Não perdemos a conexão.
Conclusão: Portanto, o avião não está atrasado.

Avaliação das Premissas:

a) Você concorda que se o avião estiver atrasado, então perderemos a conexão?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que não perdemos a conexão?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que o avião não está atrasado é logicamente válida?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 19: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Se a equipe ganhar o jogo, teremos uma comemoração.
Premissa 2: A equipe ganhou o jogo.
Conclusão: Portanto, teremos uma comemoração.

Avaliação das Premissas:

a) Você concorda que se a equipe ganhar o jogo, então teremos uma comemoração?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

b) Você concorda que a equipe ganhou o jogo?

☐ Concordo

☐ Discordo

☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que teremos uma comemoração é logicamente válida?

☐ Concordo

- ☐ Discordo
☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

QUESTÃO 20: Leia o argumento apresentado e responda às perguntas que se seguem.

Premissa 1: Ou o projeto será concluído hoje ou será adiado.
Premissa 2: O projeto não será concluído hoje.
Conclusão: Portanto, será adiado.

Avaliação das Premissas:

a) Você concorda que ou o projeto será concluído hoje ou será adiado?

- ☐ Concordo
☐ Discordo
☐ Não sei / Não tenho certeza

b) Você concorda que o projeto não será concluído hoje?

- ☐ Concordo
☐ Discordo
☐ Não sei / Não tenho certeza

Avaliação da Conclusão:

c) Você concorda que, com base nas premissas fornecidas, a conclusão de que o projeto será adiado é logicamente válida?

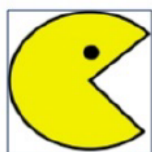
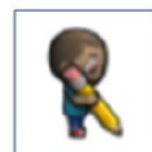
- ☐ Concordo
☐ Discordo
☐ Não sei / Não tenho certeza

Explicação:

d) Explique brevemente por que você concorda ou discorda da validade do argumento apresentado.

ANEXO B - Teste de pensamento computacional**TESTE DE PENSAMENTO COMPUTACIONAL⁸****INSTRUÇÕES⁹**

O teste é composto por 16 perguntas, distribuídas em 4 seções com aproximadamente 4 perguntas em cada uma. Todas as questões têm 4 alternativas de resposta (A, B, C ou D) das quais apenas uma é correta. A partir do início do teste, você dispõe de 30 minutos para fazer o melhor que puder. Caso necessite ampliar alguma pergunta, utilize o atalho "Control +" para chegar mais perto e "Control -" para diminuir o tamanho das imagens e textos. Antes de iniciar o teste, veremos 3 exemplos para que você se familiarize com o tipo de perguntas que encontrará e nas quais aparecerão os personagens a seguir.

**'Pac-Man'****Fantasma****Artista**

⁸ Este teste foi desenvolvido pelo Prof. Dr. Marcos Román-Gonzales da Universidad Nacional de Educación a Distancia (UNED) intitulada "Test de Pensamiento Computacional: diseño y psicometría general" e traduzido/adaptado pelos pesquisadores Rafael Marimon Boucinha e Christian Puhlmann Brackmann da Universidade Federal do Rio Grande do Sul (UFRGS). É necessário solicitar autorização do uso do teste para o autor do mesmo, e o mesmo autorizou o uso neste trabalho.

⁹ Devido a formatação exigida pela ABNT, foi necessária a ampliação do número de páginas.

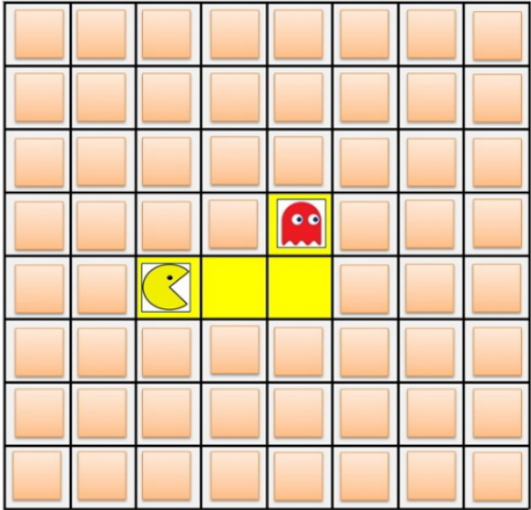
EXEMPLO 1

Neste primeiro exemplo, você é questionado sobre quais são as ordens que levam 'Pac-Man' ao fantasma pelo caminho indicado.

Ou seja, leve 'Pac-Man' EXATAMENTE até o quadrado onde está o fantasma (sem passar por cima ou parar), e seguindo rigorosamente o caminho indicado em amarelo (sem sair e sem tocar nas paredes, representadas pelos quadrados laranja)

A opção correta neste exemplo é B. Marque-a no botão de resposta correspondente, que está abaixo da pergunta.

Qual sequência leva o "Pac-Man até o fantasma pelo caminho indicado?

	Alternativa A 
	Alternativa B  
	Alternativa C 
	Alternativa D 

EXEMPLO 2

Neste segundo exemplo, se pergunta de novo quais são os comandos que levam o 'Pac-Man' até o fantasma pelo caminho assinalado. Mas neste caso, as opções de resposta, em vez de ser flechas, são blocos que encaixam uns nos outros.

Lembrando que a pergunta pede para levar o 'Pac-Man' EXATAMENTE a casa em que se encontra o fantasma (sem passar nem parar), e seguindo estritamente o caminho marcado em amarelo (sem sair e sem tocar nas paredes, representadas pelos quadrados laranja).

A alternativa correta neste exemplo é a C. Marque a alternativa correspondente, que está abaixo da pergunta.

Qual sequência leva o Pac-Man até o fantasma pelo caminho indicado?

	Alternativa A <pre> avance vire à esquerda ⤴ avance avance </pre>	Alternativa B <pre> avance vire à direita ⤵ avance avance </pre>
	Alternativa C <pre> avance avance vire à esquerda ⤴ avance </pre>	Alternativa D <pre> avance avance vire à direita ⤵ avance </pre>

EXEMPLO 3

Neste terceiro exemplo se pergunta que comandos deve seguir o artista para desenhar a figura que aparece na tela. Ou seja, como deve MOVER o lápis para que se desenhe a figura.

O comando MOVER empurra o lápis desenhando, enquanto que o comando SALTAR faz um alto ao artista sem desenhar. A seta cinza indica a direção do primeiro movimento da caneta.

A alternativa correta neste exemplo é A. Marque a alternativa correspondente, que está abaixo da pergunta.

Qual sequência o artista deve seguir para desenhar a figura abaixo? Lembrando que o lado menor mede 50 pixels e o lado maior mede 100 pixels.

	Alternativa A  <pre> avance por 50 pixels vire à esquerda por 90 graus avance por 100 pixels </pre>	Alternativa B <pre> avance por 50 pixels vire à direita por 90 graus avance por 100 pixels </pre>
	Alternativa C <pre> avance por 100 pixels vire à esquerda por 90 graus avance por 50 pixels </pre>	Alternativa D <pre> avance por 100 pixels vire à direita por 90 graus avance por 50 pixels </pre>

QUESTÃO 1

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?

	Alternativa A
	Alternativa B
	Alternativa C
	Alternativa D

QUESTÃO 2

Qual comando está faltando na sequência para levar o "Pac-Man" até o fantasma pelo caminho indicado?

	Alternativa A
	Alternativa B
	Alternativa C
	Alternativa D

QUESTÃO 3

Para levar o "Pac-Man" até o fantasma pelo caminho indicado, qual passo da sequência está INCORRETO?

--	--

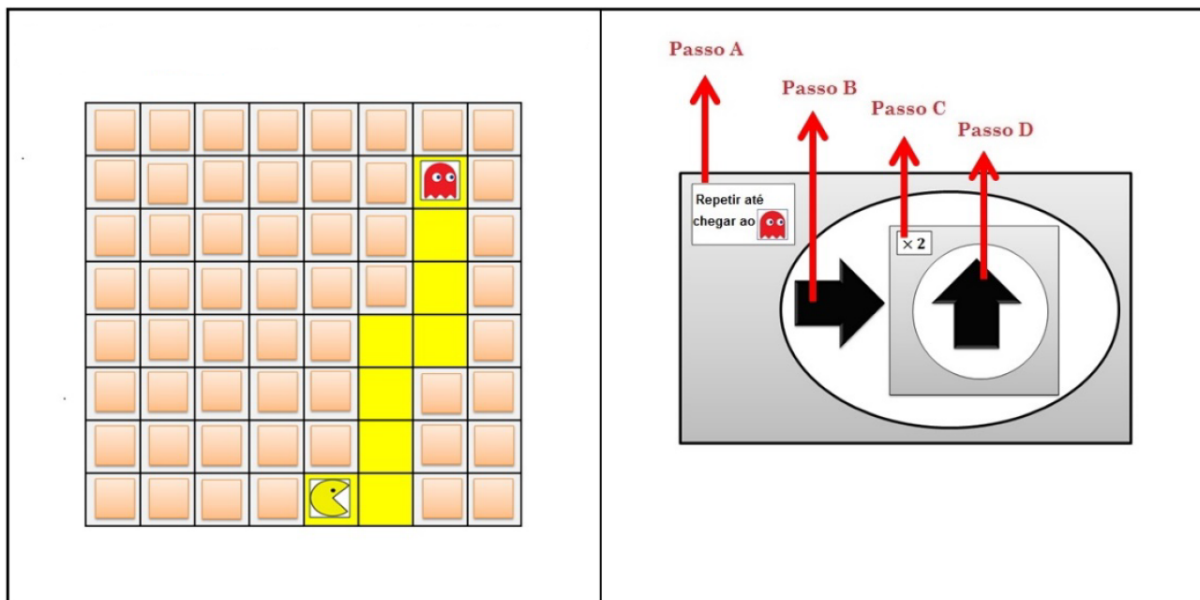
QUESTÃO 4

Para que o artista desenhe uma vez o seguinte retângulo (50 pixels de largura e 100 pixels de altura), qual passo da sequência está **INCORRETO**?

--	--

QUESTÃO 5

Para que o "Pac-Man" chegue até o fantasma pelo caminho indicado, qual passo da sequência está **INCORRETO**?



QUESTÃO 6

Qual sequência o artista deve seguir para desenhar a escada que leva até a flor? Cada degrau sobe 30 pixels

	Alternativa A <pre> Repetir até a flor faça repita 4 vezes faça avance por 30 pixels vire à direita por 90 graus pule para a frente por 30 pixels </pre>	Alternativa B <pre> Repetir até a flor faça repita 4 vezes faça avance por 120 pixels vire à direita por 90 graus pule para a frente por 30 pixels </pre>
	Alternativa C <pre> Repetir até a flor faça repita 4 vezes faça avance por 30 pixels vire à direita por 90 graus pule para a frente por 210 pixels </pre>	Alternativa D <pre> Repetir até a flor faça repita 7 vezes faça avance por 30 pixels vire à direita por 90 graus pule para a frente por 30 pixels </pre>

QUESTÃO

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?

	Alternativa A	Alternativa B
	Alternativa C	Alternativa D

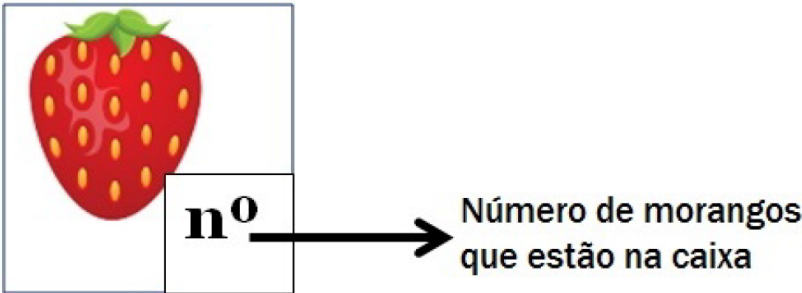
QUESTÃO 8

Qual sequência leva o "Pac-Man" até o fantasma pelo caminho indicado?

	Alternativa A	Alternativa B
	Alternativa C	Alternativa D

IMPORTANTE: LEIA COM ATENÇÃO

A IMAGEM ABAIXO IRÁ APARECER NAS PRÓXIMAS QUESTÕES. O NÚMERO QUE ESTÁ NO CANTO INFERIOR DIREITO INDICA QUANTOS MORANGOS EXISTEM NA CAIXA.



QUESTÃO 9

O que falta na seguinte sequência para levar o "Pac-Man" até o fantasma pelo caminho indicado?

repetir até chegar no se passar por ?? x3	Alternativa A
	Alternativa B
	Alternativa C
	Alternativa D Tanto a alternativa A como a alternativa C estão corretas

QUESTÃO 10

Qual sequência leva o "Pac-Man" pelo caminho indicado até os morangos? Lembre-se que o "Pac-Man" precisa comer o número de morangos indicado.

	Alternativa A <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>	Alternativa B <pre> enquanto houver caminho em frente faça avance repita 4 vezes faça coma 1 morango </pre>
	Alternativa C <pre> enquanto houver caminho em frente faça avance repita 5 vezes faça coma 1 morango </pre>	Alternativa D <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>

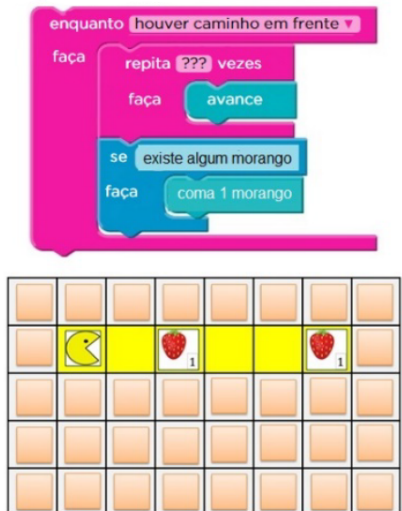
QUESTÃO 11

Qual sequência leva o "Pac-Man" pelo caminho indicado até os morangos? Lembre-se que o "Pac-Man" precisa comer o número de morangos indicado.

	Alternativa A <pre> enquanto houver caminho em frente faça repita 5 vezes faça avance repita 3 vezes faça coma 1 morango </pre>	Alternativa B <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>
	Alternativa C <pre> enquanto houver caminho em frente faça repita 3 vezes faça avance repita 5 vezes faça coma 1 morango </pre>	Alternativa D <pre> enquanto houver caminho em frente faça avance repita 3 vezes faça coma 1 morango </pre>

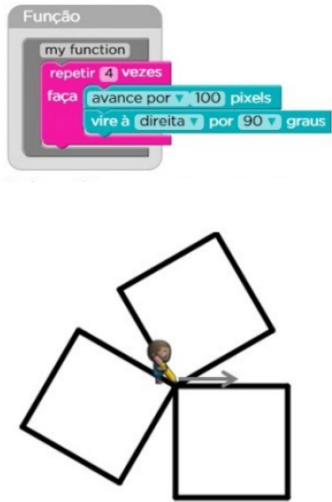
QUESTÃO 12

O que falta na seguinte sequência para que o "Pac-Man" avance pelo caminho assinalado comendo o número de morangos indicado?

 The code block is a 'while' loop with the condition 'houver caminho em frente'. Inside the loop, there is a 'repeat' block with '???' times, containing an 'advance' block. Below the loop is an 'if' block with the condition 'existe algum morango', containing a 'eat 1 mango' block. The maze grid is 8x8. Pac-Man is at (2,2). The path is marked with yellow squares: (2,2) to (2,3), (2,3) to (3,3), (3,3) to (3,4), (3,4) to (4,4), (4,4) to (4,5), (4,5) to (5,5), (5,5) to (5,6), (5,6) to (6,6), (6,6) to (6,7), (6,7) to (7,7), (7,7) to (7,8), (7,8) to (6,8), (6,8) to (5,8), (5,8) to (5,7), (5,7) to (5,6), (5,6) to (5,5), (5,5) to (5,4), (5,4) to (5,3), (5,3) to (5,2), (5,2) to (5,1), (5,1) to (4,1), (4,1) to (4,2), (4,2) to (4,3), (4,3) to (4,4), (4,4) to (4,5), (4,5) to (4,6), (4,6) to (4,7), (4,7) to (4,8), (4,8) to (3,8), (3,8) to (3,7), (3,7) to (3,6), (3,6) to (3,5), (3,5) to (3,4), (3,4) to (3,3), (3,3) to (3,2), (3,2) to (3,1), (3,1) to (2,1), (2,1) to (2,2). There are two strawberries at (3,4) and (5,6), each labeled with the number 1.	Alternativa A 1 vez
	Alternativa B 2 vezes
	Alternativa C 3 vezes
	Alternativa D 5 vezes

QUESTÃO 13

Se temos o seguinte conjunto de instruções, que chamamos de "my function", e que desenha um quadrado de 100 pixels de lado. Qual sequência o artista deve seguir para desenhá-la seguinte figura? Lembre-se que cada um dos lados do quadrado mede 100 pixels.

 The function block 'my function' contains: 'repetir 4 vezes' loop with 'avance por 100 pixels' and 'vire à direita por 90 graus'. The drawing shows four squares of side 100 pixels, each rotated 45 degrees relative to the others, meeting at a central point.	Alternativa A repetir 3 vezes faça my function vire à direita por 120 graus	Alternativa B repetir 3 vezes faça my function vire à direita por 120 graus
	Alternativa C repetir 4 vezes faça my function vire à direita por 90 graus	Alternativa D repetir 3 vezes faça my function vire à direita por 90 graus

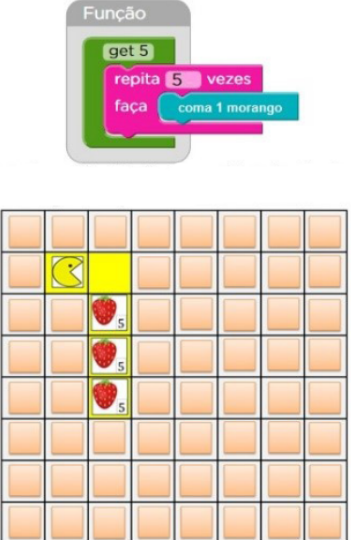
QUESTÃO 14

Se temos o seguinte conjunto de instruções, que chamamos de "my function", e que desenha um triângulo de 50 pixels de lado. O que falta na seguinte sequência para que o artista desenhe a seguinte figura? Lembre-se que cada um dos lados de cada triângulo mede 50 pixels.

 The 'my function' block contains: 'repetir 3 vezes' (repeat 3 times) containing 'avance por 50 pixels' (move forward 50 pixels) and 'vire à esquerda por 120 graus' (turn left 120 degrees). Below it, a 'repetir ??? vezes' (repeat ??? times) block contains 'my function' and 'pule para a frente por 50 pixels' (jump forward 50 pixels). The drawing shows a series of connected triangles on a horizontal line.	Alternativa A 15	Alternativa B 5
	Alternativa C 4	Alternativa D 3

QUESTÃO 15

Se temos o seguinte conjunto de instruções, que chamamos de "get 5". Qual sequência leva o "Pac-Man" pelo caminho indicado e faz com que ele coma o número de morangos correspondentes?

 The 'get 5' block contains: 'get 5' (get 5), 'repita 5 vezes' (repeat 5 times) containing 'coma 1 morango' (eat 1 strawberry). The maze is a 10x10 grid with Pac-Man at (2,2) and strawberries at (3,3), (3,4), and (3,5).	Alternativa A avance vire à direita 90 repita 3 vezes faça avance get 5	Alternativa B avance vire à direita 90 repita 3 vezes faça get 5 avance
	Alternativa C avance vire à direita 90 repita 5 vezes faça avance get 5	Alternativa D avance vire à direita 90 repita 5 vezes faça get 5 avance

ANEXO C - Termo de consentimento**TERMO DE AUTORIZAÇÃO****Participação no Projeto Conect +**

Eu, _____, portador da Cédula de Identidade RG nº _____, inscrito no CPF sob nº _____, responsável pelo(a) aluno(a) _____, matriculado(a) na ESCOLA ESTADUAL DE ENSINO FUNDAMENTAL MATHIAS LORENZON no ____ ano no período da manhã do Ensino Fundamental, autorizo sua participação no projeto Conect +, que ocorre no turno inverso ao escolar, promovido pela ARAPL INTELLIGENT MANUFACTURING e autorizado pela 15ª CRE-RS, nas dependências da ESCOLA ESTADUAL DE ENSINO FUNDAMENTAL MATHIAS LORENZON, durante o período letivo de março a dezembro de 2024.

Declaro estar ciente que ocorrerá a participação do(a) aluno(a) na pesquisa de doutorado no decorrer do projeto, realizada no contexto do Programa de Pós-Graduação em Educação da Universidade de Passo Fundo, sob a supervisão do Prof. Dr. Adriano Canabarro Teixeira. A pesquisa tem como título "A RELAÇÃO DO PENSAMENTO COMPUTACIONAL COM O DESENVOLVIMENTO DO ARGUMENTO DEDUTIVO EM CRIANÇAS", e reconheço que os dados coletados durante este estudo serão utilizados para subsidiar a tese de doutorado de Milene Giaretta (ministrante do projeto). Também estou ciente de que o sigilo dos dados será preservado, e que o nome dos participantes não será divulgado em qualquer publicação científica.

Declaro ainda estar ciente das atividades a serem desenvolvidas durante o projeto e concordo com as normas e regulamentos estabelecidos pela escola. Comprometo-me a acompanhar o progresso do aluno(a) durante sua participação no projeto, fornecendo o suporte necessário para o bom desempenho.

Autorizo também a escola e a empresa a utilizar imagens, vídeos ou áudios do aluno(a) em atividades relacionadas ao projeto escolar, bem como em materiais de divulgação institucional. Por fim, declaro estar ciente de que a participação do aluno(a) no projeto Conect + é voluntária e que, em caso de qualquer inconveniente ou desistência, comunicarei imediatamente à escola.

GETÚLIO VARGAS, _____ de março de 2024

Assinatura do Responsável: _____

Assinatura do Aluno: _____

Milene Giaretta

Cientista de Dados da ARAPL

Doutoranda do PPGEduc - UPF

CARINA DALL'AGNOL TRIQUES

Diretora da escola

WILLIAN DI DONATO

CEO/CTO da ARAPL

ANEXO D - Termo de autorização da escola***Escola Estadual de Ensino Fundamental Mathias Lorenzon***

Decreto de Criação nº 12658 de 28/09/61 – D.O 29/09/61

Portaria de Designação nº 00924 de 13/02/95 – D.O 17/02/95

CEP 99900 – 000 – Rua Domingos Chiarello, 97 – GETÚLIO VARGAS – RS

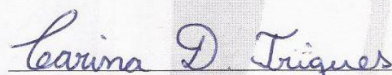
E-mail: mathiaslorenzon15cre@educar.rs.gov.br

Telefone: (54) 3341-3976

AUTORIZAÇÃO DA ESCOLA

Eu Carina Dall Agnol Triques, diretora da Escola Estadual de Ensino Fundamental Mathias Lorenzon autorizo a discente Milene Giaretta do Programa de Pós – Graduação em Educação- PPGEDU da Universidade de Passo Fundo, a realizar a pesquisa intitulada “A relação do pensamento computacional e o desenvolvimento do argumento dedutivo em crianças: um estudo sobre a programação em bloco”, no período de 02 de abril a 07 de maio de 2024.

Getúlio Vargas, 07 de maio de 2024.



Carina Dall Agnol Triques
Diretora

Carina Dall Agnol Triques
Diretora
PG.262 D.O. 03/01/2022
I.F. 1618229.01