



UNIVERSIDADE FEDERAL FLUMINENSE – UFF
INSTITUTO DE BIOLOGIA – IB
PROGRAMA DE PÓS-GRADUAÇÃO EM
CIÊNCIAS, TECNOLOGIAS E INCLUSÃO – PGCTIn

SANDRO MIRANDA DE REZENDE

**O DESENVOLVIMENTO DO PENSAMENTO
COMPUTACIONAL EM ESTUDANTES DO ENSINO MÉDIO
COM DEFICIÊNCIA VISUAL: CONSTRUINDO CÓDIGOS E
RESOLVENDO PROBLEMAS DE MATEMÁTICA ATRAVÉS
DE UMA LINGUAGEM ALGORÍTMICA**

ORIENTADOR: PROF. DR. SERGIO CRESPO COELHO DA SILVA PINTO

COORIENTADORA: PROF^a. DR^a. ANA ISABEL DE AZEVEDO SPINOLA DIAS



NITERÓI

2024

SANDRO MIRANDA DE REZENDE

**O DESENVOLVIMENTO DO PENSAMENTO
COMPUTACIONAL EM ESTUDANTES COM DEFICIÊNCIA
VISUAL DO ENSINO MÉDIO: CONSTRUINDO CÓDIGOS E
RESOLVENDO PROBLEMAS DE MATEMÁTICA ATRAVÉS
DE UMA LINGUAGEM ALGORÍTMICA**

Tese apresentada ao Programa de Pós-graduação em Ciências, Tecnologias e Inclusão – PGCTIn, da Universidade Federal Fluminense – UFF, como requisito para obtenção do título de Doutor em Ciências, Tecnologias e Inclusão.

Orientação: Prof. Dr. Sergio Crespo Coelho da Silva Pinto

Coorientação: Prof^a. Dra^a. Ana Isabel de Azevedo Spinola Dias

NITERÓI

2024

Ficha catalográfica automática - SDC/BCV
Gerada com informações fornecidas pelo autor

R467d Rezende, Sandro Miranda de
O desenvolvimento do pensamento computacional em estudantes do Ensino Médio com deficiência visual : construindo códigos e resolvendo problemas de Matemática através de uma linguagem algorítmica / Sandro Miranda de Rezende. - 2024. 312 f.

Orientador: Sergio Crespo Coelho da Silva Pinto.
Coorientador: Ana Isabel de Azevedo Spinola Dias.
Tese (doutorado)-Universidade Federal Fluminense, Instituto de Biologia, Niterói, 2024.

1. Deficiência visual. 2. Pensamento computacional. 3. Resolução de problemas. 4. Linguagem de programação algorítmica. 5. Produção intelectual. I. Pinto, Sergio Crespo Coelho da Silva, orientador. II. Dias, Ana Isabel de Azevedo Spinola, coorientadora. III. Universidade Federal Fluminense. Instituto de Biologia. IV. Título.

CDD - XXX

Bibliotecário responsável: Debora do Nascimento - CRB7/6368

SANDRO MIRANDA DE REZENDE

**O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL EM
ESTUDANTES DO ENSINO MÉDIO COM DEFICIÊNCIA VISUAL:
CONSTRUINDO CÓDIGOS E RESOLVENDO PROBLEMAS DE
MATEMÁTICA ATRAVÉS DE UMA LINGUAGEM ALGORÍTMICA**

Tese apresentada ao Programa de Pós-graduação em Ciências, Tecnologias e Inclusão – PGCTIn, da Universidade Federal Fluminense – UFF, como requisito para obtenção do título de Doutor em Ciências, Tecnologias e Inclusão.

BANCA EXAMINADORA

Dr. Clifton Clunie - Universidade Tecnológica do Panamá

Dr. Crediné Silva de Menezes - Universidade Federal do Rio Grande do Sul

Dr. Henrique Nou Schneider - Universidade Federal de Sergipe

Dr^a. Claudiane Figueiredo Ribeiro - FAETEC-RJ

Dr^a. Ruth Maria Mariani Braz - PGCTIn - Universidade Federal Fluminense

Dr. Paulo Pires de Queiroz - PGCTIn - Universidade Federal Fluminense

**Dr. Sérgio Crespo Coelho da Silva Pinto - PGCTIn - Universidade Federal Fluminense -
(Orientador/Presidente)**

Dr^a. Ana Isabel de Azevedo Spinola Dias - Universidade Federal Fluminense - (Coorientadora)

Dr^a Suzete Araújo – PGCTIn – Universidade Federal Fluminense (Suplente)

Dr^a Graziela Ferreira Guarda – Mackenzie – SP (Suplente)

AGRADECIMENTOS

À Pró-Reitoria de Pós-graduação, Pesquisa, Extensão e Cultura (PROPGPEC) do Colégio Pedro II, que autorizou a realização da pesquisa na instituição. Aos profissionais dos Napnes dos diferentes campi da escola, pelas trocas; em especial, aos professores, técnicos e profissionais de apoio escolar que atuaram e atuam nos Napnes nos quais o estudo de campo foi conduzido, pela disponibilidade e apoio. Aos estudantes que concordaram em participar da pesquisa, pela valiosa contribuição a este trabalho.

Ao meu orientador, professor Sérgio Crespo, que me acolheu, confiou no meu caminhar e me deu liberdade para conduzir a pesquisa a meu modo, sendo, ainda assim, decisivo para que o trabalho seguisse o caminho adequado. À minha co-orientadora, professora Ana Isabel Dias, que com sua sutileza, seu olhar cuidadoso, seu constante incentivo e pontuações sempre pertinentes, foi fundamental para a conclusão deste estudo.

Aos professores Clifton Clunie, Ruth Mariani, Claudiane Figueiredo e Crediné Menezes, pelas importantes contribuições na banca de qualificação, momento decisivo para a determinação dos rumos da pesquisa.

Aos colegas que, ao longo desse percurso, passaram pelo grupo de pesquisa TeCEADI+ e contribuíram, de alguma forma, para esta produção. Em especial, à Graziela Guarda, elétrica, vibrante, sempre presente e disposta a auxiliar nos momentos de incertezas e crises.

Aos colegas de trabalho do Napne do campus São Cristóvão II do Colégio Pedro II, do passado e do presente, que tanto me ensinaram e ensinam sobre inclusão. Em especial à Adriane Farah que, incansável, construiu as bases para o que somos hoje enquanto núcleo e transformou minha compreensão do que é Educação.

Aos colegas de trabalho do CIEP 405 Ministro Santiago Dantas que, comigo, seguem acreditando no poder transformador da Educação. Em especial à Joyce Pereira, parceira de feiras de ciências, olimpíadas, aulas-passeio ou qualquer projeto que, por mais desgastante que seja, no final (e no trajeto) vai nos encher de alegria e de certeza que estamos no bom caminho.

Aos estudantes do Colégio Pedro II e do CIEP 405 Ministro Santiago Dantas, do antes e do agora, por me ensinarem tanto a todo tempo. Como Paulo Freire, não tenho dúvidas que *“quem ensina aprende ao ensinar e quem aprende ensina ao aprender”*.

Por fim, a todos que, direta ou indiretamente, contribuíram nos bastidores desta jornada. A minha mãe, pelo cuidado, pelo zelo, por tanta luta. Aos amigos, por estarem presentes nos momentos em que pausas eram necessárias. E finalmente, ao meu marido, Bruno Barros, pelo ouvido atento; pelas sugestões; pela companhia durante as noites de escrita; pela compreensão, sensibilidade e suporte nos momentos de bloqueio e insegurança; por me ajudar a seguir este percurso com sanidade e leveza; por ser alicerce e incentivo, sempre.

SUMÁRIO

1. INTRODUÇÃO	17
1.1. TRAJETÓRIA ACADÊMICA E PROFISSIONAL.....	17
1.2. JUSTIFICATIVA, PROBLEMA DE PESQUISA E HIPÓTESES DE TRABALHO.....	19
2. OBJETIVOS.....	25
2.1. OBJETIVO GERAL.....	25
2.2. OBJETIVOS ESPECÍFICOS	25
3. REFERENCIAL TEÓRICO.....	26
3.1. DEFICIÊNCIA VISUAL E EDUCAÇÃO MATEMÁTICA.....	26
3.2. PENSAMENTO COMPUTACIONAL.....	32
3.3. LINGUAGENS ALGORÍTMICAS EM PORTUGUÊS ESTRUTURADO	39
4. PENSAMENTO COMPUTACIONAL E DEFICIÊNCIA VISUAL NA EDUCAÇÃO BÁSICA: ESTADO DA ARTE E TRABALHOS CORRELATOS.....	49
5. CAMINHOS METODOLÓGICOS	85
5.1. TIPO DE PESQUISA E ABORDAGEM.....	85
5.2. DESENHO DA PESQUISA.....	86
5.3. DESCRIÇÃO DOS PARTICIPANTES DA PESQUISA.....	87
5.4. LEVANTAMENTO DE DADOS.....	89
5.5. CONSIDERAÇÕES ÉTICAS.....	89
5.6. REFERENCIAL METODOLÓGICO.....	90
5.6.1. ENGENHARIA DIDÁTICA	90
5.6.2. ANÁLISE DE CONTEÚDO.....	95
6. RESULTADOS E DISCUSSÃO.....	98
6.1. DESENVOLVIMENTO DA ENGENHARIA DIDÁTICA.....	98
6.1.1. ANÁLISES PRELIMINARES.....	98
6.1.2. A FASE DE CONCEPÇÃO E ANÁLISE A PRIORI	112
6.1.3. EXPERIMENTAÇÃO	144
6.1.4. ANÁLISE A POSTERIORI.....	146
6.2. A ANÁLISE DE CONTEÚDO	199
6.2.1. ANÁLISE LEXICOGRÁFICA NO IRAMUTEQ.....	202
6.2.2. ANÁLISE CATEGORIAL	217
7. CONSIDERAÇÕES FINAIS.....	261
REFERÊNCIAS.....	267
APÊNDICES E ANEXOS.....	281

LISTA DE ABREVIATURAS E SIGLAS

Alerj - Assembleia Legislativa do Estado do Rio de Janeiro

BNCC - Base Nacional Comum Curricular

CAS - Computing at School

CEDERJ - Centro de Educação Superior a Distância do Estado do Rio de Janeiro

CEFET-RJ - Centro Federal de Educação Tecnológica Celso Suckow da Fonseca

CHD - Classificação Hierárquica Descendente

CIEP - Centro Integrado de Educação Pública

CSTA - Computer Science Teachers Association

DOS - Disk Operating System

EaD - Educação a Distância

GPL - General Public License

IB - Instituto de Biologia

IFRJ - Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro

ILA - Interpretador de Linguagem Algorítmica

IMPA - Instituto de Matemática Pura e Aplicada

INPI - Instituto Nacional de Propriedade Industrial

IRaMuTeQ - Interface de R pour les Analyses Multidimensionnelles de Textes et de Questionnaires

ISTE - International Society for Technology in Education

LBI - Lei Brasileira de Inclusão da Pessoa com Deficiência

LDB - Lei de Diretrizes e Bases da Educação Nacional

Napne - Núcleo de Atendimento a Pessoas com Necessidades Específicas

NFC - Near Field Communication

NTEM - Novas Tecnologias no Ensino da Matemática

PC - Pensamento computacional

PGCTIn - Programa de Pós-graduação em Ciências, Tecnologias e Inclusão

PNEE - Política Nacional de Educação Especial na Perspectiva da Educação Inclusiva

RELME - Reunión Latinoamericana de Matemática Educativa

STEM - Science, Technology, Engineering and Mathematics

TAE - Técnico em Assuntos Educacionais

TALE - Termo de Assentimento Livre e Esclarecido

TCLE - Termo de Consentimento Livre e Esclarecido

TDICs - Tecnologias Digitais de Informação e Comunicação

UNISINOS - Universidade do Vale do Rio dos Sinos

UNIVALI - Universidade do Vale do Itajaí

UFF - Universidade Federal Fluminense

UFRJ - Universidade Federal do Rio de Janeiro

UTF-8 - UCS Transformation Format 8

LISTA DE FIGURAS

Figura 1: <i>ACCembly</i> e seus componentes.	58
Figura 2: <i>KIBO</i> e blocos manipuláveis de madeira.....	58
Figura 3: Peças do Torino e hub principal.....	59
Figura 4: Principais elementos do <i>TACTOPI</i>	60
Figura 5: Blocos utilizados no protótipo inicial do <i>StoryBlocks</i> , incluindo personagens, ações e estruturas de controle.....	61
Figura 6: (A) Representação do <i>iCETA</i> com fone, computador, espelho, área de trabalho e blocos. (B) Caixa com blocos representando os números de 1 a 5.....	62
Figura 7: Criação de uma melodia através do <i>CodeRhythm</i>	63
Figura 8: Usuário realizando a leitura de uma sequência de blocos.	64
Figura 9: Exemplo de programa criado através do <i>Bonk</i>	64
Figura 10: Robô Dash e Dot, seu ajudante.	65
Figura 11: Robô <i>Bee-Bot</i>	66
Figura 12: Robô <i>Blue-Bot</i>	66
Figura 13: Robô <i>Ozobot</i> se movendo sobre uma linha.	67
Figura 14: <i>Super DOC</i>	67
Figura 15: Robôs criados com o <i>LEGO Mindstorms</i>	68
Figura 16: Visão geral do <i>LEGOWorld</i> com smartphone, tripé, mapa físico, blocos tangíveis e caixa com tampa.	69
Figura 17: Blocos de código de um jogo construído através do <i>Blockly</i>	70
Figura 18: <i>Osmo Coding Awbie</i>	70
Figura 19: Leitura de uma sequência de blocos através de um celular.....	71
Figura 20: <i>Puzzlets</i>	72
Figura 21: Percentual de trabalhos analisados cujas propostas promovem o desenvolvimento de cada conceito do pensamento computacional (formulação de Brennan e Resnick, 2012).....	82
Figura 22: Percentual de trabalhos analisados cujas propostas promovem o desenvolvimento de cada habilidade do pensamento computacional (formulação de Czismadia et al., 2015).....	83
Figura 23: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 1.....	113
Figura 24: Momentos 1 e 2 do encontro 1.....	114

Figura 25: Momento 3 do encontro 1.	115
Figura 26: Momentos 4 e 5 do encontro 1.....	118
Figura 27: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 2.....	119
Figura 28: Momentos 1 e 2 do encontro 2.....	120
Figura 29: Momento 3 do encontro 2.	122
Figura 30: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 3.....	124
Figura 31: Momentos 1, 2 e 3 do encontro 3.....	125
Figura 32: Momento 4 do encontro 3.	126
Figura 33: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 4.....	127
Figura 34: Momentos 1 e 2 do encontro 4.....	128
Figura 35: Segunda parte do momento 2 do encontro 4.	129
Figura 36: Momento 3 do encontro 4.	131
Figura 37: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 5.....	133
Figura 38: Momento 1 do encontro 5.	134
Figura 39: Segunda parte do momento 1 do encontro 5.	135
Figura 40: Momento 3 do encontro 5.	138
Figura 41: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 6.....	139
Figura 42: Momento 1 do encontro 6.	140
Figura 43: Momento 2 do encontro 6.	140
Figura 44: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 7.....	142
Figura 45: Momento 1 do encontro 7.	142
Figura 46: Momento 2 do encontro 7.	143
Figura 47: Nuvem de palavras do corpus textual.	204
Figura 48: Resultado da análise de similitude.....	207
Figura 49: : Dados quantitativos da Classificação Hierárquica Descendente no IRaMuTeQ.....	210
Figura 50: Dendograma da Classificação Hierárquica Descendente.	211
Figura 51: Análise fatorial de correspondência da CHD.....	217

Figura 52: Nuvem de palavras – Grupo 1	306
Figura 53: Nuvem de palavras – Grupo 2	306
Figura 54: Nuvem de palavras – Grupo 3	307
Figura 55: Nuvem de palavras – Grupo 4	307
Figura 56 – Nuvem de palavras – Encontro 1	308
Figura 57: Nuvem de palavras – Encontro 2	308
Figura 58: Nuvem de palavras – Encontro 3	309
Figura 59: Nuvem de palavras – Encontro 4	309
Figura 60: Nuvem de palavras – Encontro 5	310
Figura 61: Nuvem de palavras - Encontro 6	310
Figura 62: Nuvem de palavras – Encontro 7	311
Figura 63: Nuvem de palavras – Encontro 8	311

LISTA DE QUADROS

Quadro 1: Comparativo entre códigos de um mesmo programa construídos em Portugol Studio, G-Portugol, VisuAlg, Portugol IDE, Portugol Viana e ILA.	46
Quadro 2: Diferenças na estrutura das linguagens Portugol Studio, G-Portugol, VisuAlg, Portugol IDE, Portugol Viana e ILA.	47
Quadro 3: Passos da revisão sistemática.	50
Quadro 4: Protocolo para a revisão sistemática de literatura.	51
Quadro 5: Trabalhos selecionados na revisão sistemática.	54
Quadro 6: Recursos utilizados para o desenvolvimento do pensamento computacional, identificados nos trabalhos analisados.	56
Quadro 7: Objetivo e participantes de cada pesquisa.	74
Quadro 8: Como o pensamento computacional é abordado em cada artigo.	77
Quadro 9: Habilidades/conceitos do pensamento computacional desenvolvidos nos trabalhos analisados.	81
Quadro 10: Participantes da pesquisa.	88
Quadro 11: Habilidades da BNCC que envolvem o conceito de função.	107
Quadro 12: Perguntas elaboradas e ações esperadas em relação ao programa 1.	116
Quadro 13: Perguntas elaboradas e ações esperadas em relação ao programa 2.	116
Quadro 14: Avaliação do encontro 1 e produção esperada dos participantes. ...	118
Quadro 15: Operadores e operações associadas no ILA.	120
Quadro 16: Perguntas elaboradas e ações esperadas em relação ao programa.	121
Quadro 17: Perguntas elaboradas em relação a mudanças no programa proposto e ações esperadas.	122
Quadro 18: Avaliação do encontro 2 e produção esperada dos participantes. ...	123
Quadro 19: Algoritmo esperado dos participantes.	125
Quadro 20: Avaliação do encontro 3 e produção esperada dos participantes. ...	126
Quadro 21: Perguntas em relação ao programa e ações esperadas.	128
Quadro 22: Perguntas em relação ao segundo programa e ações esperadas. .	130
Quadro 23: Desdobramentos possíveis da terceira pergunta e ações a serem realizadas.	130

Quadro 24: Avaliação do encontro 4 e produção esperada dos participantes. ..	132
Quadro 25: Perguntas elaboradas e ações esperadas dos participantes.	134
Quadro 26: Perguntas elaboradas e ações esperadas dos participantes.	135
Quadro 27: Funcionamento do novo programa e produção esperada dos participantes.	137
Quadro 28: Avaliação do encontro 5 e produção esperada dos participantes. ..	138
Quadro 29: Situações problema e produção esperada dos participantes.	141
Quadro 30: Situações problema e produção esperada dos participantes	144
Quadro 31: Datas de realização dos encontros com cada grupo de participantes, e frequência dos estudantes em cada encontro (P: presente; A: ausente).	145
Quadro 32: Análise a posteriori do momento 3 do encontro 1 (primeiro programa).	148
Quadro 33: Análise a posteriori do momento 3 do encontro 1 (segundo programa).	150
Quadro 34: Análise a posteriori da avaliação do encontro 1.	152
Quadro 35: Análise a posteriori da primeira parte do momento 2 do encontro 2.	154
Quadro 36: Análise a posteriori da segunda parte do momento 2 do encontro 2.	158
Quadro 37: Análise a posteriori da avaliação do encontro 2.	160
Quadro 38: Análise a posteriori dos algoritmos elaborados pelos participantes.	162
Quadro 39: Análise a posteriori da avaliação do Encontro 3.	163
Quadro 40: Análise a posteriori do momento 2 do Encontro 4.	164
Quadro 41: Análise a posteriori da segunda parte do momento 2 do Encontro 4.	168
Quadro 42: Análise a posteriori da avaliação do Encontro 4.	172
Quadro 43: Análise a posteriori do momento 1 do Encontro 5.	173
Quadro 44: Análise a posteriori da segunda parte do momento 1 do Encontro 5.	174
Quadro 45: Análise a posteriori das alterações realizadas pelos participantes. ..	177
Quadro 46: Análise a posteriori da avaliação do Encontro 5.	179
Quadro 47: Análise a posteriori da primeira situação problema do Encontro 6. ..	184
Quadro 48: Análise a posteriori da segunda situação problema do Encontro 6.	188
Quadro 49: Análise a posteriori da primeira situação problema do Encontro 7. ..	192

Quadro 50: Análise a posteriori da segunda situação problema do Encontro 7.	196
Quadro 51: Quadro resumo com principais dificuldades e problemas apresentados pelos participantes nos encontros.	197
Quadro 52: Quadro resumo com principais potencialidades quanto ao uso do ILA identificadas ao longo dos encontros.	198
Quadro 53: Codificação dos textos no IRaMuTeQ.	203
Quadro 54: Categorias identificadas através da análise de conteúdo.....	219

RESUMO

Os avanços tecnológicos vêm implicando uma série de mudanças na organização econômica e cultural da sociedade, o que, em consequência, tem impacto nas dinâmicas que se dão no espaço escolar. Em um contexto histórico-social marcado pela cultura digital, a capacidade de lidar com novas tecnologias e de resolver problemas de forma criativa se torna essencial. Nesse cenário, o desenvolvimento de habilidades do pensamento computacional vem ganhando cada vez mais destaque no currículo escolar. É preciso, no entanto, que esta integração se dê de forma inclusiva, respeitando e valorizando as diferenças entre os estudantes. O objetivo desta pesquisa é analisar as potencialidades e limitações da utilização de uma linguagem algorítmica em português estruturado no desenvolvimento do pensamento computacional em estudantes com deficiência visual do Ensino Médio. Foi realizada uma pesquisa exploratória com abordagem qualitativa, conduzida junto a sete estudantes do Ensino Médio com deficiência visual de diferentes campi do Colégio Pedro II, instituição da rede federal de ensino no Rio de Janeiro. Para elaboração, condução e avaliação da sequência didática, utilizou-se como referencial metodológico os pressupostos da Engenharia Didática. De maneira complementar, foram feitas análise lexicográfica e análise de conteúdo das transcrições dos áudios dos encontros, com o objetivo de identificar padrões e realizar inferências a partir das falas dos participantes. Os resultados apontam que o uso de uma linguagem algorítmica baseada em português estruturado contribuiu para o desenvolvimento de habilidades do pensamento computacional entre os participantes. Por outro lado, fatores como uso do teclado, localização de elementos na tela do computador e uso do leitor de tela podem representar desafios no processo de construção dos códigos. Aspectos conceituais relacionados à sintaxe da linguagem algorítmica e à mobilização de conceitos matemáticos também devem ser considerados, assim como fatores relacionados à organização da sequência didática e às emoções manifestadas pelos estudantes. O estudo realizado contribui para a construção de cenários de aprendizagem inclusivos, onde estudantes videntes e não videntes podem trabalhar colaborativamente utilizando os mesmos recursos para a modelagem e resolução de situações-problema.

PALAVRAS-CHAVE: Deficiência visual. Pensamento computacional. Resolução de problemas. Linguagem de programação algorítmica.

ABSTRACT

Technological advances have been implying a series of changes in the economic and cultural organization of society, which, as a result, has an impact on the dynamics that take place in the school. In a historical-social context characterized by digital culture, the ability to deal with new technologies and solve problems creatively becomes essential. In this scenario, the development of computational thinking skills is gaining more and more prominence in the school curriculum. However, this integration must take place in an inclusive way, respecting and valuing the differences between students. The objective of this research is to analyze the potential and limitations of using an algorithmic language in structured Portuguese in the development of computational thinking in high school students with visual impairments. An exploratory research was carried out with a qualitative approach, conducted with seven visually impaired high school students from different campuses of Colégio Pedro II, Rio de Janeiro. To prepare, conduct and evaluate the didactic sequence, the assumptions of Didactic Engineering were used as a methodological reference. In a complementary manner, lexicographic analysis and content analysis of the audio transcriptions of the meetings were carried out, with the aim of identifying patterns and making inferences based on the participants' statements. The results indicate that the use of an algorithmic language based on structured Portuguese contributed to the development of computational thinking skills among the participants. On the other hand, factors such as keyboard use, location of elements on the computer screen and use of a screen reader can represent challenges in the code construction process. Conceptual aspects related to the syntax of the algorithmic language and the mobilization of mathematical concepts must also be considered, as well as factors related to the organization of the didactic sequence and the emotions expressed by students. **Conclusion:** The study carried out contributes to the construction of inclusive learning scenarios, where sighted and non-sighted students can work collaboratively using the same resources to model and resolve problem situations.

KEYWORDS: Visual impairments. Computational thinking. Problem solving. Algorithmic programming language.

1. INTRODUÇÃO

1.1. TRAJETÓRIA ACADÊMICA E PROFISSIONAL

O ingresso no Programa de Pós-graduação em Ciências, Tecnologias e Inclusão (PGCTIn) da Universidade Federal Fluminense (UFF) é resultado de uma série de vivências que, sucessiva ou paralelamente, contribuíram para que me aproximasse da pesquisa acadêmica, da Educação Matemática e da Educação Especial. Minha trajetória universitária teve início em 2004, quando me tornei aluno do curso de Engenharia de Produção da Universidade Federal do Rio de Janeiro (UFRJ). Um ano depois, iniciei minha trajetória profissional atuando como monitor de uma instituição privada de ensino. Motivado pela rica vivência na escola durante os anos seguintes, em 2007 iniciei o curso de Licenciatura em Matemática na Universidade Federal Fluminense, na modalidade Educação a Distância (EaD). Em 2008, fui aprovado em um concurso para o cargo de assistente administrativo no Centro Federal de Educação Tecnológica Celso Suckow da Fonseca (CEFET-RJ), e optei por realizar o desligamento do curso de Engenharia de Produção. No ano seguinte, obtive aprovação em concurso para o cargo de Técnico em Propriedade Industrial no Instituto Nacional de Propriedade Industrial (INPI), onde atuei por curto período.

Em 2010, objetivando voltar a trabalhar com Educação, realizei novo concurso e passei a ocupar o cargo de assistente administrativo no Instituto Federal de Educação, Ciência e Tecnologia do Rio de Janeiro (IFRJ), onde atuava como secretário escolar. No mesmo ano concluí o curso de Licenciatura em Matemática, e nos anos seguintes atuei como professor de matemática do estado do Rio de Janeiro e do município de Nova Iguaçu. Em 2012, me tornei tutor presencial do Centro de Educação Superior a Distância do Estado do Rio de Janeiro (CEDERJ), onde trabalhei com diversas disciplinas de cursos de graduação ofertados na modalidade EaD por universidades públicas do Rio de Janeiro.

Em 2013 ingressei no Mestrado Profissional em Matemática do Instituto de Matemática Pura e Aplicada (IMPA). No mesmo ano, fui selecionado como bolsista da Fundación Carolina para o Mestrado em Física e Matemática da Universidade de Granada, na Espanha. Assim, decidi realizar o trancamento do curso no IMPA e cursar o mestrado na universidade espanhola, onde optei pela especialidade em Biomatemática, tendo realizado um trabalho de pesquisa sobre a dinâmica populacional

de uma espécie de lagarta em um modelo presa-predador. A vivência durante esse período foi de extrema importância para o crescimento pessoal, acadêmico e profissional, na medida em que pude ter contato com estudantes e pesquisadores de diferentes áreas, países e culturas.

Após retornar ao Brasil, em 2014, iniciei o curso de especialização em Novas Tecnologias no Ensino da Matemática (NTEM) na Universidade Federal Fluminense, que contribuiu de maneira significativa para despertar meu interesse pela Educação Matemática. No curso, atentei especialmente para a Teoria dos Registros de Representação Semiótica (Duval, 1993, 2006, 2017), referencial teórico utilizado para a realização do trabalho de conclusão de curso, no qual investiguei o processo de ensino e aprendizagem dos números reais a partir de diferentes registros de representação semiótica.

Em 2016, me tornei professor de matemática da rede municipal de ensino de Duque de Caxias, onde atuo até então com os anos finais do Ensino Fundamental. Paralelamente, assumi o cargo de Técnico em Assuntos Educacionais (TAE) no Colégio Pedro II, campus São Cristóvão, onde também sigo atuando. Nesta escola, trabalho no Núcleo de Atendimento a Pessoas com Necessidades Específicas (Napne¹), setor responsável pelo atendimento educacional especializado a alunos que apresentem necessidades educacionais específicas de caráter permanente ou temporário. O interesse em Educação Especial e inclusão passou a crescer desde então, assim como a necessidade de uma formação específica na área e o desejo de realizar um trabalho de pesquisa sistematizado relacionado ao tema.

Objetivando retomar a proximidade com a academia, passei a participar de uma série de eventos científicos nas áreas de Educação Matemática e Educação Inclusiva, tendo trabalhos apresentados em eventos internacionais, entre eles o relato “*Creación de juegos didácticos como herramienta pedagógica para un aprendizaje activo: un relato de vivencia con alumnos de una escuela pública en Río de Janeiro*”, na 33^a Reunión Latinoamericana de Matemática Educativa (RELME), realizada em Havana, Cuba, em

¹ O Núcleo de Atendimento a Pessoas com Necessidades Específicas (Napne) é, na instituição, o núcleo responsável por desenvolver estratégias pedagógicas específicas e mediar o processo educacional, tanto dos estudantes que são público-alvo da Educação Especial, quanto daqueles que em algum momento apresentem necessidades educacionais específicas, sejam elas de natureza permanente ou temporária.

julho de 2019. Também atuei como orientador ou coorientador de projetos desenvolvidos junto a estudantes do CIEP Brizolão Municipalizado 405, escola onde leciono no município de Duque de Caxias, com o objetivo de incentivar a produção científica dos estudantes, o uso de tecnologias digitais e o debate de questões sociais. Ao longo dos últimos anos, uma série de projetos foram selecionados para apresentação em feiras municipais e estaduais de Ciência, Tecnologia e Inovação. Em 2019, fui um dos vencedores do *Prêmio Paulo Freire*, iniciativa da Comissão de Educação da Assembleia Legislativa do Estado do Rio de Janeiro (Alerj) que premiou projetos inovadores desenvolvidos por profissionais da Educação em escolas e universidades públicas do Rio de Janeiro.

Atualmente, sigo trabalhando na rede municipal de Duque de Caxias e no Colégio Pedro II. Nesta última instituição, realizo atendimento a estudantes que apresentam as mais diversas especificidades: transtorno do espectro autista, dislexia, deficiência intelectual, deficiência visual, transtorno do déficit de atenção com hiperatividade, altas habilidades, além de estudantes com dificuldades de aprendizagem. No entanto, tendo atuado com diversos alunos cegos no atendimento educacional especializado, na sala de aula regular e na experiência como tutor presencial na EaD, desenvolvi particular interesse em relação às especificidades do trabalho com esse grupo de alunos. Desse modo, passei a direcionar meu interesse, em uma perspectiva acadêmica, à aprendizagem da matemática por estudantes com deficiência visual, especialmente no que se refere ao seu processo de resolução de problemas matemáticos e as potencialidades das tecnologias digitais nesse contexto.

1.2. JUSTIFICATIVA, PROBLEMA DE PESQUISA E HIPÓTESES DE TRABALHO

Uma inquietude particular que surgiu a partir da vivência com estudantes com deficiência visual foi o fato de que muitas vezes os mesmos não apresentavam de maneira estruturada os procedimentos realizados para solucionar os problemas propostos. Em geral, estes estudantes preferiam apresentar os resultados obtidos e explicitar as estratégias e procedimentos colocados em prática na resolução através de explicações orais. Quando estimulados a descreverem as justificativas para as soluções através de instrumentos para escrita em Braille ou através de tecnologias digitais, comumente o faziam de forma desorganizada, embora tivessem chegado à resolução do

problema. Este cenário pode ser resultado de uma dificuldade enfrentada por pessoas cegas para criar e seguir uma cadeia de desenvolvimentos, como se dá na resolução de uma equação, por exemplo. Dessa forma, tais alunos tendem a construir alternativas para evitar tais dificuldades, como uso do raciocínio lógico em lugar do cálculo simbólico para se chegar à resolução do problema (Figueiras; Healy; Skovsmose, 2016).

Tal fato contribuiu para que eu pudesse construir um novo olhar em relação à pluralidade de caminhos possíveis no processo de resolução de um problema matemático e na comunicação de seus resultados. Muitas vezes, espera-se que estudantes sigam um determinado percurso argumentativo na apresentação de uma solução, entendido como mais correto ou elegante. No entanto, diferentes estratégias e recursos podem ser empregados na resolução de um mesmo problema, bem como seus resultados podem ser apresentados de diferentes formas. Sem diminuir a importância do texto escrito e da linguagem matemática formal, é importante refletir acerca dos diferentes modos de representação e das diferentes linguagens que podem ser empregadas na comunicação de ideias matemáticas. Supervalorizando o poder da matemática formal em detrimento de soluções alternativas, pode-se inibir a criatividade do estudante e limitar sua capacidade de relacionar problemas com situações reais, o que, por sua vez, pode refletir negativamente em sua confiança, em sua intuição e em seu raciocínio matemático (D'Ambrosio, 1989).

Muitos professores encaram a Matemática como um corpo de conhecimentos acabado, polido, não cabendo ao discente a oportunidade de criação e não havendo espaço para o debate de possibilidades alternativas. O estudante, assim, é levado a acreditar que, na aula de matemática, o seu papel é passivo e desinteressante (D'Ambrosio, 1989). Tal insistência na percepção da Matemática enquanto disciplina rígida, marcada pelo rigor e pelo formalismo, pode contribuir para a resistência e um progressivo distanciamento dos estudantes em relação à disciplina. Esse fato, somado à falta de situações práticas e ativas de aprendizagem, pode ser responsável pela diminuição do interesse e conseqüente baixa representatividade de pessoas com deficiência visual nas áreas e profissões relacionadas a STEM (*Science, Technology, Engineering and Mathematics*) (Bell; Silverman, 2018; Supalo; Isaccson; Lombardi, 2014; Lewis; Bodner, 2013).

Um dado curioso, no entanto, é que muitos adolescentes e adultos cegos relatam que, no passado, tinham interesse e afeição por áreas relacionadas a STEM. No entanto,

tal afeição diminui ao longo do tempo em virtude de barreiras de diferentes tipos enfrentadas no decorrer da vida acadêmica, em especial relacionadas à aprendizagem da Matemática. Isso, por vezes, acaba se tornando um fator determinante para essas pessoas optarem por continuar seus estudos em outras áreas (Bell; Silverman, 2018; Supalo; Isaccson; Lombardi, 2014). Dessa forma, para que este fato não siga ocorrendo é importante a construção de situações de aprendizagem acessíveis, bem como o emprego de recursos e ferramentas pedagógicas que promovam o engajamento e estimulem o interesse de estudantes com deficiência visual em relação às áreas STEM.

Uma forma de estimular tal interesse é através da elaboração de situações de aprendizagem desafiadoras, que promovam uma postura ativa e investigativa do estudante. Figueiras e Arcavi (2015) argumentam, por exemplo, que na ausência de visão as pessoas lidam com tarefas matemáticas envolvendo uma rica variedade de recursos e processos de raciocínio matemático. No entanto, muitas vezes tais recursos e potencialidades são subestimados pelo fato de as situações de aprendizagem elaboradas para estudantes com deficiência visual serem planejadas, na verdade, como se fossem destinadas a estudantes com dificuldades de aprendizagem. Esse fato frequentemente acaba tendo como consequência uma baixa expectativa em relação ao potencial dos estudantes, o que leva ao oferecimento de oportunidades limitadas para o envolvimento em práticas matemáticas mais desafiadoras (Figueiras; Healy; Skovsmose, 2016).

Neste cenário impregnado de novas vivências, leituras e olhares em relação à Educação Especial e, mais especificamente, em relação à Educação Matemática de estudantes com deficiência visual, as inquietudes que inicialmente se colocavam se transformaram. Se anteriormente o que me chamava a atenção eram os mecanismos movidos por estes estudantes para responder a problemas antigos, passou a me instigar a valorização de novas formas de resolver problemas. Formas irreverentes, disruptivas, que fujam do tradicional e dialoguem de maneira mais próxima com a cultura digital na qual estamos inseridos. Em particular, as possibilidades de aliar resolução de problemas, modelagem matemática e programação de computadores no contexto da Educação Básica passaram a me parecer convidativas. Passei, então, a me interessar por linguagens de programação, pelo pensamento computacional (PC) e, mais especificamente, por questões relacionadas a acessibilidade e inclusão neste contexto.

De maneira geral, entre os diferentes tipos de ambientes e linguagens que podem ser utilizados para uma introdução à programação, aqueles baseados em blocos apresentam vantagens relevantes. Nestes ambientes, trechos de códigos previamente definidos são visualmente representados como objetos que podem ser arrastados, soltos, encaixados ou clicados duas vezes para ativar. Devido à aparência de bloco desses objetos, em geral tais ambientes e linguagens são considerados “baseados em blocos”. Como nesses ambientes não há necessidade de utilizar elementos textuais por vezes complexos para a construção dos códigos, evita-se a ocorrência de erros de sintaxe. Além disso, esses recursos tornam a atividade de programação lúdica e dinâmica e, assim, os mesmos são bastante utilizados por professores para o desenvolvimento de habilidades introdutórias de programação (Kölling; Brown; Altadmri, 2017).

No entanto, para alunos com deficiência visual, ambientes de programação baseados em blocos apresentam barreiras que devem ser consideradas. Como os mesmos são intrinsecamente visuais e em geral demandam a utilização do mouse para a realização de operações com os blocos, frequentemente não são acessíveis a alunos com esse tipo de impedimento (Hadwen-Bennett; Sentance; Morrison, 2018). Para permitir o trabalho colaborativo entre estudantes com diferentes especificidades, caberia privilegiar um ambiente de programação que fosse ao mesmo tempo atraente para pessoas videntes e pessoas com impedimento visual, evitando a segregação de estudantes devido a suas características individuais (Utreras; Pontelli, 2021). Nesse sentido, a criação de um ambiente desplugado de programação baseado em blocos táteis pode ser uma alternativa. Uma outra possibilidade seria o uso de uma linguagem de programação baseada em texto que atendesse a requisitos de acessibilidade e, além disso, apresentasse uma sintaxe simples, permitindo o desenvolvimento de habilidades de programação junto a iniciantes. Neste contexto, pseudolinguagens baseadas na língua natural do estudante apresentam vantagens, na medida em que tendem a diminuir erros de sintaxe característicos de uma linguagem algorítmica.

Após caminhos, descaminhos, reformulações e ressignificações, a ideia do trabalho a ser desenvolvido estava posta e o problema de pesquisa chegava a sua forma definitiva. Desejando trabalhar a resolução de problemas e o pensamento computacional junto a estudantes com deficiência visual, buscávamos uma abordagem que privilegiasse a valorização de potencialidades através da elaboração de situações de aprendizagem da matemática criativas e desafiadoras. Mais especificamente, objetivávamos melhor

compreender as potencialidades e limitações da utilização da programação de computadores como ferramenta pedagógica para resolução de problemas matemáticos e desenvolvimento do pensamento computacional junto a estudantes com deficiência visual no Ensino Médio da Educação Básica. Só nos faltava compreender melhor, então, a que tipo de problemas estávamos nos referindo.

Considerando o público-alvo da pesquisa, a resolução de problemas que pudessem ser resolvidos através da modelagem da função afim despontou como uma proposta promissora, na medida em que, no currículo de Matemática, funções são abordadas com mais profundidade a partir do 1º ano do Ensino Médio. Dessa forma, por fim, formulamos o problema de pesquisa nos seguintes termos:

- *Quais são as potencialidades e limitações de uma linguagem algorítmica baseada em português estruturado no desenvolvimento do pensamento computacional junto a estudantes com deficiência visual no Ensino Médio da Educação Básica?*

Já em relação às hipóteses de trabalho propusemos, em um primeiro momento, que:

- *H1: O uso de uma linguagem algorítmica baseada em português estruturado apresenta uma série de vantagens quando utilizada como recurso para o desenvolvimento do pensamento computacional junto a estudantes com deficiência visual. Dentre tais vantagens, destacam-se:*
 - *H1.1: facilita a compreensão dos significados dos comandos utilizados pelo fato de estes serem escritos na linguagem natural do estudante;*
 - *H1.2: diminui a ocorrência de erros de sintaxe característicos de linguagens de programação textuais;*
 - *H1.3: permite a criação de algoritmos para a resolução de problemas específicos, desenvolvendo habilidades do pensamento computacional como abstração, decomposição, reconhecimento de padrões e construção de algoritmos.*

- *H2: Podem existir limitações associadas ao uso do leitor de tela², na medida que a leitura de alguns símbolos comumente utilizados em linguagens de programação não é realizada de maneira adequada em alguns casos.*

Por fim, o trabalho está estruturado como explicitado a seguir. Nesta primeira seção, foi apresentada a trajetória percorrida pelo pesquisador, bem como foram colocados o problema de pesquisa e as hipóteses de trabalho. Na seção 2, são apresentados o objetivo geral e os objetivos específicos da pesquisa. Na seção 3, o referencial teórico que embasa a tese é abordado. Na seção 4, é apresentada uma revisão sistemática de literatura relacionando pensamento computacional e deficiência visual, com o objetivo de identificar trabalhos correlatos e caracterizar o estado da arte no que se refere ao tema da pesquisa. Na seção 5, são mencionados os caminhos metodológicos que orientaram o desenvolvimento da pesquisa. Na seção 6, apresentamos os resultados e realizamos uma discussão acerca do que foi identificado a partir do trabalho. Finalmente, são colocadas as conclusões e considerações finais.

² Leitores de tela são aplicativos utilizados para obter respostas de um computador, *tablet* ou *smartphone* por meio de som. Desse modo, é possível que o usuário ouça um conteúdo ao invés de visualizá-lo.

2. OBJETIVOS

2.1. OBJETIVO GERAL

OG: Analisar as potencialidades e limitações da utilização de uma linguagem algorítmica em português estruturado no desenvolvimento do pensamento computacional em estudantes do Ensino Médio com deficiência visual.

2.2. OBJETIVOS ESPECÍFICOS

OE1: Elaborar uma sequência didática que possibilite utilizar a programação de computadores como recurso para a modelagem da função afim e o desenvolvimento do pensamento computacional em estudantes do Ensino Médio com deficiência visual, utilizando para tal os pressupostos da Engenharia Didática;

OE2: Desenvolver habilidades básicas de programação de computadores em estudantes com deficiência visual por meio da utilização da linguagem algorítmica ILA;

OE3: Aplicar a linguagem algorítmica ILA na modelagem da função afim e resolução de problemas matemáticos por estudantes do Ensino Médio com deficiência visual;

OE4: Identificar potencialidades e limitações da utilização do ILA como recurso para o desenvolvimento do pensamento computacional junto a estudantes do Ensino Médio com deficiência visual.

3. REFERENCIAL TEÓRICO

3.1. DEFICIÊNCIA VISUAL E EDUCAÇÃO MATEMÁTICA

A Educação Especial figura na política educacional brasileira desde o final da década de 1950, de modo que sua situação atual é fruto de um percurso marcado por transformações que ao longo de décadas determinaram os rumos para o atendimento escolar de alunos com deficiência (Mantoan, 2002). Ressalta-se, no entanto, que tal pauta ganha força no contexto nacional especialmente a partir de meados da década de 1990, após a participação do país na Conferência Mundial sobre Necessidades Educacionais Especiais, em 1994, que culminou com a assinatura da Declaração de Salamanca. Nesse momento, reafirmou-se o compromisso da nação com a Educação para Todos e reconheceu-se a necessidade e urgência de providenciar uma educação para as crianças, jovens e adultos com necessidades educacionais especiais dentro do sistema regular de ensino (Salamanca, 1994; Rezende; Pinto, 2021).

Ainda na década de 90 foi sancionada a Lei de Diretrizes e Bases da Educação Nacional (LDB 9.394/96), refletindo as transformações nas políticas públicas voltadas para a Educação Especial observadas no período. Se, por um lado, as Leis de Diretrizes e Bases da Educação Nacional de 1961 e 1971 apresentam poucos artigos sobre a “educação de excepcionais”, ou “de alunos com deficiências físicas ou mentais” (Brasil, 1961; Brasil, 1971), a LDB 9.394/96 possui um capítulo destinado à Educação Especial, reforçando o atendimento preferencial na rede regular e caracterizando as garantias que deverão ser asseguradas pelos sistemas de ensino aos educandos com deficiência, transtornos globais do desenvolvimento e altas habilidades ou superdotação (Brasil, 1996).

Ao longo das décadas seguintes, diversos outros dispositivos legais são implementados no contexto nacional, alinhando as políticas públicas locais ao movimento mundial pela educação inclusiva, destacando-se a Política Nacional de Educação Especial na Perspectiva da Educação Inclusiva (PNEE) (Brasil, 2008) e a Lei Brasileira de Inclusão da Pessoa com Deficiência (LBI) (Brasil, 2015). No entanto, embora se tenha percebido a propagação e implementação de políticas educacionais destinadas a normatizar o processo de inclusão de estudantes público alvo da Educação Especial, muitos professores colocam que não tiveram formação adequada e não se sentem preparados para enfrentar esse desafio, como mostram Fernandes e Healy

(2007) em estudo sobre a inclusão na Educação Matemática. Especificamente no que tange à educação de estudantes com deficiência visual, tal despreparo pode resultar na adoção de situações de aprendizagem baseadas na forma de aprender do vidente, negligenciando as peculiaridades e potencialidades características do seu percurso formativo.

Como apontam Sá, Silva e Simão (2010), a deficiência visual engloba tanto alterações que podem ser relativamente simples e passíveis de correção por meio de auxílios ópticos ou cirurgia, como degenerações graves, atrofias e lesões oculares que podem culminar com a cegueira. De acordo com o Decreto Federal 5296/04, que regulamenta normas gerais e critérios básicos para promoção de acessibilidade a pessoas com deficiência ou mobilidade reduzida, a deficiência visual pode ser caracterizada por comprometimentos na acuidade³ ou no campo visual⁴, de acordo com critérios específicos. Nessa perspectiva, em termos técnicos, uma pessoa é considerada com deficiência visual se se enquadra em alguma das seguintes categorias:

Deficiência visual - cegueira, na qual a acuidade visual é igual ou menor que 0,05 no melhor olho, com a melhor correção óptica; a baixa visão, que significa acuidade visual entre 0,3 e 0,05 no melhor olho, com a melhor correção óptica; os casos nos quais a somatória da medida do campo visual em ambos os olhos for igual ou menor que 60°; ou a ocorrência simultânea de quaisquer das condições anteriores (Brasil, 2004).

É importante ressaltar no entanto que a visão atual de deficiência ultrapassa o âmbito estritamente biológico da lesão, que seria uma característica corporal. Como argumentam Sá, Silva e Simão (2010), se durante o século XX as classificações da deficiência visual eram influenciadas pela hegemonia do modelo⁵ médico, que compreende a deficiência como um fenômeno estritamente biológico, atualmente as

³ De acordo com Sá, Silva e Simão (2010, p.9), acuidade visual é “uma avaliação quantitativa que revela dados percentuais sobre a capacidade de discriminação de estímulos visuais em uma escala linear e gradual”, tendo como referência parâmetros como o tamanho, a nitidez dos objetos e a distância em que são percebidos de um ponto a outro, a partir de um padrão de normalidade da visão.

⁴ De acordo com Sá, Silva e Simão (2010, p.11), o campo visual é “a área de abrangência na qual um objeto pode ser visto enquanto o olho está fixado em um determinado ponto”.

⁵ Como França (2014, p.7) entendemos modelo, aqui, como “a associação entre uma conceitualização específica, as ações desenvolvidas sobre o tema e a construção do lugar da pessoa com deficiência na sociedade”.

conceituações não se limitam ao diagnóstico clínico porque se inserem em uma rede de construtos sociais.

De acordo com o modelo médico, hegemônico no passado, a deficiência era entendida como a consequência lógica e natural de um corpo com lesão, adquirida inicialmente por meio de uma doença. Nessa perspectiva, a deficiência seria uma incapacidade física que levaria os indivíduos a uma série de desvantagens sociais. Segundo França (2014), esse modelo é pautado por uma ideologia normalizadora que consiste na crença na existência de um parâmetro normal que deve ser percebido e almejado pelos indivíduos e populações. Nesse contexto, a habilitação e a reabilitação seriam os principais marcadores de tais abordagens, na medida em que o que se esperava do corpo e da performance do indivíduo seria uma suposta normalidade, caracterizada pela execução de ações como faria a média das pessoas que não têm deficiência, segundo o esperado para a sua idade. Assim, a deficiência era uma característica a ser sanada e, para tal, deveriam ser feitas intervenções sobre o corpo deficiente para promover seu melhor funcionamento (França, 2013).

No entanto, tal modelo de compreensão da deficiência passa a ser alvo de críticas a partir da década de 1960, especialmente pelo fato de partir de uma abordagem individualista e restrita ao corpo, preconizando ações normalizadoras, rotulando pessoas com deficiência como inaptas e ignorando estruturas mais amplas que impediriam a participação social deste grupo (França, 2013). Ao longo das décadas seguintes, novos modelos se desenvolveram, com destaque para o modelo social da deficiência, que a caracteriza como uma construção social que marginaliza e oprime pessoas que possuem lesões específicas ou limitações aparentes em seus corpos, e o modelo biopsicossocial da deficiência, tentativa de síntese entre o modelo médico e o modelo social, que coloca a deficiência como um fenômeno complexo caracterizado pela interação entre as características físicas da pessoa e o contexto no qual ela vive (França, 2014). A legislação brasileira em vigor caracteriza a deficiência a partir do modelo biopsicossocial, na medida em que a LBI considera pessoa com deficiência aquela que tem um “impedimento de longo prazo de natureza física, mental, intelectual ou sensorial, o qual, em interação com uma ou mais barreiras, pode obstruir sua participação plena e efetiva na sociedade em igualdade de condições com as demais pessoas” (Brasil, 2015).

Mais especificamente em relação à deficiência visual, Ormelezi (2000) acredita que é preciso considerar o aspecto social da falta da visão e tratar a deficiência pelo enfoque

do respeito à diferença, indo além de indicadores de natureza estritamente biomédica e superando uma abordagem pautada na correção do déficit e da normalização. Como argumenta a autora:

Ser visto como coitado, ou como sábio, como o que tem que ser o melhor, ou aquele sobre quem paira sempre uma dúvida em relação a sua capacidade, tudo isso repercutirá sobre o caminho de desenvolvimento do sujeito cego que vive em um mundo construído pela visão, onde o ver é sinônimo de conhecer; onde esse sentido tem um papel central na formação da pessoa (Ormelezi, 2000, p.21).

Cabe destacar, aqui, que o enfoque do respeito à diferença passa pelo reconhecimento e pela valorização de tal diferença, e não pela sua negação, que não traz nenhum benefício para o desenvolvimento da pessoa com deficiência (Rosario, 2023). Santin e Simmons (1996), neste sentido, já apontavam que a minimização dos efeitos de uma deficiência pode distorcer o entendimento da contribuição particular da mesma para o desenvolvimento total de uma criança. Não se pode negligenciar que esse processo de desenvolvimento é qualitativamente diferente, uma vez que envolve estimulações provocadas pela necessidade de encontrar meios para lidar com algum tipo de impedimento; no caso da deficiência visual, as limitações impostas pela carência, parcial ou total, de um órgão sensorial (Fernandes; Healy, 2023).

Nessa perspectiva, é preciso considerar que, por terem um equipamento sensorial diferente, estudantes com deficiência visual organizam suas percepções de mundo de maneira particular, diferente da dos videntes, e apresentam ainda um desenvolvimento sensorial, afetivo e cognitivo da linguagem peculiar (Nunes; Lomônaco, 2008; Santin; Simmons, 1996). Desse modo, é importante que professores que trabalham com estudantes com essa característica estejam atentos às especificidades do trabalho com estes discentes. Como argumentam Santin e Simmons (1996, p. 1-2):

Não se pode realisticamente assumir o compromisso de educar a criança “total” enquanto não se tiver compreendido o desenvolvimento exclusivo de cada criança. Considerar a criança que nasceu cega como sendo uma *criança normal sem a visão* será impor a ela um conjunto artificial de dimensões, tais como o desenvolvimento sensorial sem a visão, ou o desenvolvimento afetivo sem a visão (Santin; Simmons, 1996, p. 1-2, grifo nosso).

Mais especificamente no que tange à Educação Matemática, é necessário considerar o estudante como um ser cultural e investigar como diferentes aspectos desse ser podem impactar nas suas formas particulares de se apropriar das práticas matemáticas escolares. De acordo com Healy e Powell (2013), por exemplo, ao esperar-se que todos os alunos, independentemente de suas diferenças, experimentem e se apropriem da mesma forma dos artefatos que atualmente compõem a matemática escolar, corre-se o risco de falhar em reconhecer como válidas formas de apropriação e uso de ferramentas matemáticas que desviam do esperado. Fernandes e Healy (2023), nesse sentido, destacam a importância da linguagem, do corpo e dos gestos nas interações com as diferentes representações de objetos matemáticos, reforçando a relevância dessas ferramentas semióticas no desenvolvimento cognitivo de indivíduos cuja carência de um dos órgãos dos sentidos implica uma apropriação da cultura de um modo particular.

Fernandes e Healy (2023) argumentam, ainda, em favor da construção de cenários de aprendizagem nos quais os “diferentes” possam vivenciar trajetórias que favoreçam o compartilhamento e a negociação de significados para os objetos matemáticos estudados. Assim, em lugar de inserir o estudante “diferente” em um contexto escolar organizado para um aprendiz padrão, as autoras defendem a construção de um ambiente que possibilite a exploração da Matemática em um espaço compartilhado, compreendendo que as formas pelas quais aprendemos podem variar de acordo com nossas diferentes bagagens e experiências sensoriais, linguísticas e culturais (Healy; Fernandes, 2011).

Na mesma linha, Skovsmose (2019) defende a ideia de uma Educação Inclusiva que possa promover um encontro das diferenças ao invés de encaixar pessoas com deficiência em um padrão de normalidade. Inserindo tal interpretação no contexto da Educação Matemática, o autor argumenta que é possível privilegiar ambientes de aprendizagem que possibilitem que pessoas com deficiência visual e pessoas videntes trabalhem de maneira conjunta na resolução dos mesmos problemas. Assim, embora o autor considere que a elaboração de objetos de aprendizagem apropriados seja vital para que estudantes com deficiência visual possam acessar adequadamente a Matemática escolar, tais objetos não podem ser pensados como exclusivos para estudantes com essa característica. Desse modo, os objetos de aprendizagem, em um cenário inclusivo, devem poder ser explorados por qualquer grupo de estudantes, favorecendo a colaboração entre os diferentes. O autor expande, ainda, a natureza dos

encontros possíveis, argumentando que se pode pensar e construir cenários que facilitem qualquer tipo de encontro: entre diferentes idades, habilidades, culturas e religiões, entre outros.

Marcelly (2015) também destaca a importância da utilização de outros recursos metodológicos e cenários de aprendizagem de modo que os olhos não sejam tidos como o único meio de entrada de informação, especialmente no ensino da Matemática. Em particular, a autora ressalta a relevância do uso de materiais manipuláveis enquanto ferramentas de *visualização*, produção de significados e desenvolvimento do pensamento matemático por estudantes cegos. No entanto, a autora vai além, argumentando que o uso de tais materiais é uma estratégia que traz contribuições para qualquer estudante. Desse modo, a abordagem da autora dialoga com o encontro das diferenças defendido por Skovsmose (2019), na medida em que, com os recursos disponíveis, se cria um cenário de aprendizagem acessível ao estudante com deficiência visual que, ao mesmo tempo, é adequado aos estudantes que enxergam.

Entendemos, assim como os autores mencionados, que para que estudantes com deficiência visual de fato tenham possibilidade de acessar e construir significados acerca dos conteúdos da matemática escolar, é necessário o uso de materiais didáticos adequados e acessíveis, que explorem suas potencialidades sem deixar de levar em conta sua limitação sensorial. Materiais manipuláveis, por exemplo, são importantes ferramentas na formação de conceitos por pessoas com deficiência visual. No entanto, é possível construir cenários passíveis de serem explorados por estudantes com diferentes especificidades, utilizando tanto materiais de natureza tátil quanto recursos auditivos e de outras naturezas.

Por fim, é importante refletir sobre o papel das tecnologias digitais nesse contexto. O cenário pandêmico vivenciado entre 2020 e 2023, por exemplo, implicou a necessidade de isolamento social e adoção de práticas pedagógicas de caráter não presencial, mediadas por tecnologias digitais (Rezende; Pinto, 2021). Como concluem Klingenberg e Holkesvik (2020) em revisão de literatura sobre aprendizagem virtual de matemática entre alunos com deficiência visual severa, por mais que tecnologias assistivas baseadas em áudio e toque sejam ferramentas potentes para melhorar as habilidades matemáticas de alunos com esse tipo de impedimento, há necessidade de pesquisas adicionais e mais reflexão sobre o uso de tecnologias digitais na

aprendizagem da matemática junto a esses alunos, especialmente os mais novos, o que reforça a relevância do tema.

3.2. PENSAMENTO COMPUTACIONAL

O uso de tecnologias digitais tem se tornado cada vez mais frequente em diferentes espaços sociais, o que vem provocando transformações importantes na forma como interagimos e praticamos ações cotidianas. Tal fato implica uma série de mudanças na organização econômica e cultural da sociedade, o que, em consequência, também impacta as dinâmicas que se dão no espaço escolar. Atualmente, as relações constituídas pelos estudantes dentro e fora da escola são amplamente mediadas por tecnologias digitais e, desse modo, é importante que as instituições educacionais abordem o uso de tais recursos no espaço escolar a partir de diferentes perspectivas.

Valente (2016) alerta, no entanto, para o tipo de uso que se faz dessas tecnologias no contexto educacional. O autor entende que, para além de uma percepção das tecnologias de informação e comunicação como simples ferramentas de escritório, é preciso instrumentalizar estudantes para o uso crítico desses recursos como ferramentas de criação. Países como Estados Unidos, Canadá e Israel, assim como diversos países da Europa, já vêm implementando na Educação Básica o ensino de habilidades relacionadas a tecnologias digitais, ao desenvolvimento do pensamento computacional e à introdução à programação de computadores (Ferri; Rosa, 2016; Valente, 2016):

[...] na Europa, a Comissão Europeia publicou o relatório *European Schoolnet* (2014) baseado no estudo da situação atual em 20 países europeus, sendo que em 13 desses países a programação já faz parte de disciplinas obrigatórias do ensino infantil ao nono ano (K- 9, equivalente ao nosso Ensino Fundamental), como na Estônia e Grécia. A Inglaterra alterou a disciplina obrigatória de Informática (denominada ICT), que explorava as ferramentas de escritório, substituindo-a pela Computing, estruturada no tripé: Ciência da Computação, Tecnologia da Informação e Letramento Digital [...] (Valente, 2016, p.867).

A expressão “pensamento computacional” foi utilizada pela primeira vez pelo matemático Seymour Papert em 1980, inicialmente a partir de uma abordagem construcionista, na qual o envolvimento social e afetivo dos estudantes na aprendizagem do conteúdo técnico seria importante para tornar a programação uma ferramenta interdisciplinar, utilizada também na aprendizagem de outras disciplinas. Papert defendia

o uso da programação no ensino de matemática para crianças, acreditando que aprender a se comunicar com um computador poderia interferir na maneira como outras aprendizagens ocorriam. Brackman (2017) destaca que embora o termo só tenha sido empregado em 1980, as ideias do pensamento computacional já podiam ser observadas em um trabalho anterior de Papert e Solomon (1972), que aborda a cultura dos computadores e o papel da tecnologia no ensino de crianças.

Uma concepção mais moderna da expressão advém do trabalho seminal de Jeanette Wing (2006), que argumenta que aprender a pensar como um cientista da computação seria um benefício para todas as pessoas, independente da profissão, e que o pensamento computacional representa uma lente para compreender a estrutura algorítmica da atualidade (Lodi; Martini, 2021; Nascimento; Santos; Tanzi, 2018). No trabalho, embora a autora não forneça uma definição única e formal para o termo, são apresentados diversos aspectos que o envolvem. De maneira geral, o pensamento computacional não seria, como o termo poderia sugerir, uma forma de fazer com que seres humanos pensem como computadores, mas sim estaria relacionado com a maneira como humanos resolvem problemas. Entre os aspectos do pensamento computacional que a autora menciona no texto, se destacam:

- é uma habilidade fundamental para todas as pessoas, e não apenas cientistas;
- envolve a reformulação de um problema aparentemente difícil em outro que sabemos resolver;
- envolve resolução de problemas, *design* de sistemas e a compreensão do comportamento humano a partir de conceitos fundamentais da ciência da computação;
- envolve o uso da abstração e da decomposição ao enfrentar um problema ou construir um sistema complexo;
- envolve escolher uma representação apropriada para um problema, modelando de maneira adequada aspectos relevantes do mesmo.

Em 2008, Wing argumenta novamente que o pensamento computacional irá influenciar todos os campos do conhecimento, o que traria um novo desafio educacional para a sociedade. Por mais que ainda não seja apresentada, neste trabalho, uma definição formal do termo, a autora coloca que o mesmo representa uma forma de pensamento analítico que se relaciona com três tipos diferentes de pensamento: o

matemático, pelas diversas maneiras como podemos resolver um problema; o da Engenharia, pelas diferentes formas como podemos abordar o projeto e a avaliação de sistemas complexos que operam de acordo com as restrições do mundo real; e o científico, pelas diferentes maneiras como podemos abordar a compreensão da computabilidade, da inteligência, da mente e do comportamento humano (Wing, 2008).

Ainda na década de 2000, Blikstein (2008) argumenta que o pensamento computacional já estaria transformando profundamente a academia e a indústria. Além disso, para o autor, a lista de habilidades e conhecimentos necessários para o exercício pleno da cidadania no novo século seria extensa, indo muito além da capacidade de leitura, escrita e realização de operações aritméticas. Entre tais habilidades essenciais, o autor menciona mais especificamente o pensamento computacional, caracterizando-o como a capacidade de usar o computador como um instrumento de aumento do poder cognitivo e operacional humano. Nesse contexto, em lugar de simplesmente navegar na internet, enviar emails, publicar em blogs ou operar processadores de texto, seria importante usar computadores e redes de computadores para aumentar nossa produtividade, inventividade, e criatividade.

Em 2010, Wing apresenta sua primeira definição formal para o pensamento computacional. Nesse momento, a autora define o PC como o processo de pensamento envolvido na formulação de um problema e na expressão de sua solução de modo que a solução seja representada de maneira que possa efetivamente ser executada por um agente processador de informações. A autora esclarece ainda que, na definição proposta, o termo “expressar” significa criar uma representação linguística com o propósito de comunicar a solução para outros, pessoas ou máquinas. Informalmente, o pensamento computacional descreveria a atividade mental na formulação de um problema de modo que se admita uma solução computacional (Wing, 2010).

Ao longo dos anos seguintes, novas abordagens e definições foram desenvolvidas para o termo pensamento computacional. Em 2011, a *International Society for Technology in Education* (ISTE) e a *Computer Science Teachers Association* (CSTA) publicam um trabalho que, além de fornecer uma definição operacional para o pensamento computacional no contexto da Educação Básica, apresenta um conjunto de habilidades que seriam características do PC, e uma série de sugestões de atividades passíveis de serem desenvolvidas com estudantes desse contexto (ISTE/CSTA, 2011). De acordo com a definição apresentada pelas associações, o pensamento

computacional seria um processo de resolução de problemas que inclui, mas não se limita às seguintes características:

- Formulação de problemas de um modo que nos permita usar um computador e outras ferramentas para ajudar a resolvê-los;
- Organização e análise lógica de dados;
- Representação de dados por meio de abstrações como modelos e simulações;
- Automatização de soluções através de pensamento algorítmico;
- Identificação, análise e implementação de possíveis soluções com o objetivo de alcançar a combinação mais eficiente e eficaz de etapas e recursos;
- Generalização e transferência deste processo de resolução de problemas para uma ampla variedade de problemas.

Além disso, as características mencionadas anteriormente seriam reforçadas por algumas atitudes essenciais no âmbito do pensamento computacional. Tais atitudes seriam a confiança para lidar com a complexidade, a persistência em trabalhar com problemas difíceis, a tolerância à ambiguidade, a capacidade de lidar com problemas abertos e a capacidade de se comunicar e trabalhar com outras pessoas para alcançar um objetivo comum (ISTE/CSTA, 2011).

Em 2012, uma nova definição é apresentada pela *The Royal Society*, academia científica independente do Reino Unido, segundo a qual o pensamento computacional seria o processo de reconhecer aspectos da computação no mundo que nos rodeia, aplicando ferramentas e técnicas da Ciência da Computação para entender e raciocinar sobre sistemas e processos naturais e artificiais (Royal Society, 2012). No mesmo ano, Aho (2012) define o PC como o processo de pensamento envolvido na formulação de problemas de modo que as soluções possam ser representadas através de passos computacionais e algoritmos.

Brennan e Resnick (2012) investigam como atividades de aprendizagem baseadas no processo de programação de mídias interativas promovem o desenvolvimento do pensamento computacional em crianças. Embora não seja proposta uma definição para o PC no estudo, os autores descrevem uma série de características do pensamento computacional a partir de três dimensões. A primeira dimensão se refere aos *conceitos* que os estudantes mobilizam e aprendem enquanto programam utilizando uma linguagem de programação. Nesse sentido, os autores identificam sete conceitos

característicos do pensamento computacional, cuja mobilização é comum em diferentes linguagens de programação (André, 2018; Rodriguez; Reis; Isotani, 2017). Os conceitos identificados são:

- sequências (identificação de uma série de etapas de uma tarefa);
- loops (execução de uma mesma sequência várias vezes);
- paralelismo (ocorrência simultânea de ações);
- eventos (quando um acontecimento causa outro acontecimento);
- condicionais (tomada decisões com base em condições estabelecidas);
- operadores (expressão de operações matemáticas e lógicas);
- dados (armazenagem, recuperação e atualização de valores).

Brennan e Resnick (2012) colocam, no entanto, que a caracterização do pensamento computacional simplesmente em torno de conceitos não é suficiente, na medida em que seriam negligenciados outros elementos dos processos de aprendizagem e de criação de quem aprende. Desse modo, seria importante ir além do que se aprende e refletir também sobre como se aprende, culminando em uma segunda dimensão de caracterização do pensamento computacional. Assim, através da observação das estratégias adotadas pelas crianças, os autores identificam quatro conjuntos de *práticas* características do PC, desenvolvidas pelos estudantes durante as atividades de desenvolvimento e implementação de projetos, a saber:

- ação iterativa e incremental (realizar ciclos iterativos de imaginação e construção, desenvolvendo, verificando e implementando melhorias a partir das experiências vivenciadas e do surgimento de novas ideias);
- teste e depuração (desenvolver estratégias para antecipar, encarar e corrigir problemas, em um processo de tentativa e erro para garantir o funcionamento adequado do programa);
- reutilização e reformulação (fazer algo a partir do que já foi feito por outros, possibilitando a realização de criações mais complexas que aquelas que seriam feitas por conta própria, e fomentando a discussão sobre propriedade e autoria dos códigos);
- abstração e modulação (construir algo grande a partir da união de peças menores).

A terceira dimensão mencionada por Brennan e Resnick (2012) se refere a perspectivas mais amplas que seriam consequência do desenvolvimento do pensamento

computacional, possibilitando a promoção de alterações na compreensão dos estudantes sobre o mundo que o cerca e sobre eles mesmos (André, 2018; Rodriguez; Reis; Isotani, 2017). De acordo com os autores, as perspectivas do pensamento computacional seriam:

- capacidade de se expressar (compreender a computação como um meio de criação e expressão de ideias);
- capacidade de conectar (reconhecer os benefícios das interações, sejam elas presenciais ou via redes *online*, e da criação com e para os outros);
- capacidade de questionar (compreender a importância de se fazer perguntas sobre - e através da – tecnologia e, de maneira mais ampla, sobre o mundo).

Em 2014, em trabalho no qual descreve como o pensamento computacional tem se difundido na pesquisa e na educação, beneficiando pesquisadores e profissionais de um número crescente de áreas para além da ciência da computação, Wing apresenta uma nova definição para o termo. Embora similar à definição anterior, desta vez a autora é mais concisa, colocando o pensamento computacional como o processo de pensamento envolvido na formulação de um problema e na expressão de sua solução de modo que um computador – humano ou máquina – possa efetivamente realizar (Wing, 2014).

Em 2015, a Computing at School (CAS), uma associação do Reino Unido que envolve professores, universidades, indústrias e sociedades profissionais, entre outros membros, atuando para promover e apoiar o ensino da computação e disciplinas relacionadas em escolas da nação (Crick; Sentance, 2011), publicou um guia com o objetivo de desenvolver uma compreensão compartilhada do ensino do pensamento computacional nas escolas (Csizmadia et al., 2015). No guia, os autores colocam que o pensamento computacional é o processo de reconhecer aspectos da computação no mundo que nos rodeia, aplicando ferramentas e técnicas da computação para compreender e raciocinar sobre recursos naturais, sociais e sistemas e processos artificiais.

Csizmadia et al. (2015) destacam, ainda, uma série de conceitos que estariam diretamente relacionados com o pensamento computacional, a saber: a abstração, ato de tornar um artefato mais compreensível através da redução de detalhes desnecessários; a decomposição, um modo de pensar sobre artefatos a partir de suas partes componentes; o pensamento algorítmico, modo de chegar à solução de um problema através de uma definição clara de passos; a generalização, que se relaciona

com a identificação e exploração de padrões, similaridades e conexões; e a avaliação, ato de assegurar que uma solução, seja ela um algoritmo, um sistema ou um processo, é de fato boa.

Brackmann et al. (2016, p. 197) colocam que o pensamento computacional é “uma abordagem de ensino que usa diversas técnicas oriundas da Computação para a resolução desses problemas em conjunto com as novas competências do século 21”. No trabalho os autores defendem, ainda, que o PC tem por base quatro pilares interdependentes e de grande importância durante o processo de formulação de soluções computacionalmente viáveis para problemas. Tais pilares seriam a decomposição (divisão de um problema em etapas menores), o reconhecimento de padrões (identificação de problemas similares previamente resolvidos), a abstração (identificação dos elementos relevantes para a resolução do problema), e os algoritmos (elaboração de passos ou regras simples para resolver cada um dos problemas criados).

Brackmann (2017) novamente destaca a importância da abstração, da decomposição, do reconhecimento de padrões e da algoritmização enquanto pilares do pensamento computacional, apresentando um conjunto de referências reforçando tal formulação (Code.Org, 2015; BBC Learning, 2015; Liukas, 2015; Csizmadia et al., 2015; Grover; Pea, 2013). Dessa vez, no entanto, o autor apresenta uma definição mais refinada para o PC, argumentando que o termo não pode ser entendido como uma simples aptidão para o manuseio de aplicativos ou dispositivos eletrônicos, o que, na realidade, se relacionaria com o conceito de alfabetismo digital. Tampouco, segundo o autor, deve-se entender o termo como uma forma mecânica de se pensar, limitando a criatividade humana. Para ele, o pensamento computacional pode ser definido como:

(...) uma distinta capacidade criativa, crítica e estratégica humana de saber utilizar os fundamentos da Computação, nas mais diversas áreas do conhecimento, com a finalidade de identificar e resolver problemas, de maneira individual ou colaborativa, através de passos claros, de tal forma que uma pessoa ou uma máquina possam executá-los eficazmente (Brackmann, 2017, p.29).

Por fim, Guarda e Pinto (2020) defendem que o pensamento computacional seria uma abordagem para a resolução de problemas que explora processos cognitivos e envolve a capacidade de compreender as situações propostas e de criar soluções através de modelos matemáticos, científicos ou sociais. Além disso, tal abordagem seria

capaz de promover um aumento em nossa criatividade, inventividade e produtividade. No estudo, é proposto um novo modelo que abrange e sintetiza dimensões e habilidades do pensamento computacional abordadas por outros autores em trabalhos anteriores. Entendendo o PC como um modo de resolver problemas algorítmicamente, os autores definem os três estágios sequenciais característicos do processo de resolução:

- 1º estágio: definição do problema
 - Dimensões do PC: formulação do problema, abstração, reformulação do problema, decomposição;
- 2º estágio: solução do problema
 - Dimensões do PC: coleta e análise de dados, pensamento algorítmico, paralelização e iteração, automação;
- 3º estágio: análise da solução
 - Dimensões do PC: generalização, teste, avaliação.

Considerando que, no contexto desta pesquisa, abordamos o pensamento computacional em uma perspectiva educacional e, além disso, voltada para a Educação Básica, entendemos que as perspectivas de Brennan e Resnick (2012) e Csizmadia et al. (2015) são mais adequadas enquanto formulação teórica orientadora do trabalho. Ademais, para além das quatro habilidades tidas como pilares do pensamento computacional (Brackmann, 2017; Brackmann et al., 2016), entendemos que a avaliação também é uma habilidade fundamental em se tratando de resolução de problemas. Desse modo, assim como Csizmadia et al. (2015), nos centraremos em cinco habilidades tidas, para nós, como fundamentais do pensamento computacional: abstração, decomposição, generalização (reconhecimento de padrões), algoritmização e avaliação.

3.3. LINGUAGENS ALGORÍMICAS EM PORTUGUÊS ESTRUTURADO

Um programa de computador é um conjunto de instruções que são compreendidas e executadas de dentro da memória da máquina, pela sua unidade central de processamento. De maneira geral, um programa é escrito por seres humanos usando uma linguagem de programação que, por sua vez, consiste em um conjunto de símbolos e regras internas que regem a escrita e estruturação correta de um programa (Bassil, 2019). Como coloca Manzano (2017, p.15), se a linguagem é um meio sistemático de comunicar ideias através de símbolos, “no caso dos computadores os símbolos de uma

linguagem de programação são as palavras reservadas da linguagem que dão à máquina uma ordem de execução”. Em suma, respeitando-se o vocabulário e as regras específicas da linguagem, é possível criar um código formado por um conjunto de signos que serão compreendidos pelo computador na forma de uma ordem de execução.

No entanto, muitas vezes estudantes enfrentam uma série de dificuldades quando têm seus primeiros contatos com linguagens de programação de computadores. De acordo com Noschang et al. (2014), não são raros os estudos que abordam tais dificuldades, sob diferentes aspectos. Para os autores, o ensino de programação introdutória deveria focar no desenvolvimento da lógica de programação e da habilidade de resolução de problemas, sendo a aprendizagem de uma determinada linguagem um aspecto secundário. Assim, o uso de linguagens e ambientes de programação profissionais, que privilegiam a produtividade em lugar da aprendizagem, pode trazer dificuldades adicionais aos estudantes que se iniciam na tarefa de produzir algoritmos e traduzi-los em programas.

Para Souza, Batista e Barbosa (2016), o processo de ensino e aprendizagem de programação é complexo, sendo comum que os estudantes apresentem dificuldades de diferentes naturezas, desde problemas com a compreensão de novos conceitos, como ponteiros, recursão e declaração de variáveis, até a própria aplicação dos conceitos desenvolvidos na construção de programas. Além disso, a maioria das linguagens de programação existentes utilizam comandos em língua inglesa para expressar sua sintaxe e vocabulário, o que acaba representando um dificultador a mais para estudantes que não têm o inglês como sua primeira língua (Bassil, 2019). Tal fato também é mencionado por Noschang et al. (2004), que destacam ainda a baixa fluência em língua inglesa apresentada por estudantes que concluem o Ensino Médio no Brasil.

Em mapeamento sistemático sobre problemas no ensino e aprendizagem de programação, Souza, Batista e Barbosa (2016), além de identificarem dificuldades comumente observadas no processo, também procuraram identificar soluções que vêm sendo implementadas para amenizar a questão. Entre as propostas identificadas é destacada a utilização da linguagem natural como uma alternativa atraente e promissora pelo fato de o estudante não precisar aprender uma gramática formal para compreender

fundamentos de programação. Dessa forma, o uso de uma linguagem algorítmica⁶ na qual os comandos são escritos na mesma língua do estudante diminuiria o esforço empregado na tarefa de construir um programa (Cambranes, 2013; Oliveira; Monteiro; Roman, 2013; Souza; Batista; Barbosa, 2016). Na mesma linha, Evaristo e Crespo (2000) colocam que existem divergências em relação ao melhor modo de iniciar a aprendizagem de programação, utilizando-se uma pseudolinguagem (linguagem algorítmica) ou uma linguagem de programação usual. Os autores colocam, no entanto, que a primeira hipótese traria o benefício de possibilitar uma aprendizagem menos formal, sem a necessidade de recorrer a aspectos técnicos de um sistema de computação, permitindo que todo esforço do estudante seja dedicado ao desenvolvimento da lógica de programação.

De acordo com Manzano (2017), os princípios da escrita de códigos utilizando elementos da língua natural se baseiam em uma técnica norte-americana denominada PDL (Program Design Language), desenvolvida por Stephen H. Caine e E. Kent Gordon na década de 1970 para a documentação de códigos de programas. Com o tempo, o método passou a ser utilizado academicamente em salas de aula com o objetivo de simplificar a escrita de códigos, passando também a ser adaptado e utilizado em diversos países, de acordo com o idioma local. Em particular, uma linguagem algorítmica que reúne elementos da língua portuguesa e possui estrutura semelhante a uma linguagem típica para programação de computadores é comumente referida como português estruturado, ou *portugol* (Sousa et al., 2022). Este último termo, cunhado no Brasil a partir do trabalho de Guimarães e Lages (1985), passou a ser utilizado com frequência na área acadêmica para se referir ao português estruturado.

Existem diferentes estilos de escrita em português estruturado, não havendo uma norma técnica que forneça diretrizes e padronizações para o seu uso. Ainda assim, como menciona Souza (2009), o uso desse tipo de estrutura permite que o estudante utilize uma sintaxe de programação mais simples, com comandos mais próximos da língua portuguesa, ao passo que também utiliza conceitos de programação como declaração de variáveis, linhas de código, comandos de entrada e de saída, entre outros. Dessa

⁶ No contexto deste trabalho, linguagem algorítmica é entendida como uma linguagem cujos símbolos reúnem um subconjunto restrito da linguagem natural, além de operadores lógicos, aritméticos e relacionais, que é utilizada para descrever algoritmos que possam ser executados por computadores.

forma, na medida em que remete à liberdade da linguagem natural sem deixar de lado o formalismo de linguagens de programação, o português estruturado apresenta potencialidades importantes como recurso pedagógico para o desenvolvimento de conceitos e aplicações da lógica de programação.

Na mesma perspectiva, Manso, Oliveira e Marques (2009) argumentam que a definição de uma linguagem algorítmica estruturada e em português possibilita que o estudante codifique seus programas com instruções expressas no léxico da sua língua materna, simplificando a sua aprendizagem de técnicas de programação. Esta linguagem algorítmica, que serve de suporte à definição de dados e aos algoritmos que os manipulam, permite representar de forma clara e inequívoca as instruções a serem executadas pelo computador. Além disso, o uso deste recurso possibilita o desenvolvimento do raciocínio algorítmico numa linguagem formal simples, focando o interesse do aluno no desenvolvimento do algoritmo e proporcionando uma melhor transição para os ambientes de programação tradicionais.

Diferentes linguagens algorítmicas baseadas em português estruturado são mencionadas na literatura, sendo alguns exemplos Portugol Studio (Noschang et al., 2014; Esteves et al., 2019), Portugol IDE (Manso; Oliveira; Marques, 2009), G-Portugol (Silva, 2006), Portugol Viana (Jesus, 2011), CIFluxProg (Santiago; Dazzi, 2004), CIFluxProgII (Miranda et al., 2005), PortuCol (Barbosa; Couto; Terra, 2016), WEB-UNERJOL (Ferrandin; Stephani, 2012), WebPortugol (Hostins; Raabe, 2007), VisuAlg (Souza, 2009), ILA (Manzano; 2017; Evaristo; Pinto, 2000), MACP, CompAlg (Bilabila, 2017), UFMA-CP (Soares Neto et al., 2008) e AMBAP (Almeida et al., 2002). No entanto, após verificação de links ou orientações existentes nos trabalhos mencionados, bem como realização de buscas em sítios web, apenas foram encontradas versões atualmente disponíveis para *download* e utilização das linguagens Portugol Studio, G-Portugol, VisuAlg, Portugol IDE, Portugol Viana e ILA, cujas especificidades são detalhadas a seguir.

Especificidades das diferentes linguagens algorítmicas

O Portugol Studio é um ambiente de desenvolvimento que foi criado com o objetivo de facilitar a aprendizagem de programação para usuários que dominam a língua portuguesa e reduzir as dificuldades enfrentadas por iniciantes em programação ao se depararem pela primeira vez com as linguagens de programação e ambientes de

desenvolvimento comerciais. Seu desenvolvimento foi iniciado em 2007 pelo estudante Luis Fernando Noschang, graduando de Ciência da Computação da Universidade do Vale do Itajaí (UNIVALI) à época (Esteves et al., 2019; Bilabila, 2017). A primeira versão do recurso começou a ser utilizada em 2011 dentro da própria universidade.

Em 2013, o Portugol Studio foi lançado dentro de grupos acadêmicos, atingindo 2000 *downloads* em oito meses. Desde seu lançamento, diversas mudanças foram implementadas, tornando seu *layout* mais moderno, melhorando seu desempenho, e adicionando novas bibliotecas, exemplos de programas, inspetores para depuração e a possibilidade de adicionar *plugins* à ferramenta, de modo que a última versão do recurso (v.2.7.5) foi lançada em 2020. Além disso, com o objetivo de tornar a ferramenta acessível em qualquer lugar, trazendo todos os recursos do ambiente *desktop* para a Internet, o ambiente *online* Portugol WebStudio foi desenvolvido como uma adaptação do Portugol Studio (Esteves et al., 2019).

O G-Portugol é projeto de *software* livre que fornece um compilador capaz de traduzir para a linguagem C, *assembly* ou código executável, e interpretar códigos escritos através do código-fonte. Pode ser executado nos sistemas operacionais Windows e Linux, mas como é distribuído na forma de código-fonte, primeiramente deve ser compilado para que então possa ser utilizado. A proposta da linguagem é disponibilizar uma implementação do português estruturado com enfoque didático, fornecendo recursos de edição, compilação, execução e depuração de programas escritos nessa linguagem, e favorecendo estudantes que iniciam a aprendizagem de desenvolvimento de *softwares*, bem como professores que ensinam disciplinas relacionadas a computação (Medeiros, 2015; Silva, 2006).

Já o VisuAlg é uma ferramenta que se assemelha a um ambiente de desenvolvimento formal, como Delphi ou Visual Basic, mas que é voltado para o meio acadêmico. Utilizando uma variação de português estruturado inspirada na linguagem Pascal, o recurso fornece aos estudantes de disciplinas de programação ferramentas para digitar, executar e depurar códigos para resolver problemas. Além disso, a ferramenta também fornece aos professores recursos didáticos voltados para o detalhamento da execução passo a passo e a visualização do conteúdo das variáveis. A execução do programa pode ser feita de três modos distintos: o modo padrão, em que as instruções do código são interpretadas e executadas imediatamente; o modo passo a passo, em que o usuário comanda a execução do programa linha por linha, para efeitos

de depuração; e o modo de animação, semelhante ao modo padrão, mas com uma pausa que pode variar de 0.2 segundos a 5 segundos antes da execução de uma linha de código, permitindo ao estudante acompanhar o fluxo de processamento (Souza, 2009; Medeiros, 2015).

O Portugal IDE, por sua vez, é um *software* livre que permite a construção, a execução e a depuração de algoritmos nas linguagens algorítmica e fluxográfica. Foi desenvolvido inicialmente como um ambiente de aprendizagem para o ensino da programação que, além de utilizar o português para o desenvolvimento de algoritmos, disponibiliza um conjunto de ferramentas pedagógicas para a aprendizagem de programação de computadores. O recurso, constituído pelas duas linguagens de definição de algoritmos e um ambiente de exploração das mesmas, pode ser executado nos sistemas operacionais Windows, Linux e Mac OS X (Manso; Oliveira; Marques, 2009; Medeiros, 2015).

O Portugal Viana é um projeto de *software* livre desenvolvido a partir do Portugal IDE que fornece uma ferramenta que possibilita a edição, verificação, execução e depuração de algoritmos codificados nessa linguagem. O recurso foi desenvolvido em 2008 pela Escola Superior de Tecnologia e Gestão de Viana do Castelo, em Portugal, com o objetivo de criar um simulador de linguagem de programação para apoio às aulas de Introdução à Programação dos cursos de Engenharia da instituição (Medeiros, 2015; Bilabila, 2017).

Por fim, o ILA não é propriamente um ambiente, mas sim um interpretador que permite a testagem de algoritmos em um português estruturado. O Projeto ILA (Interpretador de Linguagem Algorítmica), iniciado em 1990 na Universidade do Vale do Rio dos Sinos (UNISINOS), nasceu com o objetivo de utilizar o português estruturado com vistas a proporcionar um ambiente de programação com sintaxe simples, minimizando problemas na construção de algoritmos e atendendo às expectativas de professores e estudantes. Inicialmente desenvolvido para o sistema operacional Windows de 32 bits, o projeto passou a ser disponibilizado para sistema operacional Windows de 64 bits, transformando-se em ILA+, e sua condução passou a ser centralizada na Universidade Federal Fluminense, Campus Rio das Ostras (Manzano, 2017).

Para utilizar o ILA, é necessário instalar o interpretador da linguagem em um diretório de trabalho no computador. Os arquivos para instalação se encontram

disponíveis para *download* de forma gratuita no sítio web <https://www.professores.uff.br/screspo/2017/09/06/ila-interpretador-de-linguagem-algoritmica/>. Após a instalação, é preciso que o interpretador seja executado a partir da linha de comando do local onde está armazenado, ou que tenha seu local associado ao *path* do sistema operacional para possa ser executado a partir de qualquer diretório. Como o ILA não apresenta um editor de textos, os programas são construídos através de um editor qualquer e salvos no formato txt. A identificação do arquivo deve seguir o padrão XXX.ila, onde XXX representa o nome dado para o arquivo, que não deve ultrapassar oito caracteres. Além disso, o arquivo deve ser salvo na pasta que contém o executável ILA. Já a execução pode ser realizada através do prompt de comando do DOS (*Disk Operating System*), utilizando-se o termo “ila” antes do nome do arquivo; ou seja, ila XXX.ila.

Comparativo entre as diferentes linguagens algorítmicas

Como mencionado anteriormente, o português estruturado é caracterizado por uma falta de padronização semântica, não havendo uma norma que forneça diretrizes para o seu uso. Desse modo, diferentes linguagens algorítmicas podem utilizar diferentes palavras reservadas da língua para representar as mesmas instruções. Um exemplo é o caso da instrução “escrever”, que pode ser encontrada também sob a forma de “escreva” ou “imprima” (Manso; Oliveira; Marques, 2009).

Como o objetivo deste trabalho não se centra na aprendizagem de uma linguagem de programação formal, mas sim na análise das potencialidades e limitações de uma linguagem algorítmica como recurso para o desenvolvimento do pensamento computacional em estudantes que, em princípio, não tiveram contatos anteriores com a programação de computadores, a escolha de uma linguagem mais simples parece ideal. Além disso, considerando que os participantes da pesquisa são estudantes com deficiência visual, é essencial que a construção dos códigos na linguagem escolhida possa ser realizada com o auxílio de um leitor de telas.

Para realizar um comparativo entre as diferentes linguagens, utilizou-se cada uma delas para construir o código de um programa que solicita dois números ao usuário e retorna a soma entre os mesmos. No Quadro 1 são exibidos os códigos construídos através de cada linguagem.

Quadro 1: Comparativo entre códigos de um mesmo programa construídos em Portugol Studio, G-Portugol, VisuAlg, Portugol IDE, Portugol Viana e ILA.

Linguagem algorítmica	Código construído
Portugol Studio	<pre> programa { funcao inicio() { real a, b, c escreva("Digite o primeiro numero: ") leia(a) escreva("Digite o segundo numero: ") leia(b) c = a + b escreva("O resultado da soma e: ", c) } } </pre>
G- Portugol	<pre> algoritmo soma; variáveis a, b, c: real; fim-variáveis início imprima("Digite o primeiro numero: "); a:= leia(); imprima("Digite o segundo numero: "); b:= leia(); c:= a + b; imprima("O resultado da soma e: ", c); fim </pre>
VisuAlg	<pre> algoritmo "soma" var a, b, c: real início escreva("Digite o primeiro numero: ") leia(a) escreva("Digite o segundo numero: ") leia(b) c:= a + b escreva("O resultado da soma e: ", c) finalgoritmo </pre>
Portugol IDE	<pre> início variavel real a, b, c escrever "Digite o primeiro numero: " ler a escrever "Digite o segundo numero: " ler b c <- a + b escrever "O resultado da soma e: ", c fim </pre>
Portugol Viana	<pre> início variavel real a, b, c escrever "Digite o primeiro numero: " ler a escrever "Digite o segundo numero: " ler b c <- a + b escrever "O resultado da soma é: ", c fim </pre>
ILA	<pre> variaveis numerico a, b, c início escrever "Digite o primeiro numero: " ler a escrever "Digite o segundo numero: " ler b soma = a + b escrever "O resultado da soma é: ", c fim </pre>

Fonte: elaborado pelo autor (2024).

A análise do Quadro 1 indica que as diferentes linguagens algorítmicas possuem comandos semelhantes, com estrutura que varia em alguns pontos. No que se refere ao programa construído, é possível observar diferenças, por exemplo, em relação ao processo de declaração de variáveis, tipos de variáveis, comandos de entrada e saída, e atribuição de valores às variáveis, entre outros. No Quadro 2 são ilustradas algumas das diferenças identificadas no que se refere ao programa construído.

Quadro 2: Diferenças na estrutura das linguagens Portugol Studio, G-Portugol, VisuAlg, Portugol IDE, Portugol Viana e ILA.

	Portugol Studio	G-Portugol	VisuAlg	Portugol IDE	Portugol Viana	ILA
Comando para declaração de variáveis	Não há	variáveis	var	variavel	variavel	variaveis
Necessário indicar o fim da declaração de variáveis	Não	Sim	Não	Não	Não	Não
Necessário utilizar ponto-e-vírgula para separar instruções	Não	Sim	Não	Não	Não	Não
Comando utilizado para iniciar o bloco de instruções	inicio	início	inicio	inicio	inicio	inicio
Comando de saída	escreva ()	imprima ()	escreva ()	escrever	escrever	escrever
Comando de entrada	leia ()	leia ()	leia ()	ler	ler	ler
Comando para atribuição de valor a variável	=	:=	:=	<-	<-	=
Comando utilizado para finalizar o bloco de instruções	Não há	fim	fimalgoritmo	fim	fim	fim

Fonte: elaborado pelo autor (2024).

O Quadro 2 revela algumas diferenças na sintaxe das linguagens apresentadas. Considerando que no contexto deste trabalho buscamos um recurso que seja acessível a estudantes com deficiência visual, é essencial que o processo de construção de códigos possa ser realizado através de um leitor de telas. Desse modo, cabem algumas considerações acerca de determinadas especificidades das linguagens mencionadas.

Em primeiro lugar, o emprego frequente de sinais gráficos (acentos, sinais de pontuação, parênteses, entre outros) na construção de códigos pode ser um dificultador para estudantes com deficiência visual, na medida em que a identificação desses símbolos pelo leitor de telas por vezes é irregular. Nesse aspecto, um problema comum é o fato de que ao realizar a leitura por linha através do leitor de telas, muitas vezes alguns destes sinais são omitidos. A linha “*escrever ‘O resultado da soma é: ’, c*”, por exemplo, é comumente lida pelos leitores de tela como “escrever o resultado da soma é c”. Para identificar vírgulas, aspas, parênteses e outros símbolos através do leitor, é necessário realizar a leitura caracter por caracter, o que pode não ser uma tarefa simples. Desse modo, linguagens que utilizem menos sinais gráficos em sua sintaxe parecem apresentar vantagens para estudantes com deficiência visual.

Em segundo lugar, o ambiente de programação da linguagem utilizada deve ser acessível a pessoas com deficiência visual. Os ambientes das linguagens VisuAlg, Portugol Viana e Portugol IDE apresentam problemas de acessibilidade na medida em que o programa construído não é lido adequadamente pelo leitor de tela. No caso dessas linguagens, quando o programa é rodado, as saídas são exibidas em tela mas não podem ser lidas pelo leitor de tela, a não ser que o mouse seja utilizado. A linguagem Portugol Studio também possui ambiente de programação próprio que, por sua vez, não apresenta o problema mencionado anteriormente. No entanto, a exploração das funcionalidades do ambiente exclusivamente pelo teclado pode ser difícil para estudantes que não têm experiência com esse tipo de prática.

Por fim, no caso das linguagens G-Portugol e ILA, os códigos são construídos através do bloco de notas e executados pelo prompt de comando do computador, o que pode ser feito através do leitor de tela. No entanto, dentre as linguagens apresentadas, o G-Portugol é a que mais utiliza sinais gráficos, o que, como mencionado, por ser um dificultador para estudantes com deficiência visual. Desse modo optamos, finalmente, pela utilização do ILA como linguagem algorítmica para condução da pesquisa.

4. PENSAMENTO COMPUTACIONAL E DEFICIÊNCIA VISUAL NA EDUCAÇÃO BÁSICA: ESTADO DA ARTE E TRABALHOS CORRELATOS

De acordo com Medeiros *et al.* (2015), a pesquisa bibliográfica é uma das primeiras etapas do método científico, compreendendo uma investigação acerca de assuntos semelhantes àquele que está sendo pesquisado e contribuindo para evitar a duplicidade de trabalhos. Nesse sentido, com o objetivo de reforçar a originalidade de uma pesquisa, é comum a realização de estudos de revisão, entre eles a revisão sistemática. Segundo Sampaio e Mancini (2007), esta é uma metodologia de pesquisa que utiliza como fonte de dados a literatura científica sobre determinado tema e disponibiliza um resumo das evidências que se relacionam com um dado recorte. Por sua vez, a estruturação de tal recorte se dá a partir de uma definição clara do que se pretende pesquisar, da aplicação de métodos explícitos e sistematizados de busca e da apreciação crítica dos resultados obtidos. Tal fato permite incorporar um espectro maior de resultados relevantes à pesquisa, em lugar de limitar e enviesar as conclusões obtidas por conta de uma leitura não criteriosa das produções sobre determinado tema.

Para Sampaio e Mancini (2007), as revisões sistemáticas são especialmente úteis para reunir as informações de um conjunto de estudos que foram realizados separadamente e podem apresentar resultados conflitantes e/ou coincidentes. Além disso, a condução de tais revisões possibilita identificar temas que necessitam de evidência ou complementações, auxiliando na orientação para investigações futuras. Considerando a relevância dos fatos mencionados, procedeu-se à realização de uma revisão sistemática de literatura para identificar trabalhos científicos correlatos a esta pesquisa e apresentar elementos que apontem para a originalidade deste trabalho.

Nesta perspectiva, Sampaio e Mancini (2007) detalham um conjunto de passos a serem seguidos para a elaboração do protocolo de pesquisa. Tais passos foram adaptados e sistematizados por Medeiros *et al.* (2015), e são ilustrados no Quadro 3.

Quadro 3: Passos da revisão sistemática.

Passos	Descrição
Passo 1: Definindo a pergunta	O primeiro passo a ser dado no início de qualquer estudo é estabelecer o que se deseja pesquisar. Questões mal formuladas podem conduzir a decisões obscuras sobre o que incluir na revisão posteriormente.
Passo 2: Buscando a evidência	Esta etapa realiza-se em bases de dados eletrônicas (<i>databases</i>) indexadas (a partir da seleção de unitermos, também conhecidos como descritores construídos com as palavras-chave e operadores booleanos AND, NOT, OR etc.).
Passo 3: Revisando e selecionando os estudos	De posse de todos os estudos a serem incluídos, são estabelecidos critérios para determinar a sua validade e se há possibilidade dos resultados possuírem vieses.
Passo 4: Analisando a qualidade metodológica dos estudos	Com base nas semelhanças entre artigos, os dados serão agrupados para a obtenção das conclusões finais (ou da metanálise, se este for o caso). Cada um destes agrupamentos deve - preferencialmente - ser pré-estabelecido previamente, evitando a tendenciosidade.
Passo 5: Apresentando os resultados	Nas etapas finais, a redação dos resultados deve ser feita levando-se em conta a questão norteadora estabelecida no primeiro passo supracitado.

Fonte: Medeiros et al. (2015), adaptado de Sampaio e Mancini (2007).

Para a construção do protocolo de pesquisa para a realização da revisão sistemática, baseamo-nos no conjunto de passos ilustrados no Quadro 3 e, ainda, no *checklist* proposto por sistematizados por Medeiros et al. (2015, p. 102). O Quadro 4 ilustra o protocolo construído.

Quadro 4: Protocolo para a revisão sistemática de literatura.

Sub-etapa	Descrição
<p>1. Determinação dos objetivos e questões de pesquisa</p>	<p>Qual(is) o(s) tema(s) principal(ais) da pesquisa? Pensamento computacional e deficiência visual</p> <p>O que se deseja pesquisar? Tipos de recursos utilizados para o desenvolvimento do pensamento computacional junto a estudantes com deficiência visual da Educação Básica e suas aplicações</p> <p>Quais as questões que se deseja responder através da revisão? Q1: Quais são os recursos que têm sido utilizados para o desenvolvimento do pensamento computacional em estudantes com deficiência visual da Educação Básica? Q2: As propostas apresentadas nos trabalhos analisados são destinadas a que público-alvo (estudantes da Educação Infantil, estudantes do Ensino Fundamental, estudantes do Ensino Médio, responsáveis, professores)? Q3: Que habilidades do pensamento computacional são desenvolvidas junto aos estudantes com deficiência visual nos trabalhos analisados? Q4: Como as habilidades do pensamento computacional são desenvolvidas junto aos estudantes com deficiência visual nos trabalhos analisados?</p>
<p>2. Escolha de bases de dados relevantes</p>	<p>Considerando o tema e as questões de pesquisa elaboradas, quais as bases de dados devem ser utilizadas no estudo? Web of Science, Scopus, IEEE Xplorer, Eric, Wiley Online Library, ACM Digital Library, Scielo e ScienceDirect</p>
<p>3. Determinação de uma expressão de busca</p>	<p>Quais os termos (palavras-chave) que melhor descrevem o objeto? Deficiência visual; cegueira; pensamento computacional; recursos educacionais; Educação Básica.</p> <p>Como integrar os termos em uma expressão de busca através da utilização de operadores lógicos? ("Deficiência visual" OR cegueira) AND "pensamento computacional" AND "recursos educacionais" AND "Educação Básica".</p> <p>Após teste da expressão construída, pode-se concluir que é pertinente à busca? Se não, que expressão é a mais adequada? Após concluir que a expressão inicial não é adequada, chegou-se à conclusão de que uma melhor expressão seria: "computational thinking" AND ("visual impairments" OR "vision impairments" OR "visually impaired" OR blind OR blindness)</p>

Sub-etapa	Descrição
<p>4. Determinação dos critérios de inclusão e exclusão de artigos para a revisão</p>	<p>Que critérios serão utilizados para inclusão e exclusão de artigos na revisão?</p> <p>Critérios de inclusão: CI1: O estudo apresenta recursos ou situações de aprendizagem cuja utilização/exploração por estudantes com deficiência visual é explícita no texto. CI2: O público-alvo dos recursos ou situações de aprendizagem apresentados no artigo são: estudantes da Educação Básica com deficiência visual, ou professores ou responsáveis de estudantes da Educação Básica com deficiência visual. CI3: O estudo aborda o pensamento computacional.</p> <p>Critérios de exclusão: CE1: O resultado retornado está entre um dos seguintes formatos: livro, anais de evento, resumo de anais de evento, resumo expandido, editorial ou artigo de opinião. CE2: O estudo é um artigo de revisão. CE3: A versão completa do artigo não foi encontrada. CE4: O artigo não foi publicado entre 2013 e 2023.</p>
<p>5. Realização da busca e filtragem dos resultados</p>	<p>Realização da busca utilizando a expressão construída em todas as bases de dados, e filtragem dos resultados obtidos utilizando os critérios de inclusão e exclusão</p>
<p>6. Sistematização da bibliografia</p>	<p>Construção de uma planilha sistematizando os resultados obtidos na etapa anterior e armazenamento dos artigos incluídos na revisão</p>
<p>7. Análise dos artigos e elaboração do relatório</p>	<p>Análise dos artigos incluídos, identificação de elementos que possam ser utilizados para responder às questões de pesquisa elaboradas na etapa 1 e escrita do relatório apresentando análises, interpretações e conclusões</p>

Fonte: elaborado pelo autor (2024).

Na primeira etapa da revisão, conforme ilustrado no Quadro 4, foram inicialmente determinados os objetivos da busca, para então serem construídas as questões de pesquisa a serem resolvidas a partir da revisão. Em primeiro lugar, desejava-se identificar os recursos que são comumente utilizados para o desenvolvimento do pensamento computacional em estudantes com deficiência visual na Educação Básica. No entanto, para além de uma simples identificação, entendemos que seria importante analisar como as propostas apresentadas nos trabalhos foram implementadas, assim como o público-alvo das mesmas. Desse modo, foram elaboradas as seguintes questões de pesquisa: Q1: “*Quais são os recursos que têm sido utilizados para o desenvolvimento do pensamento computacional junto a estudantes com deficiência visual da Educação Básica?*”; Q2: “*As propostas apresentadas nos trabalhos analisados são destinadas a*

que público-alvo (estudantes da Educação Infantil, estudantes do Ensino Fundamental, estudantes do Ensino Médio, responsáveis, professores)?”; Q3: “Que habilidades do pensamento computacional são desenvolvidas junto aos estudantes com deficiência visual nos trabalhos analisados?”; e Q4: “Como as habilidades do pensamento computacional são desenvolvidas junto aos estudantes com deficiência visual nos trabalhos analisados?”.

Com o objetivo de realizar uma busca abrangente, optou-se pela utilização de uma expressão de busca que contemplasse diferentes termos associados à deficiência visual na língua inglesa. Assim, a expressão usada nas diferentes bases de dados foi "computational thinking" AND ("visual impairments" OR "vision impairments" OR "visually impaired" OR blind OR blindness). Optou-se, ainda, pela busca de artigos que apresentassem os termos mencionados em qualquer parte do estudo. Para a condução das buscas, foram utilizadas as seguintes bases: Web of Science, Scopus, IEEE Xplorer, Eric, Wiley Online Library, ACM Digital Library, Scielo e ScienceDirect.

Em relação aos critérios para inclusão de artigos na análise, foram utilizados os seguintes: CI1: “*O estudo apresenta recursos ou situações de aprendizagem cuja utilização/exploração por estudantes com deficiência visual é explícita no texto*”; CI2: “*O público-alvo dos recursos ou situações de aprendizagem apresentados no artigo são: estudantes da Educação Básica com deficiência visual, ou professores ou responsáveis de estudantes da Educação Básica com deficiência visual*”; e CI3: “*O estudo aborda o desenvolvimento do pensamento computacional*”. Já os critérios de exclusão foram elaborados da seguinte forma: CE1: “*O resultado retornado está entre um dos seguintes formatos: livro, anais de evento, resumo de anais de evento, resumo expandido, editorial ou artigo de opinião*”; CE2: “*O estudo é um artigo de revisão*”; CE3: “*A versão completa do artigo não foi encontrada*”; e CE4: “*O artigo não foi publicado entre 2013 e 2023*”.

Após a aplicação dos critérios de inclusão e exclusão, o total de artigos selecionados em cada base (representado por n), bem como o número de trabalhos excluídos por não atenderem a tais critérios, são detalhados na Tabela 1.

Tabela 1: Quantitativo de artigos levantados em cada base, e número de trabalhos excluídos de acordo com os critérios de inclusão e exclusão.

	Scopus n=29	Web of Science n=16	Eric n=0	Science Direct n=67	Wiley Online Library n=46	Scielo n=0	ACM Digital Library n=215	IEEE Xplorer n=2
CI1: O estudo apresenta recursos ou situações cuja utilização/exploração por estudantes com deficiência visual é explícita no texto	6	5	-	53	46	-	133	1
CI2: O público-alvo dos recursos ou situações de aprendizagem apresentados no artigo são: estudantes da Educação Básica com deficiência visual, ou professores ou responsáveis de estudantes da Educação Básica com deficiência visual.	1	1	-	-	-	-	8	1
CI3: O estudo aborda o desenvolvimento do pensamento computacional	1	1	-	1	-	-	4	-
CE1: O resultado retornado está entre um dos seguintes formatos: livro, anais de evento, resumo de anais de evento, resumo expandido, editorial ou artigo de opinião.	8	1	-	5	-	-	43	-
CE2: O estudo é um artigo de revisão	-	-	-	7	-	-	5	-
CE3: A versão completa do artigo não foi encontrada	3	-	-	-	-	-	3	-
CE4: O artigo não foi publicado entre 2013 e 2023	-	-	-	-	-	-	-	-
Total de artigos selecionados	10	8	0	1	0	0	19	0

Fonte: elaborado pelo autor (2024).

Após a eliminação de 18 artigos duplicados em diferentes bases, chegou-se a um quantitativo final de 20 produções. Para uma melhor sistematização dos resultados, cada artigo selecionado foi identificado através de um código A[n], com $1 \leq n \leq 20$. No Quadro 5 são apresentados a identificação, os títulos, o tipo de estudo, a autoria e o ano de publicação de todos os trabalhos que atenderam aos critérios de inclusão e exclusão.

Quadro 5: Trabalhos selecionados na revisão sistemática.

Identificação	Título	Tipo	Autoria e ano de publicação
A1	Fostering collaboration with asymmetric roles in accessible programming environments for children with mixed-visual-abilities	Artigo publicado em anais de evento	ROCHA et al., 2021
A2	Melodic - teaching computational thinking to visually impaired kids	Artigo publicado em anais de evento	COSTA; ARAÚJO; HENRIQUES, 2021

Identificação	Título	Tipo	Autoria e ano de publicação
A3	Accembly at home: accessible spatial programming for children with visual impairments and their families	Artigo publicado em anais de evento	ROCHA <i>et al.</i> , 2021
A4	TACTOPI: a playful approach to promote computational thinking to visually impaired children	Dissertação	ABREU, 2021.
A5	Conceptual learning through accessible play: Project Torino and computational thinking for blind children in India	Artigo publicado em anais de evento	INDIA <i>et al.</i> , 2020
A6	Computational thinking as play: experiences of children who are blind or low vision in India	Artigo publicado em anais de evento	INDIA <i>et al.</i> , 2019
A7	Accessibility, making and tactile robotics: facilitating collaborative learning and computational thinking for learners with visual impairments	Artigo publicado em anais de evento	SEO; RICHARD, 2018
A8	Inclusive'R'Stories: an inclusive storytelling activity with an emotional robot	Artigo publicado em anais de evento	ANTUNES <i>et al.</i> , 2022
A9	Enabling collaboration in learning computer programming inclusive of children with vision impairments	Artigo publicado em anais de evento	THIEME <i>et al.</i> , 2017
A10	Exploring accessible programming with educators and visually impaired children	Artigo publicado em anais de evento	PIRES <i>et al.</i> , 2020
A11	Tangible music programming blocks for visually impaired children	Artigo publicado em anais de evento	SABUNCUOGLU, 2020
A12	A tangible math game for visually impaired children	Artigo publicado em anais de evento	PIRES <i>et al.</i> , 2019
A13	CodeRhythm: a tangible programming toolkit for visually impaired students	Artigo publicado em anais de evento	RONG <i>et al.</i> , 2020
A14	Robots for inclusive play: co-designing an educational game with visually impaired and sighted children	Artigo publicado em anais de evento	METATLA <i>et al.</i> , 2020
A15	Learning maths with a tangible user interface: lessons learned through participatory design with children with visual impairments and their educators	Artigo publicado em periódico	PIRES <i>et al.</i> , 2022
A16	Enhanced robotics! Improving building and programming learning experiences for students with visual impairments	Artigo publicado em anais de evento	LUDI; BERNSTEIN; MUTCH-JONES, 2018
A17	Digital technology as support of cognitive processes of people with visual impairments	Artigo publicado em anais de evento	SPINAROVA; VACHALOVA, 2021
A18	Bonk: accessible programming for accessible audio games	Artigo publicado em anais de evento	KANE; KOUSHIK; MUEHLBRADT, 2018

Identificação	Título	Tipo	Autoria e ano de publicação
A19	LEGOWorld: repurposing commodity tools & technologies to create an accessible and customizable programming environment	Artigo publicado em anais de evento	CARDOSO et al., 2021
A20	StoryBlocks: a tangible programming game to create accessible audio stories	Artigo publicado em anais de evento	KOUSHIK; GUINNESS; KANE, 2019

Fonte: elaborado pelo autor (2024).

Todos os trabalhos listados foram lidos na íntegra com o objetivo de responder às questões elaboradas no protocolo da revisão sistemática. Os resultados obtidos de acordo com cada questão de pesquisa são apresentados a seguir.

Q1: Quais são os recursos que têm sido utilizados para o desenvolvimento do pensamento computacional em estudantes com deficiência visual da Educação Básica?

A partir da leitura dos artigos levantados foi possível identificar uma série de recursos que vêm sendo utilizados para promover o desenvolvimento do pensamento computacional em pessoas com deficiência visual. No Quadro 6 são listados todos aqueles que são empregados nos trabalhos analisados, bem como são apontados em quais artigos eles são usados. Em seguida, são apresentadas características de cada um dos recursos identificados nas produções.

Quadro 6: Recursos utilizados para o desenvolvimento do pensamento computacional, identificados nos trabalhos analisados.

Recurso	Trabalhos em que o recurso foi empregado	Quantidade de trabalhos
ACCembly	A3	1
KIBO	A7	1
Torino	A5, A6, A9	3
iCETA	A12, A15	2
Bonk	A18	1
CodeRhythm	A13	1
Robô Ozobot	A8, A14, A17, A19	4
Robô Dash	A1, A3, A10	3

Robô Super DOC	A10	1
Robô BlueBot	A17	1
Robô BeeBot	A17	1
Blockly	A10	1
Osmo Coding AWBIE	A10	1
PUZZLETS	A10	1
LEGO Mindstorms	A16	1
LEGO World	A19	1
Melodic	A2	1
“Tangible music programming blocks”	A11	1
TACTOPI	A4	1
Storyblocks	A20	1

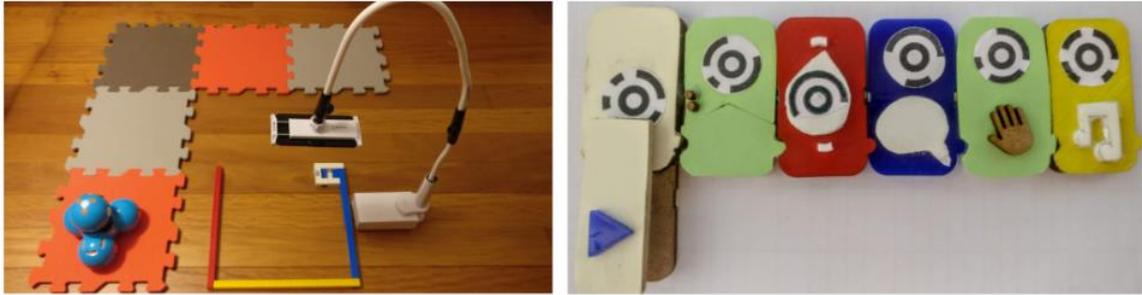
Fonte: elaborado pelo autor (2024).

ACCembly

O *ACCembly* é um ambiente de programação baseado em blocos que permite que crianças com deficiência visual realizem atividades de programação espacial (seguir um determinado caminho, evitar obstáculos, buscar recompensas), organizando blocos tangíveis para programar um robô multimodal. É composto por blocos manipuláveis, um robô, espumas para construir um mapa e um aplicativo *Android*. O aplicativo usa a câmera do dispositivo para reconhecer a sequência de blocos no espaço de trabalho, interpretar os blocos conectados e enviar instruções para o robô (Rocha et al., 2021).

O recurso dispõe de três tipos diferentes de blocos: os de ação (amarelo e azul), os de direção e os de loop. Os blocos de ação podem ser de dois tipos: de dança (amarelos), que têm uma nota musical em relevo e fazem o robô girar enquanto emite o som de uma onomatopeia; e de “oi”, que possui um balão de fala em relevo e faz o robô dizer “oi”. Os blocos de direção (vermelhos) fazem com que o robô se movimente uma unidade de distância, que corresponde a um bloco no mapa de espuma, e possuem uma seta em relevo, que determina o sentido em que o objeto irá se movimentar. Já os blocos de loop (verdes) são usados para repetir instruções (Rocha et al., 2021).

Figura 1: ACCembly e seus componentes.



Fonte: Rocha et al. (2021).

KIBO

KIBO é um kit robótico baseado em blocos manipuláveis de madeira, projetado para o desenvolvimento de conceitos de programação por meio de jogos. Para isso, utiliza-se um leitor de código de barras que identifica as instruções de cada bloco, eliminando a necessidade de uso de computadores. O robô, que pode ser personalizado pelo usuário possibilitando a criação de personagens, oferece experiências lúdicas para crianças pequenas, na medida em que desenvolvem habilidades do pensamento computacional, de codificação e engenharia através da construção e programação de uma estrutura robótica (Papadakis, 2020).

O kit permite o uso de sequências, loops e variáveis, assim como motores, sensores e outros materiais artesanais, permitindo que crianças aprendam sobre algoritmos e redesenho de processos, entre outras possibilidades. O recurso, que foi desenvolvido durante anos contando com a colaboração de estudantes e professores, é recomendado para crianças com idades entre 4 e 7 anos.

Figura 2: KIBO e blocos manipuláveis de madeira.



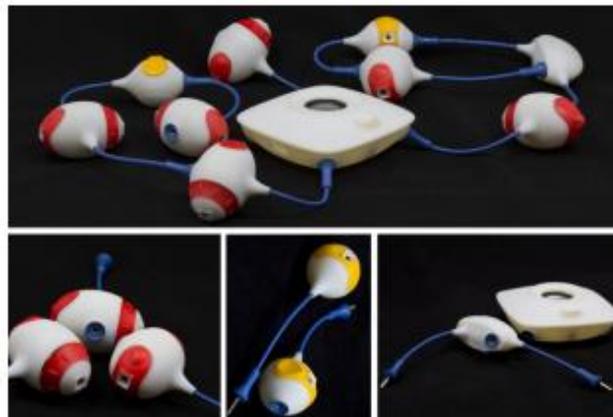
Fonte: Abreu (2021).

Torino

Torino é uma linguagem de programação física para o ensino de programação e do pensamento computacional para crianças com idade entre 7 e 11 anos. O recurso consiste em um ambiente de programação tangível composto por diferentes peças e um hub, que quando conectados fisicamente constituem programas de computador que geram músicas ou histórias.

O design do recurso é baseado em *feedbacks* auditivos e objetos manipuláveis que podem ser identificados através do tato, possibilitando a criação de programas independentemente do grau de visão do usuário. Para criar os programas, basta conectar os objetos manipuláveis (que podem ser do tipo tocar, pausar, e *loop*), o que permite a geração de músicas ou histórias na linguagem *Sonic Pi* (Thieme et al.2017).

Figura 3: Peças do Torino e hub principal.



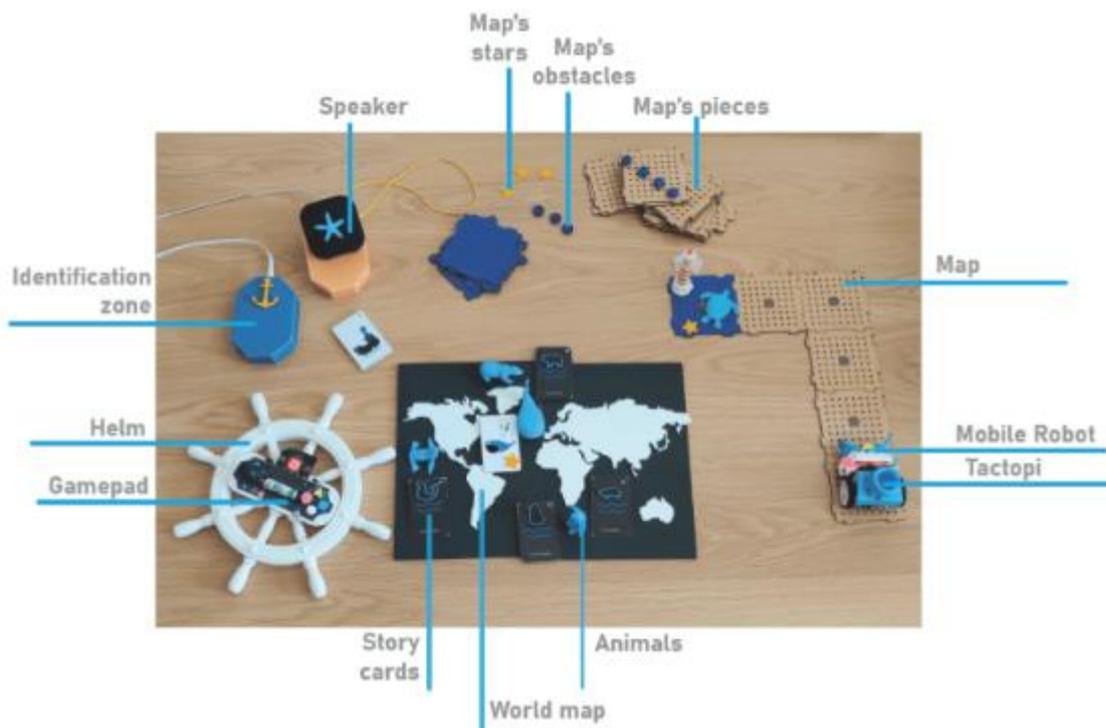
Fonte: Thieme et al. (2017).

TACTOPI

O *TACTOPI* é um sistema que consiste em um ambiente tangível que promove o desenvolvimento de habilidades de navegação e enriquece experiências sensoriais por meio de sons e elementos visuais e táteis. Trata-se de um jogo náutico exploratório e inclusivo que possibilita a aprendizagem de conceitos introdutórios de programação e do pensamento computacional através de atividades lúdicas envolvendo contação de histórias, promovendo educação ambiental para crianças com deficiência visual entre 4 e 7 anos de idade.

O nome *TACTOPI* é uma combinação das palavras “tato” (*tactus*, no latim), e polvo (*octopus*), na medida em que o recurso almeja fornecer uma experiência tátil enriquecedora para crianças com deficiência visual. Além disso, *TACTOPI* é o nome do personagem principal do jogo, um polvo com um excelente tato. O recurso é composto por 25 peças modulares, com pontos perfurados que permitem a customização e criação de diferentes caminhos; um leme a ser controlado pelo usuário; cinco animais impressos em 3D com etiquetas inteligentes; cartões de missões com figuras táteis e etiquetas inteligentes; e um mp3 e um alto-falante para reproduzir sons.

Figura 4: Principais elementos do *TACTOPI*.



Fonte: Abreu (2021).

StoryBlocks

O *StoryBlocks* é um jogo tangível baseado em blocos que permite que pessoas cegas aprendam conceitos básicos de programação através da criação de histórias em áudio usando blocos de código (Abreu, 2021). O sistema compreende um conjunto de blocos tangíveis, um espaço de trabalho aumentado e um *software* para interpretação e reprodução dos programas construídos pelos usuários.

No recurso, cada bloco representa um componente de uma história interativa, incluindo personagens, ações e estruturas de controle utilizadas nos programas. Cada objeto contém um símbolo tátil e uma etiqueta visual, não apresentando elementos eletrônicos. Os programas devem ser criados em um espaço denominado área de trabalho, cujos limites são delimitados por bordas táteis. As imagens do programa construído são capturadas por uma webcam conectada a um computador portátil, e interpretadas por um *software* específico baseado na linguagem Python. Por fim, o computador emite uma saída em áudio que corresponde ao código criado pelo usuário.

Figura 5: Blocos utilizados no protótipo inicial do *StoryBlocks*, incluindo personagens, ações e estruturas de controle.



Fonte: Koushik, Guinness e Kane (2019).

iCETA

O *iCETA* é um sistema interativo inclusivo para a aprendizagem da matemática, que permite que crianças se engajem de maneira autônoma em tarefas que envolvam a operação de adição. O recurso é uma adaptação do *CETA*, um sistema aberto, portátil, de baixo custo e realidade mista, criado para utilização em ambientes escolares (Pires et al., 2019, 2022).

O sistema consiste em um conjunto de blocos que são detectados por uma câmera redirecionada para a área de trabalho através de um espelho. Os blocos, que representam números de um a cinco, variam em tamanho (de forma proporcional), textura e cor. Cada bloco é composto por um número de componentes circulares que pode variar de um a cinco, cada um deles apresentando um marcador *TopCode* (Horn, 2012) em sua superfície superior. Além disso, existem marcações táteis para representar as divisões entre cada componente circular.

Os *feedbacks* sonoros dos blocos são emitidos de acordo com a sua quantidade de componentes circulares. Em virtude das diferenças que crianças podem apresentar na identificação do fim de um bloco e início de outro, um parâmetro temporal ajustável foi introduzido. Além disso, outros parâmetros como tom e timbre foram adicionados com o objetivo de possibilitar a representação, através de sons, de diferentes ações nos jogos criados, como dar passos, bater em portas ou derramar poções (Pires et al., 2019). Com isso, é possível criar jogos de diferentes níveis de acordo com a performance do usuário em Matemática. Para utilizar o sistema, o usuário deve dispor os blocos em uma área de trabalho. Estes são reconhecidos através de marcadores *TopCode* (Horn, 2012), e após a decodificação, o computador fornece um retorno auditivo.

Figura 6: (A) Representação do *iCETA* com fone, computador, espelho, área de trabalho e blocos. (B) Caixa com blocos representando os números de 1 a 5.

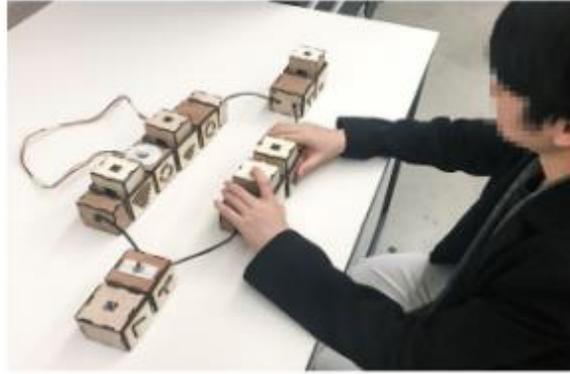


Fonte: Pires et al. (2022).

CodeRhythm

O *CodeRhythm* é um kit tangível de ferramentas de programação criado com o objetivo de envolver usuários com deficiência visual na aprendizagem de conceitos básicos de programação a partir da criação de melodias simples (Rong et al., 2020). O kit possui duas categorias de blocos: os silábicos: dó, ré, mi, fá, sol, lá e si, e os blocos de função: tocar, alternar, *loop* e pausar. Cada bloco silábico representa uma nota na melodia, e o usuário pode controlar a duração da nota puxando e empurrando um cubo saliente no topo de cada um desses blocos. Além disso, todos os blocos são conectados por ímãs, o que segundo estudo preliminar realizado pelos desenvolvedores, favorece a utilização por usuários com deficiência visual.

Figura 7: Criação de uma melodia através do CodeRhythm.



Fonte: Rong et al. (2020).

Melodic

O *Melodic* é um recurso capaz de auxiliar no ensino do pensamento computacional a crianças cegas através do uso de jogos envolvendo música, possibilitando a resolução de problemas genéricos ou de programação. O recurso é composto por um conjunto de blocos táteis com diferentes formatos, texturas e significados. Os blocos especiais, que possuem uma superfície lisa, podem ser dos seguintes tipos: os de controle, que demarcam o início e o fim de um algoritmo ou instrução; os de velocidade, que determinam o intervalo de tempo entre notas musicais; ou os de instrumento, que determinam qual instrumento musical deverá ser tocado. Os blocos numéricos, que possuem uma superfície que apresenta linhas, representam números de um a dez e são utilizados na determinação do número de iterações em um *loop* ou da quantidade de vezes que uma nota deve ser repetida. Já os blocos de notas musicais, que possuem uma textura áspera, representam as notas musicais. Utilizando esses blocos táteis, o usuário pode criar algoritmos para representar melodias, sendo capaz de aumentar a complexidade das criações na medida em que desenvolve habilidades do pensamento computacional (Costa; Araújo; Henriques, 2021).

O recurso deve ser utilizado em conjunto com um *smartphone* ou um *tablet*. Costa, Araújo e Henriques (2021) ressaltam que, para evitar incompatibilidades entre dispositivos *Apple* e *Android*, foi utilizada no desenvolvimento do projeto a plataforma *React Native*, que promove paridade entre as duas plataformas anteriores através do mesmo código *JavaScript*. A utilização do recurso consiste na formação de uma sequência de blocos com o objetivo de produzir um determinado som desejado. Cada bloco possui um QRcode em sua parte traseira, de modo que utilizando a câmera do

dispositivo móvel, é possível decodificar a sequência formada através de um aplicativo, criando um som que será emitido pelo dispositivo eletrônico.

Figura 8: Usuário realizando a leitura de uma sequência de blocos.



Fonte: Costa, Araújo e Henriques (2021).

Bonk

O *Bonk* é um ambiente de programação acessível que permite a criação de *audiogames* interativos utilizando a linguagem de programação JavaScript. De acordo com os desenvolvedores, o ambiente fornece uma interface estruturada e simplificada para a criação de interações auditivas complexas. Os programas são renderizados de várias formas: como texto, como áudio renderizado em um leitor de tela ou através de um sistema de conversão de texto em fala.

Figura 9: Exemplo de programa criado através do *Bonk*.

```
function run(game) {
  game.speak("You walk down a narrow stairwell.");
  var sound = new Sound("stairs");
  sound.play();
  game.speak("There is a big crowd here. On your left is the concession stand. The line is very long. On the right is the way to the restroom. It is completely packed.");
  var crowd = new Sound("crowd");
  crowd.play();
  game.speak("You probably don't have time for either of these, so you better get to your seat! Another stairway ahead of you goes to the seating area.");
  game.speak("Press w to walk up the stairs, and s to go back.");
}

function on_key_press(key) {
  if (key == 'w') {
    game.go_to_room("Either_one_seats");
  } else if (key == 's') {
    game.go_to_room("Either_one_first_choice v.2");
  }
}
}
```

Fonte: Kane, Koushik e Muehlbradt (2018).

Robô Dash

O Dash é um robô educativo programável que pode ser utilizado para auxiliar crianças de 5 a 12 anos a adquirirem suas primeiras noções de programação enquanto brincam e interagem com o recurso. O recurso se move, emite sons e brilha, responde a comandos de voz ou comandos baseados em um aplicativo através de um dispositivo móvel. Além disso, inclui microfones, giroscópio, transmissores infravermelho e sensores de longo alcance (Papadakis, 2020).

Figura 10: Robô Dash e Dot, seu ajudante.



Fonte: Papadakis (2020).

Robô Bee-Bot

O *Bee-Bot* é um dispositivo robótico programável que se assemelha a uma abelha devido ao fato de possuir listras amarelas e pretas. Além de ser atrativo para crianças, o recurso se torna adequado para usuários com deficiência visual em virtude de seu contraste de cores. O robô permite desenvolver diversas atividades de orientação espacial e é equipado com sinais sonoros que são acionados quando se introduz uma ação ou se conclui algum percurso. O recurso é capaz de memorizar até quarenta passos e se movimenta de maneira ideal em uma superfície quadriculada (Spinarova; Vachalova, 2021).

Figura 11: Robô *Bee-Bot*.



Fonte: Papadakis (2020).

Robô *Blue-Bot*

O *Blue-Bot* é uma versão atualizada do *Bee-Bot*, que assim como o anterior, pode ser programado para desenvolver diferentes atividades de orientação espacial (Pinto; Osorio; Monteiro, 2017). No entanto, também pode ser conectado a um *tablet* ou computador via *Bluetooth*. Desse modo, uma criança pode utilizar aplicativos gratuitos para programar remotamente o recurso.

Figura 12: Robô *Blue-Bot*.



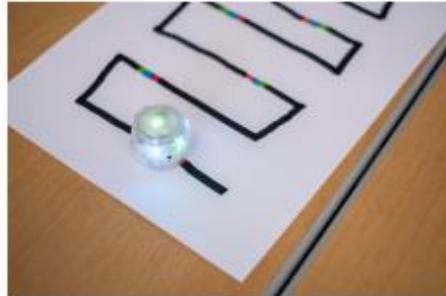
Fonte: Papadakis (2020).

Robôs *Ozobot*

O *Ozobot Bit* e o *Ozobot Evo* são robôs que podem se mover, emitir sons e piscar em diferentes cores, além de possuírem sensores que podem identificar a cor da superfície que estão percorrendo. Apresentando controles simples e intuitivos, os robôs *Ozobot* podem ser programados através do aplicativo *OzoBlockly*, cujo ambiente de programação segue o padrão baseado em blocos. Enquanto o *Ozobot Bit* é mais adequado para crianças menores (6 a 10 anos), o *Ozobot Evo*, com LEDs adicionais,

capacidade de emitir sons, *bluetooth*, sensores infravermelhos de distância e alguns outros recursos adicionais, é um pouco mais complexo.

Figura 13: Robô Ozobot se movendo sobre uma linha.



Fonte: Körber et al. (2021).

Robô Super DOC

O *Super DOC*, um robô com olhos grandes e brilhantes, pode ser utilizado para uma introdução aos conceitos básicos de programação às crianças de maneira criativa, gradual e divertida. O robô pode ser programado livremente ou com o apoio de cartas que sugerem associações lógicas simples, podendo também ser usado em desafios com dois jogadores. O *Super DOC* também pode interpretar quatro personagens diferentes, transformando-se em dragão, mágico, fada e cavaleiro, cada um com características próprias e missões a cumprir. O recurso possibilita o desenvolvimento das habilidades lógicas e de resolução de problemas, estimulando a imaginação e a criatividade.

Figura 14: Super DOC.



Fonte: Super DOC (2023).

LEGO Mindstorms

O *LEGO Mindstorms* é um kit de montagem que contém blocos e uma unidade de controle programável, permitindo a construção de diversos tipos de robôs. Com o recurso, é possível montar dispositivos robóticos, programar seu movimento e manusear sensores e motores sem a necessidade de focar em aspectos técnicos dos elementos mecânicos e elétricos envolvidos (Afari; Khine, 2017).

Os processos de montagem e colocação do robô em funcionamento envolvem a compreensão elementar de princípios físicos e de design, além de habilidades básicas de programação. Desse modo, o recurso representa uma ferramenta com grandes potencialidades no desenvolvimento de habilidades relacionadas ao pensamento computacional (Afari; Khine, 2017).

Figura 15: Robôs criados com o *LEGO Mindstorms*.



Fonte: Mindstorms (2023).

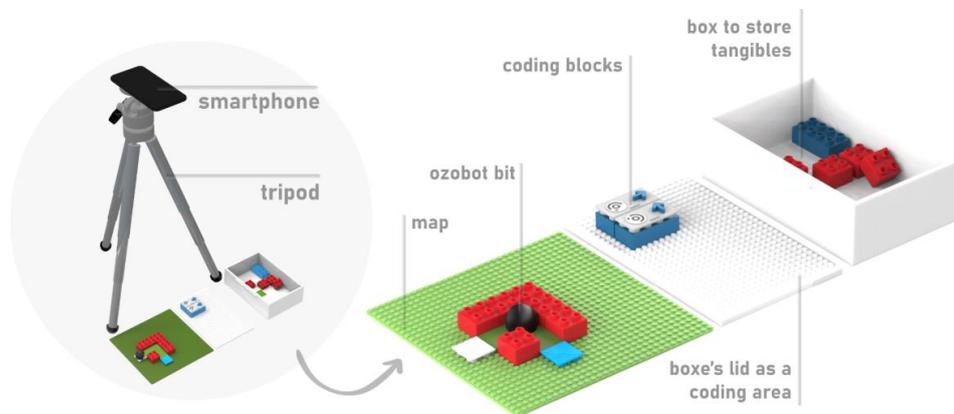
LEGOWorld

O *LEGOWorld* foi desenvolvido para auxiliar a aprendizagem de conceitos de programação e o desenvolvimento do pensamento computacional em crianças com deficiência visual. Para aprender tais conceitos, as crianças devem participar de um jogo com vários níveis, simulando as aventuras de um robô protagonista. O sistema é composto por um robô do tipo *Ozobot Bit*, um mapa físico *LEGO* sobre o qual o robô se move, blocos tangíveis *LEGO* para construir sequências de instruções para os movimentos do robô e um aplicativo móvel (Cardoso et al., 2021).

A sequência de blocos tangíveis que formam o código deve ser montada verticalmente, de cima para baixo, sobre uma placa branca que serve como área para

construção. Os blocos possuem relevos em 3D que permitem que crianças com deficiência visual identifiquem instruções como frente, direita, esquerda, início do *loop*, fim do ciclo ou começar. Além disso, no topo de cada bloco há um adesivo *TopCode* para identificação do código construído através do aplicativo móvel. Já o ambiente sobre o qual o robô se move é composto por uma placa *LEGO* que serve como base, blocos *LEGO* que funcionam como obstáculos, e peças impressas em 3D que funcionam como um caminho para a movimentação do robô (Cardoso et al., 2021).

Figura 16: Visão geral do LEGOWorld com smartphone, tripé, mapa físico, blocos tangíveis e caixa com tampa.

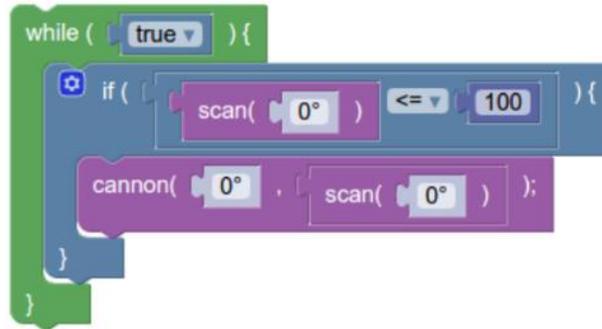


Fonte: Cardoso et al. (2021).

Blockly

Blockly é uma biblioteca de código aberto que possibilita a adição de programação baseada em blocos a um aplicativo. Inicialmente disponibilizado em maio de 2012, o recurso fornece um editor de blocos e uma estrutura para geração de código em linguagens baseadas em texto. Enquanto uma biblioteca, o *Blockly* não constitui uma linguagem de programação em si, nem um aplicativo pronto para o usuário final. A ferramenta, na realidade, permite a criação de novas linguagens baseadas em blocos, na medida em que fornece uma gramática e uma representação própria para a programação construída por desenvolvedores em seus aplicativos (Pasternak; Fenichel; Marshall, 2017).

Figura 17: Blocos de código de um jogo construído através do Blockly.



Fonte: Pasternak, Fenichel e Marshall (2017).

Osmo Coding Awbie

O *Osmo Coding Awbie* é um brinquedo híbrido que combina um aplicativo de jogo e 19 blocos tangíveis magnéticos de codificação *Osmo*, incluindo ações como andar, pular, agarrar e estruturas como loop e se. O aplicativo de codificação *Awbie* é compatível exclusivamente com dispositivos *Apple iPad* (iOS). Na utilização do sistema, crianças usam os comandos de codificação para controlar *Awbie*, um personagem que adora morangos e deve coletá-los em um mundo cheio de caminhos (Papadakis, 2020).

Figura 18: Osmo Coding Awbie.



Fonte: Papadakis (2020).

Music Programming Blocks

Plataforma musical tangível e acessível para crianças com deficiência visual, que tem por objetivo o ensino de conhecimentos básicos de programação através da criação musical. Ordenando blocos tangíveis em uma estrutura física, as crianças podem criar

melodias utilizando diferentes oitavas, ritmos e sons. O *design* do recurso foi elaborado priorizando três considerações: evitar o uso extensivo do alfabeto Braille (excessivo tempo de leitura), não utilizar muitas modalidades de *feedback* para o reconhecimento dos blocos (possibilidade de confusão), e usar modelos básicos e abstratos para o formato dos blocos, em lugar de ícones detalhados (facilitar o reconhecimento).

Para o protótipo do recurso foram elaborados vinte e cinco blocos tangíveis de papelão, além de um aplicativo *Android*. Os blocos possuem formatos variados e diferentes detalhes na superfície, permitindo a diferenciação entre os distintos elementos de codificação. Cada bloco possui um adesivo NFC (*Near Field Communication*) no seu interior, cuja leitura é realizada através do aplicativo em um telefone móvel. Por fim, os blocos são dispostos pelo usuário em uma superfície perfurada, de modo que para que a música seja reproduzida, basta mover o telefone bloco por bloco até o de reprodução, quando a melodia finalmente é tocada.

Figura 19: Leitura de uma sequência de blocos através de um celular.



Fonte: Sabuncuoglu (2020).

Puzzlets

Puzzlets é um acessório *Bluetooth* para computador que permite explorar jogos relacionados à codificação, matemática ou teoria das cores, combinando elementos tradicionais de quebra-cabeças com tecnologia digital interativa. O recurso inclui jogos interativos para trabalhar habilidades básicas de programação, matemática e arte, estimulando o aprendizado visual e as habilidades de pensamento crítico por meio de uma abordagem prática e lúdica (Puzzlets, 2023a).

Figura 20: Puzzlets.



Fonte: Puzzlets (2023b).

Síntese dos recursos identificados

Como colocado anteriormente, o Quadro 6 evidencia uma grande variedade de recursos utilizados para o desenvolvimento do pensamento computacional em estudantes com deficiência visual. Além disso, em diferentes trabalhos mais de um recurso foi utilizado, o que aponta para a possibilidade de exploração simultânea das potencialidades de diferentes ferramentas.

A análise do Quadro 6 revela que o uso de robôs é um exemplo de utilização conjunta de recursos no contexto dos trabalhos analisados. É possível observar que 40% dos estudos (A1, A3, A7, A8, A10, A14, A17 e A19) exploraram ao menos um tipo de robô disponível comercialmente (*KIBO*, *Dash*, *Ozobot*, *SuperDoc*, *BlueBot* ou *BeeBot*). Em particular, os robôs *Ozobot* e *Dash* foram utilizados em 20% (A8, A14, A17 e A19) e 15% (A1, A3 e A10) dos estudos, respectivamente. No entanto, muitos dos trabalhos que utilizam algum tipo de robô exploram algum outro recurso simultaneamente, seja outro tipo de robô, seja um recurso de outra natureza (A3: Robô *Dash* + *ACCembly*; A10: Robô *Dash* + *Blockly* + *Osmo Coding AWBIE* + *PUZZLETS*; A17: Robô *Ozobot* + Robô *BlueBot* + Robô *BeeBot*; A19: Robô *Ozobot* + *LEGOWorld*). Cabe mencionar, ainda, o uso de kits que possibilitam a construção de robôs (*LEGO Mindstorms*), bem como a elaboração própria de robôs (*TACTOPI*).

Outro ponto de destaque é o fato de que muitos trabalhos exploram ambientes de programação desenvolvidos pelos próprios pesquisadores (A2, A3, A4, A11, A12, A13, A15, A18, A20). Tal fato evidencia o empenho de pesquisadores na construção de novos

recursos e estratégias para o desenvolvimento do pensamento computacional em estudantes com deficiência visual. Cabe, no entanto, refletir acerca da aplicabilidade de tais experimentos em contextos concretos de aprendizagem na Educação Básica. Por um lado, a condução de propostas que utilizam robôs e outros kits disponíveis no mercado muitas vezes esbarra nas limitações estruturais e financeiras enfrentadas pelas escolas, em especial na rede pública de ensino. Por outro lado, a condução de experimentos e situações de aprendizagem desenvolvidas pelos próprios pesquisadores muitas vezes se torna inviável pela dificuldade em obter ou reproduzir recursos que não se encontram disponíveis, e cuja construção não é simples.

Por fim, em quase todos os arquivos analisados foi explorado algum recurso ou ambiente de programação tangível, ou algum material manipulável associado a algum recurso de programação através de texto ou áudio. Em contextos que envolvem aprendizes com deficiência visual, o uso de recursos desse tipo é importante na medida em que possibilita a exploração de outros sentidos para além da visão. No entanto, apenas um dos trabalhos (A18) explorou uma linguagem de programação baseada em texto. Além disso, tal linguagem foi desenvolvida pelos próprios pesquisadores. Tal fato indica a escassez de trabalhos que explorem as possibilidades das linguagens de programação baseadas em texto no desenvolvimento do pensamento computacional em estudantes com deficiência visual da Educação Básica.

Q2: As propostas apresentadas nos trabalhos analisados são destinadas a que público-alvo (estudantes da Educação Infantil, estudantes do Ensino Fundamental, estudantes do Ensino Médio, responsáveis, professores)?

Para responder à segunda questão elaborada, foram identificados os objetivos dos artigos selecionados, bem como os participantes de cada uma das pesquisas analisadas. Os resultados obtidos são apresentados no Quadro 7.

Quadro 7: Objetivo e participantes de cada pesquisa

Trabalho	Objetivo do trabalho	Participantes da pesquisa
A1	Apresentar um ambiente de programação que, através da designação de papéis distintos e informações assimétricas a diferentes crianças, promove a realização de atividades colaborativas relacionadas ao pensamento computacional	Doze professores da Educação Especial, dois responsáveis, um pesquisador e um trabalhador de instituição envolvida com a reintegração de pessoas com deficiência visual (idades não mencionadas)
A2	Apresentar um recurso de aprendizagem, o <i>Melodic</i> , capaz de auxiliar no ensino do pensamento computacional a crianças cegas através de jogos, possibilitando a resolução de problemas genéricos ou de programação através da abordagem	Crianças com deficiência visual (idades e nível de escolarização não mencionados)
A3	Analisar se o <i>ACCembly</i> pode promover o desenvolvimento do pensamento computacional, se o recurso possibilita que crianças com deficiência visual se envolvam em tarefas de programação espacial, quais aspectos da ferramenta são efetivos no engajamento de crianças e seus responsáveis, e quais os papéis que tais responsáveis assumem na mediação das tarefas propostas em um contexto envolvendo pessoas com habilidades visuais mistas	Sete grupos de responsáveis, com idades entre 34 e 48 anos, de crianças com deficiência visual com idades entre 7 e 14 anos
A4	Apresentar o <i>TACTOPI</i> , um sistema que consiste em um ambiente tangível que proporciona treinamento em habilidades de orientação e enriquece experiências sensoriais usando elementos sonoros, visuais e táteis, e identificar, entre outras questões, se o recurso é eficiente para a introdução do pensamento computacional para crianças com deficiência visual	Doze pesquisadores e dois educadores de diferentes áreas (idades não mencionadas)
A5	Analisar se é possível que crianças com deficiência visual aprendam conceitos do pensamento computacional a partir de uma abordagem que utiliza o Torino como um brinquedo para exploração criativa de música ao invés de uma ferramenta estruturada de ensino	Doze crianças com idades entre 6 e 12 anos
A6	Analisar se é possível apresentar o Torino como um brinquedo voltado para a criatividade utilizando músicas e histórias, e ainda assim introduzir conceitos do pensamento computacional para crianças	Doze crianças de 7 a 13 anos
A7	Analisar como aprendizes com deficiência visual utilizam plataformas tangíveis acessíveis, como isso afeta a eficácia e colaboração de tais aprendizes, e quais são os elementos de design que são essenciais para que haja de fato aprendizagem e colaboração de pessoas com deficiência visual	Nove participantes, sendo eles estudantes de Ensino Médio e jovens adultos com idades entre 15 e 19 anos
A8	Apresentar um sistema interativo e multissensorial de contação de histórias e explorar até que ponto um robô emocional pode ser utilizado para dar promover experiências inclusivas	Dezesseis crianças, com média etária de 8,75 anos.
A9	Compreender como o <i>Torino</i> promove a colaboração de crianças com habilidades visuais mistas, e como isso se relaciona com a aprendizagem de alguns conceitos de programação	Dez crianças com idades entre 7 e 12 anos, apresentando habilidades visuais mistas

Trabalho	Objetivo do trabalho	Participantes da pesquisa
A10	Identificar qualidades e defeitos em acessibilidade de ambientes de programação em contextos educacionais através de um grupo focal com professores da Educação Especial e instrutores de tecnologia da informação	Quatro professores da Educação Especial e dois instrutores de tecnologia da informação com idades entre 37 e 57 anos, e sete crianças com deficiência visual
A11	Apresentar uma plataforma musical tangível e acessível para crianças com deficiência visual que permite o ensino de habilidades básicas de programação através da criação de músicas	Quatorze estudantes com deficiência visual, com idade média de 12.5 anos
A12	Apresentar o <i>iCETA</i> , um sistema interativo inclusivo para a aprendizagem da matemática que permite que crianças se engajem de maneira autônoma em tarefas que envolvam a operação de adição.	Três professores do Ensino fundamental, dois diretores de escolar, um professor de música e um professor de tecnologia da informação, além de onze crianças com deficiência visual entre 5 e 10 anos
A13	Apresentar o <i>CodeRhythm</i> , um kit tangível de ferramentas de programação desenvolvido para engajar pessoas com deficiência visual na aprendizagem de conceitos básicos de programação através da criação de melodias	Uma professora com deficiência visual com 30 anos
A15	Desenvolver uma interface tangível acessível e de baixo custo, além de um jogo de computador, que possibilitasse promover o aprendizado de matemática incluindo crianças com deficiência visual e professores no processo de desenvolvimento	Professores do Ensino fundamental e professores de música e de Braille com deficiência visual (idades não mencionadas), além de onze crianças com deficiência visual do primeiro ano do Ensino Fundamental (idades não mencionadas)
A16	Apresentar esforços realizados no desenvolvimento de ferramentas e adaptações para tornar a unidade curricular de Robótica (currículo americano) acessível a estudantes com deficiência visual, especialmente a partir de adaptações de software e hardware do <i>LEGO Mindstorms</i>	Dezenove estudantes dos Ensinos Fundamental e Médio (idades não mencionadas)
A17	Apresentar atividades construídas no contexto de dois projetos idealizados com o objetivo de utilizar tecnologias digitais para desenvolver processos cognitivos em pessoas com deficiência visual, que envolvem a programação de robôs de modo que possam se movimentar de acordo com um determinado objetivo (alcançar um ponto ou uma imagem, por exemplo).	As atividades são apresentadas, mas não é realizado estudo de caso com participantes.
A18	Apresentar o <i>Bonk</i> , um ambiente de programação acessível que permite a criação de <i>audiogames</i> através da construção de códigos textuais utilizando um subconjunto da linguagem de programação JavaScript.	Dez estudantes com idades entre 14 a 18 anos
A19	Explorar o reaproveitamento de objetos e tecnologias, especialmente blocos <i>LEGO</i> , para a criação de um ambiente de programação multimodal flexível e acessível a crianças com deficiência visual, que pode ser utilizado em diversos desafios relacionados à programação.	Três professores de robótica e três professores da Educação Especial.

Trabalho	Objetivo do trabalho	Participantes da pesquisa
A20	Apresentar o StoryBlocks, um jogo tangível baseado em blocos que permite que pessoas cegas aprendam conceitos básicos de programação através da criação de histórias em áudio.	Cinco estudantes com deficiência visual dos Ensinos Fundamental e Médio, oito professores de pessoas com deficiência visual, dois transcritores de Braille e um funcionário de apoio de uma pré-escola, com idades entre 11 e 65 anos.

Fonte: elaborado pelo autor (2024).

A análise do Quadro 7 revela pluralidade nas características dos participantes dos artigos incluídos na revisão. Observa-se que, de maneira geral, além de estudantes com deficiência visual do Ensino Fundamental e do Ensino Médio, participaram dos estudos professores e responsáveis de estudantes com deficiência visual, pesquisadores, diretores de escolas, instrutores de tecnologia da informação, revisores de Braille e profissionais de apoio administrativo, entre outros.

Entre os trabalhos que envolveram a participação de estudantes, a maior parte se voltou para participantes do Ensino Fundamental, com idades até 14 anos (10 estudos). Em apenas 4 estudos participaram estudantes do Ensino Médio - ou pessoas que apresentavam idades compatíveis com essa etapa de escolarização. Cabe destacar, ainda, que nenhum dos trabalhos analisados teve como participantes crianças que estivessem na Educação Infantil.

No que se refere à natureza das propostas apresentadas, apenas dois estudos apresentam recursos e situações de aprendizagem com foco no desenvolvimento de conteúdos ou habilidades relacionadas à Matemática. Tais estudos exploraram o *iCETA*, recurso que, como mencionado anteriormente, permite que crianças realizem tarefas que envolvam a operação de adição. Desse modo, o recurso explora operações aritméticas elementares, não sendo ideal para situações de aprendizagem voltadas para estudantes do Ensino Médio.

É possível concluir, então, que dentre os estudos que exploram o desenvolvimento do pensamento computacional em estudantes com deficiência visual na Educação Básica, a maior parte se concentra em estudantes do Ensino Fundamental, sendo escassos os estudos voltados para estudantes do Ensino Médio. Além disso, poucos estudos com esse recorte exploram, de maneira específica, situações de aprendizagem

da Matemática, o que indica necessidade de maior desenvolvimento teórico explorando esta temática.

Q3: Que habilidades do pensamento computacional são desenvolvidas junto aos estudantes com deficiência visual nos trabalhos analisados? e Q4: Como as habilidades do pensamento computacional são desenvolvidas junto aos estudantes com deficiência visual nos trabalhos analisados?

Para responder às questões 3 e 4, inicialmente buscou-se identificar como o pensamento computacional é abordado em cada um dos artigos analisados. O Quadro 8 sintetiza os resultados obtidos.

Quadro 8: Como o pensamento computacional é abordado em cada artigo.

Trabalho	Recursos utilizados	Como o pensamento computacional é abordado
A1	Robô Dash	Os autores argumentam que o treinamento do pensamento computacional desde a primeira infância potencializa o desenvolvimento cognitivo e prepara crianças para viverem em uma sociedade altamente computacional. Nesse sentido, propõem o desenvolvimento de um ambiente de programação que, através da designação de papéis distintos e do fornecimento de informações assimétricas para crianças, possibilita a promoção de atividades colaborativas relacionadas ao pensamento computacional. Entre os resultados, colocam que educadores reconheceram as potencialidades do sistema no treinamento do pensamento computacional e na introdução de conceitos de programação na primeira infância.
A2	Melodic	Os autores colocam que estudantes de programação normalmente enfrentam dificuldades devido às práticas tradicionais que em geral são desenvolvidas nos ambientes de ensino, ao invés de receberem treinamento relacionado ao pensamento computacional através de recursos de aprendizagem adequados e adaptados quando necessário. Nesse sentido, apresentam no trabalho um recurso de aprendizagem capaz de auxiliar o ensino de conceitos do pensamento computacional a crianças com deficiência visual, usando jogos baseados em música para tal. Os autores explicitam que, por meio do uso de blocos táteis, o usuário pode criar algoritmos para representar algumas melodias, podendo aumentar a complexidade das criações na medida em que desenvolve habilidades do pensamento computacional.
A3	ACCembly, robô Dash	Os autores argumentam que o pensamento computacional está se tornando uma habilidade fundamental, como a leitura e a escrita, e apresentam ferramentas já elaboradas e amplamente utilizadas para o seu desenvolvimento. Nesse cenário, buscam identificar se o aprendizado do pensamento computacional pode ser promovido através do uso de um recurso específico, o ACCembly. Conclui-se que a utilização da ferramenta provocou o uso de diferentes conceitos e competências do pensamento computacional, como decomposição, coleta de dados, reconhecimento de padrões e depuração.

Trabalho	Recursos utilizados	Como o pensamento computacional é abordado
A4	TACTOPI	A autora defende que a introdução do pensamento computacional na Educação Infantil tem se tornado benéfica para a aquisição de bases valiosas para uma posterior aprendizagem de uma linguagem de programação. Além disso, coloca que crianças podem programar e usar uma série de processos do pensamento computacional, como abstração, construção de algoritmos, depuração, decomposição e reconhecimento de padrões através de uma abordagem lúdica. Considerando tal fato, apresenta o TACTOPI, um sistema que consiste em um ambiente tangível que proporciona treinamento em habilidades de orientação, enriquece experiências sensoriais usando elementos sonoros, visuais e táteis, e permite a introdução aos conceitos do pensamento computacional através de atividades lúdicas envolvendo contação de histórias.
A5	Torino	Os autores argumentam que a aprendizagem de computação pode ser transformadora para pessoas com deficiência visual, podendo inclusive ter importantes desdobramentos em termos profissionais. No entanto, um requisito fundamental para isso é a introdução do pensamento computacional ainda na infância; para os autores, sem tal intervenção inicial, crianças com deficiência visual podem ser privadas de uma série de oportunidades que requerem habilidades relacionadas ao pensamento computacional. No estudo, os autores buscam analisar se é possível que crianças com deficiência visual aprendam conceitos do pensamento computacional a partir de uma abordagem que utiliza o Torino como um brinquedo para exploração criativa de música ao invés de uma ferramenta estruturada de ensino. Os autores colocam que não foram utilizados testes padronizados para avaliar a aprendizagem dos conceitos do pensamento computacional. Em lugar disso, os facilitadores sugeriam atividades para as crianças e as envolviam em brincadeiras casuais relacionadas ao que estava ocorrendo. A partir das interações, avaliava-se as respostas e reações das crianças, o que sugeriria se houve compreensão dos conceitos do pensamento computacional.
A6	Torino	Os autores relatam que a partir dos programas criados pelos participantes, pôde-se inferir que foram compreendidos diferentes conceitos do pensamento computacional, como fluxo de controle, variáveis, <i>loops</i> , <i>bugs</i> , embora não no mesmo vocabulário. No entanto, os autores não colocam como tais conceitos são desenvolvidos.
A7	KIBO	Embora o título do trabalho mencione a facilitação da aprendizagem colaborativa e do pensamento computacional para crianças com deficiência visual, em nenhum momento é sequer mencionado o pensamento computacional ao longo do texto.
A8	Robô Ozobot	Os autores apresentam exemplos na literatura especializada que endossam a utilização de robôs para auxiliar crianças a aprenderem conceitos do pensamento computacional em contextos em que há pessoas com capacidades visuais mistas. No entanto, não relatam como a proposta apresentada pode ser utilizada no desenvolvimento do pensamento computacional.
A9	Torino	Os autores relatam que o Torino é uma linguagem de programação física para o ensino de programação e do pensamento computacional para crianças com idade entre 7 e 11 anos, incluindo aquelas com deficiência visual. No entanto, não mencionam como o pensamento computacional pode ser trabalhado a partir do uso do recurso.

Trabalho	Recursos utilizados	Como o pensamento computacional é abordado
A10	Robô Dash, Robô Super DOC, OsmoCoding Awbie, PUZZLETS	<p>Os autores colocam que o treinamento do pensamento computacional desenvolve as habilidades de criança e as prepara para a vida, possibilitando crescimento pessoal e preparação para uma carreira.</p> <p>Ademais, mencionam que o desenvolvimento do pensamento computacional em crianças vem sendo gradativamente mais explorado, especialmente através de ambientes virtuais baseados em blocos, que por sua marcada visualidade, não são acessíveis a pessoas com deficiência visual. Nesse cenário, os autores selecionam um conjunto de ambientes que possuem ao menos um componente tangível, focados em atividades de natureza especial relacionadas com o pensamento computacional. Posteriormente, apresentam para professores e instrutores de crianças com deficiência visual as principais características dos recursos utilizados, de modo que eles possam compartilhar suas impressões em relação aos ambientes, e realizam um workshop com crianças com deficiência visual utilizando um ambiente com um robô.</p>
A11	Tangible music programming blocks	<p>O autor argumenta que ao oferecer uma interface que pode ser operada por várias mãos simultaneamente, os blocos tangíveis apresentados no trabalho facilitam a interação e promovem a colaboração, possibilitando a criação de diversas ferramentas educacionais de programação para ensinar o pensamento computacional de forma colaborativa. Por outro lado, o autor também menciona que, para possibilitar o desenvolvimento completo do pensamento computacional, é necessário introduzir blocos adicionais em trabalhos futuros que permitam usar a abstração, decomposição, reconhecimento de padrões e construção de algoritmos. Mais explicitamente, menciona-se a criação de dois blocos específicos: um para criar funções e um para construir estruturas paralelas.</p>
A12	iCETA	<p>Os autores relatam que o iCETA fornece um ambiente lúdico multissensorial para crianças com diferentes habilidades aprenderem a matemática. Mencionam, ainda, que entre os objetivos futuros do projeto estão a expansão da variedade de conceitos matemáticos abordados no recurso e a utilização de aparatos eletrônicos embutidos que forneçam devolutivas mais variadas (sons, vibração e força), aplicando a abordagem também para outros domínios de aprendizagem, como o pensamento computacional. No entanto, não é explicitado como isso ocorrerá.</p>
A13	CodeRhythm	<p>Os autores colocam que o kit foi desenhado para o aprendizado do pensamento computacional. Mencionam ainda o desejo de, no futuro, projetar um currículo detalhado e bem estruturado para ensinar sistematicamente o pensamento computacional a estudantes com deficiência visual. No entanto, não relatam como o kit pode ser utilizado para tal, nem apresentam resultados especificamente neste sentido.</p>
A14	Robô Ozobot	<p>Os autores mencionam que após serem oferecidos dois <i>workshops</i> para aprendizagem das funcionalidades básicas e programação dos robôs utilizados na pesquisa, o terceiro <i>workshop</i> foi destinado à introdução aos algoritmos e ao pensamento computacional. Após a condução do estudo, concluíram que entre as diferentes camadas de aprendizado geradas no trabalho, está a de utilização do pensamento computacional para engajamento crítico com aspectos técnicos do jogo.</p>

Trabalho	Recursos utilizados	Como o pensamento computacional é abordado
A15	iCETA	Os autores apresentam pesquisas que vêm sendo desenvolvidas com o foco na criação de sistemas que permitam o aprendizado de habilidades do pensamento computacional. Além disso, colocam que estão atualmente explorando diferentes ambientes com interface tangível para que crianças possam aprender o pensamento computacional e se engajar em atividades de natureza espacial envolvendo robôs. No entanto, não explicitam como isso está sendo feito.
A16	LEGO Mindstorms	Os autores colocam que a participação em atividades de robótica pode promover o pensamento crítico, o pensamento computacional, a capacidade de resolução de problemas e a compreensão de sistemas complexos. No entanto, não se menciona como as adaptações propostas no estudo podem promover o desenvolvimento do pensamento computacional.
A17	Robô BlueBot, robô BeeBot e robô Ozobot	Os autores colocam que através do uso do robô Ozobot é possível desenvolver a criatividade, a lógica e o pensamento computacional de crianças, já que o recurso é atrativo e permite o ensino de habilidades básicas de programação e robótica. No trabalho, são apresentados exemplos de atividades que objetivam o desenvolvimento da concentração, atenção, percepção visual, memória, raciocínio lógico, orientação espacial, pensamento algorítmico e pensamento computacional através do uso dos robôs.
A18	Bonk	Os autores colocam que ambientes introdutórios de programação devem enfatizar as habilidades do pensamento computacional, como sequências, condições, eventos e <i>loops</i> . No entanto, não explicitam como tais habilidades podem ser exploradas através do uso do Bonk.
A19	LEGO, robô Ozobot	Os autores colocam que recentemente o pensamento computacional tem se tornado uma disciplina nas escolas, e que as habilidades relacionadas vêm sendo desenvolvidas especialmente através do uso de ambientes de programação, com destaque para aqueles baseados em blocos. Nesse sentido, os autores desenvolvem e implementam um ambiente multimodal com blocos LEGO e recursos de voz para a aprendizagem de programação e desenvolvimento do pensamento computacional em crianças com deficiência visual. No entanto, os autores não mencionam explicitamente como as habilidades do pensamento computacional podem ser desenvolvidas.
A20	StoryBlocks	Os autores argumentam que, mesmo para estudantes que não seguem carreiras relacionadas à computação, aprender sobre esta ciência pode promover o pensamento computacional que, por sua vez, pode ser aplicado a diferentes domínios. Nesse sentido, desenvolvem o StoryBlocks, um jogo tangível baseado em blocos que permite que pessoas cegas aprendam conceitos básicos de programação através da criação de histórias em áudio. Os autores identificam diversos aspectos do StoryBlocks que promovem elementos do pensamento computacional, destacando, entre eles, a possibilidade de desenvolver atividades que exploram conceitos como sequências, loops e condicionais, além de práticas como iteração e avaliação. Por outro lado, colocam que há pouco ou nenhum espaço para a exploração de eventos, paralelismos, operadores, abstração e modularização. No estudo de caso, os autores concluem que, embora os grupos tenham utilizado processos diferentes para a criação de histórias, todos engajaram em práticas relacionadas ao pensamento computacional. Por fim, em versões futuras do StoryBlocks, pretendem aumentar a capacidade do ambiente de desenvolver o trabalho criativo e promover a resolução de problemas e o pensamento computacional.

Fonte: elaborado pelo autor (2024).

O Quadro 8 mostra que em diversos trabalhos, por mais que se coloque que as práticas propostas promovem o desenvolvimento do pensamento computacional, não se esclarece como ou por que se chega a essa conclusão. Em alguns casos, o pensamento computacional é apenas mencionado no título ou na fundamentação teórica do estudo, mas sequer é abordado explicitamente ao longo do texto. Por mais que os estudos apresentem propostas voltadas para a resolução de problemas utilizando conceitos e habilidades relacionadas ao pensamento computacional, poucos são os que exploram de maneira mais detalhada o desenvolvimento do PC através das atividades conduzidas.

No que se refere a quais habilidades do pensamento computacional são desenvolvidas através das propostas, os autores, em geral, mencionam o desenvolvimento de conceitos como sequências, *loops*, condicionais e paralelismo, e também de habilidades como abstração, decomposição, reconhecimento de padrões, algoritmização e depuração. O Quadro 9 sintetiza os resultados obtidos.

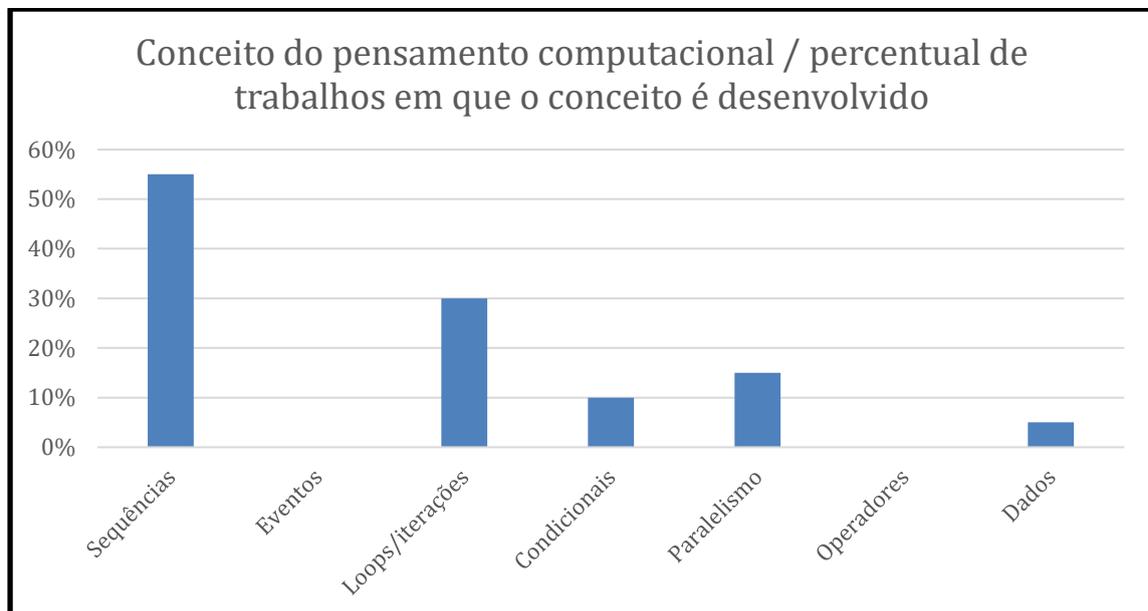
Quadro 9: Habilidades/conceitos do pensamento computacional desenvolvidos nos trabalhos analisados.

Habilidade/ conceito	Trabalhos em que se menciona o desenvolvimento da habilidade/ do conceito	Quantidade de trabalhos
Sequências	A1, A2, A3, A4, A5, A6, A9, A10, A15, A19, A20	11
Loops/iterações	A2, A3, A5, A6, A10, A20	6
Condicionais	A5, A20	2
Paralelismo	A5, A6, A9	3
Coleta de dados	A3	1
Variáveis	A6, A9	2
Abstração	A9	1
Decomposição	A3, A9	2
Reconhecimento de padrões	A3	1
Algoritmização	A2, A3, A4, A11, A14, A17	6
Depuração	A2, A3, A4, A5, A6, A9, A10, A15, A19, A20	10

Fonte: elaborado pelo autor (2024).

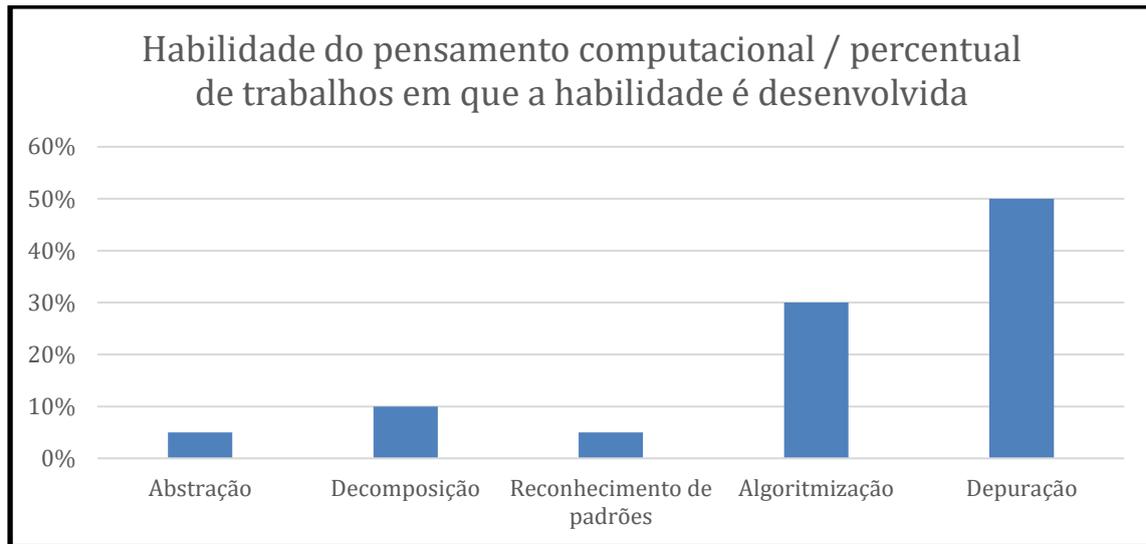
Dos sete conceitos mencionados por Brennan e Resnick (2012) como característicos do pensamento computacional, cinco são mencionados nos trabalhos analisados. Já no que se refere aos conceitos (habilidades) colocados por Csizmadia et al. (2015) e Brackmann (2017) (pilares do pensamento computacional), todos são mencionados. No entanto, poucos são os trabalhos que abordam uma grande variedade de conceitos. A Figura 21 e a Figura 22 ilustram os percentuais de trabalhos nos quais cada conceito é desenvolvido, de acordo com as formulações de Brennan e Resnick (2012) e de Csizmadia et al. (2015), respectivamente.

Figura 21: Percentual de trabalhos analisados cujas propostas promovem o desenvolvimento de cada conceito do pensamento computacional (formulação de Brennan e Resnick, 2012).



Fonte: elaborado pelo autor (2024).

Figura 22: Percentual de trabalhos analisados cujas propostas promovem o desenvolvimento de cada habilidade do pensamento computacional (formulação de Czismadia et al., 2015).



Fonte: elaborado pelo autor (2024).

As figuras evidenciam que a maior parte das habilidades e conceitos do pensamento computacional são desenvolvidos apenas em um pequeno percentual dos trabalhos analisados. Em relação aos conceitos do PC na formulação de Brennan e Resnick (2012), a noção de *sequências* é a que mais aparece nos estudos. Uma possível explicação para isso recai na simplicidade do conceito em relação aos demais, na medida em que para o desenvolvimento das ideias de *loop*, condicional e eventos, entre outras, seria necessária a construção de estruturas mais complexas, por vezes incompatíveis com a proposta do trabalho. Pires et al. (2022), por exemplo, colocam que os resultados obtidos indicam que a ferramenta desenvolvida (*iCETA*) pode ser eficiente em uma fase inicial da aprendizagem de programação, onde conceitos simples são introduzidos. Por outro lado, o recurso apresentaria limitações na construção de programas mais complexos.

Já considerando as habilidades do PC definidas por Czismadia et al. (2015), a mais abordada nos trabalhos é a de depuração. Em diversos estudos, os autores mencionam a realização de testes pelos participantes com o objetivo de verificar se o programa construído funcionava de acordo com o esperado. Havendo algum problema ou divergência em relação às expectativas, os participantes deveriam identificar e corrigir possíveis erros no algoritmo construído. Por sua vez, a habilidade de algoritmização foi a segunda mais abordada nos trabalhos. Rocha et al. (2021) mencionam, por exemplo, a identificação, ordenação e montagem de blocos com o objetivo de construir algoritmos

para a solução de problemas. Na mesma perspectiva, outros autores também abordam o desenvolvimento da habilidade de construção de algoritmos.

Por fim, por mais que se conclua que dada proposta promove o desenvolvimento de algum conceito ou habilidade do pensamento computacional, em muitos trabalhos não se expõe de maneira clara como se chega a essa conclusão. Não havendo clareza nos critérios e mecanismos de avaliação, a análise e interpretação dos resultados de experimentos pode se tornar excessivamente subjetiva, o que pode, inclusive, prejudicar sua reprodutibilidade.

5. CAMINHOS METODOLÓGICOS

Segundo Gil (2008), uma pesquisa científica deve ser desenvolvida a partir dos conhecimentos disponíveis acerca do fenômeno estudado, através da utilização cuidadosa de métodos, técnicas e outros procedimentos científicos. Tal processo envolve diversas etapas, que vão desde a formulação adequada do problema de pesquisa até a apresentação dos resultados.

Destacando que o conceito de ciência está ligado ao de método científico, Richardson (1999, p.22) define método como “o caminho ou a maneira para chegar a determinado fim ou objetivo”. Na mesma linha, Marconi e Lakatos (2003, p. 83) argumentam que todas as ciências se caracterizam pela utilização de métodos científicos. Para as autoras, o método pode ser definido como um conjunto de atividades “sistemáticas e racionais que, com maior segurança e economia, permite alcançar o objetivo - conhecimentos válidos e verdadeiros -, traçando o caminho a ser seguido, detectando erros e auxiliando as decisões do cientista”.

Nessa perspectiva, apresentamos nesta seção os caminhos que foram seguidos no sentido de alcançar nossos objetivos de pesquisa.

5.1. TIPO DE PESQUISA E ABORDAGEM

A pesquisa realizada é exploratória, tipo de estudo que objetiva tornar um problema mais explícito, constituindo hipóteses e aprimorando ideias ou intuições. Como coloca Gil (2008), a pesquisa exploratória tem planejamento flexível, possibilitando a consideração dos mais variados aspectos relativos ao fato estudado e proporcionando maior familiaridade com o problema. Ainda de acordo com o autor, na maior parte dos casos esse tipo de pesquisa envolve levantamento bibliográfico, entrevistas e análises de exemplos que estimulem a compreensão do fenômeno estudado. No contexto deste trabalho, os procedimentos empregados foram pesquisa documental, pesquisa bibliográfica e pesquisa de campo.

Além disso, a pesquisa tem abordagem qualitativa, na medida em que se baseia na interpretação dos fenômenos observados e do significado que carregam, considerando a realidade em que tais fenômenos estão inseridos e a particularidade de cada sujeito objeto da pesquisa (Nascimento, 2016). Nesse caso, a categorização dos

dados envolve textos narrativos, matrizes e esquemas, de modo que o conjunto inicial de categorias pode ser reexaminado e modificado sucessivamente, objetivando alcançar ideias mais abrangentes e significativas. A análise qualitativa depende de fatores como a natureza dos dados coletados, o tamanho da amostra, os instrumentos utilizados para coleta e os pressupostos teóricos que orientaram a investigação. De modo geral, a execução de tal análise envolve a redução dos dados coletados, sua categorização, sua interpretação e, por fim, a redação de um relatório final (Gil, 2008).

5.2. DESENHO DA PESQUISA

O desenvolvimento do trabalho foi dividido em três etapas, descritas a seguir. Na primeira etapa buscou-se maior aprofundamento teórico em relação ao objeto de pesquisa, momento em que foram realizadas as pesquisas documental e bibliográfica. Nesse sentido, foi realizado um levantamento de referências sobre deficiência visual, Educação Especial, Educação Matemática, pensamento computacional e acessibilidade em ambientes de programação, recorrendo-se a publicações científicas e documentos oficiais para tal. Esta etapa foi complementada posteriormente pela realização de uma revisão sistemática de literatura acerca do desenvolvimento do pensamento computacional em estudantes com deficiência visual, objetivando caracterizar o estado da arte acerca da temática, identificar publicações correlatas recentes e validar o ineditismo da pesquisa. Os resultados obtidos nesta fase foram utilizados para construir a fundamentação teórica da tese e são apresentados nas seções 3 e 4 deste trabalho.

A segunda etapa do trabalho consistiu na realização de pesquisa de campo com o objetivo de analisar as potencialidades do uso de uma linguagem algorítmica baseada em português estruturado como recurso para o desenvolvimento das habilidades do pensamento computacional em estudantes com deficiência visual. A pesquisa de campo compreendeu a concepção e a aplicação de uma sequência didática junto a um grupo de estudantes com deficiência visual, todos do Ensino Médio de diferentes campi do Colégio Pedro II, no estado do Rio de Janeiro. Para a construção, aplicação e avaliação dos resultados da sequência didática, optou-se pela combinação de duas metodologias complementares: a Engenharia Didática (Artigue, 1988) e a análise de conteúdo (Bardin, 1977). A primeira delas, metodologia de pesquisa desenvolvida a partir dos avanços da Didática Matemática francesa na década de 1980, justifica-se pelo fato de se tratar de uma abordagem voltada para a construção, análise e validação de sequências didáticas.

Já a segunda foi utilizada na análise das interações vivenciadas entre pesquisador e participantes durante a aplicação das sequências didáticas.

A última etapa consistiu na análise das informações colhidas no estudo de campo. Nessa fase, utilizando os referenciais teóricos da Engenharia Didática e da análise de conteúdo, os dados obtidos na pesquisa de campo foram simplificados e transformados de acordo com os objetivos da pesquisa, organizados para identificação de semelhanças e divergências, e revisados para extração de significados, padrões e explicações. Com isso, objetivou-se identificar vantagens e desvantagens da utilização da programação de computadores para resolução de problemas matemáticos junto a estudantes com deficiência visual a partir de regularidades observadas nos dados coletados. Por fim, com a análise das entrevistas realizadas junto aos estudantes, objetivou-se obter informações e explicações adicionais advindas dos próprios participantes, tanto em relação às atividades conduzidas, quanto no que se refere à linguagem algorítmica utilizada e ao próprio ato de programar.

5.3. DESCRIÇÃO DOS PARTICIPANTES DA PESQUISA

Todos os participantes da pesquisa eram estudantes do Ensino Médio do Colégio Pedro II, instituição pública que integra a rede federal de Educação Profissional, Científica e Tecnológica do país. O complexo escolar da instituição é composto por 14 campi e um Centro de Referência em Educação Infantil, localizados em seis bairros da cidade do Rio de Janeiro e também nos municípios de Duque de Caxias e Niterói, abrangendo turmas desde a Educação Infantil até o Ensino Médio Regular e Integrado, além da Educação de Jovens e Adultos (Proeja).

No momento de início do estudo de campo, de acordo com a Coordenação Geral dos Núcleos de Atendimento a Pessoas com Necessidades Específicas (Napnes) da instituição, o Colégio Pedro II possuía aproximadamente vinte estudantes com deficiência visual matriculados no Ensino Médio em diferentes campi. Após contato com diversos Napnes da instituição com o objetivo de avaliar a viabilidade da condução da pesquisa em cada campus, optou-se por limitar o estudo de campo a três campi em virtude das especificidades de cada núcleo e das disponibilidades do pesquisador e dos possíveis participantes.

Após apresentação da pesquisa aos estudantes com deficiência visual dos três campi mencionados, a pesquisa teve início com a participação de sete estudantes, dos quais três se mantiveram até o último encontro. No Quadro 10 é apresentada a identificação utilizada para cada participante, bem como suas idades, séries no momento de início do estudo de campo e características da deficiência visual (baixa visão ou cegueira). Todos os participantes iniciaram os estudos no Colégio Pedro II no Ensino Médio, tendo realizado as etapas de escolarização anteriores em outras instituições de ensino. Nos anos escolares cursados no Colégio Pedro II, nenhum dos participantes apresentou histórico de reprovações em Matemática. Ressalta-se, aqui, que para preservar a identidade dos estudantes foram utilizados nomes fictícios para todos, além de letras para designação dos campi onde os mesmos estudavam.

Para a condução da sequência didática, os participantes foram divididos em grupos de acordo com os turnos em que estudavam e a disponibilidade de participação nos encontros. Foram formados quatro grupos, com os quais seriam conduzidos encontros semanais em horários previamente estipulados. A divisão de grupos também é exibida no Quadro 10.

Quadro 10: Participantes da pesquisa.

Identificação do(a) participante	Característica do laudo	Campus	Idade (em anos)	Série	Grupo
Adriana	Baixa visão	A	17	3º ano	1
Marcos	Cegueira congênita	A	16	1º ano	2
Renata	Cegueira congênita	B	19	2º ano	3
Sara	Baixa visão	B	17	2º ano	3
Ana	Baixa visão	C	18	1º ano	4
Fábio	Baixa visão	C	22	3º ano	4
Helena	Baixa visão	C	17	1º ano	4

Fonte: elaborado pelo autor (2024).

Os encontros foram conduzidos nos Napnes dos campi, tendo a duração de aproximadamente 80 minutos cada. No caso de alguns grupos, no entanto, foi necessário realizar os encontros finais de maneira remota. A caracterização e o detalhamento dos encontros, bem como a descrição dos casos extraordinários, são apresentados na seção referente aos resultados deste trabalho.

5.4. LEVANTAMENTO DE DADOS

De acordo com Marconi e Lakatos (2003), esta é a etapa da pesquisa em que se inicia a aplicação dos instrumentos elaborados e das técnicas selecionadas, com o objetivo de se efetuar a coleta dos dados previstos inicialmente. As autoras colocam, ainda, que o rigoroso controle na aplicação dos instrumentos de pesquisa é fator fundamental para evitar erros ou tendenciosidade nos dados. Neste trabalho, foram utilizados os seguintes procedimentos para levantamento de dados: observação direta, gravação de áudios dos encontros realizados e gravação de vídeos das telas dos computadores utilizados pelos participantes. No encontro final de cada grupo também foi realizada entrevista semiestruturada com os participantes.

Em relação aos procedimentos de coleta utilizados, no início de cada encontro os participantes eram informados de que seria realizada a gravação do áudio de toda a sessão. Ao final do encontro, o arquivo era nomeado através de um código e armazenado em uma plataforma de armazenamento pessoal do pesquisador, para posterior realização de análise de conteúdo do material. Além disso, em momentos da sequência didática que envolviam a construção de programas, os participantes eram informados de que seria realizada a gravação da tela do computador que estivessem utilizando, de maneira que pudesse ser registrado o processo de construção do código. De modo análogo ao ocorrido com os áudios, tais gravações eram identificadas com um código e armazenadas em plataforma de armazenamento pessoal do pesquisador. Por fim, ao final da sequência didática foi realizada entrevista semiestruturada com os participantes, com o objetivo de registrar suas impressões, dificuldades e sugestões, tanto em relação à sequência, quanto em relação à utilização da linguagem algorítmica empregada.

5.5. CONSIDERAÇÕES ÉTICAS

A pesquisa tem anuência da Pró-Reitoria de Pós-Graduação, Pesquisa, Extensão e Cultura do Colégio Pedro II, tendo sido aprovada pelos Comitês de Ética da Universidade Federal Fluminense e do Colégio Pedro II. Os riscos envolvidos na pesquisa foram apresentados no Termo de Consentimento Livre e Esclarecido (TCLE), bem como no Termo de Assentimento Livre e Esclarecido (TALE), documentos que se encontram nos Apêndices deste trabalho. Com o objetivo de minimizar os riscos

mencionados, o pesquisador comunicou inicialmente aos participantes que a pesquisa não tinha qualquer relação com as atividades acadêmicas escolares, não fazendo parte da composição de notas de nenhuma disciplina. Também foi comunicado aos participantes que não havia qualquer problema se houvesse dificuldade na realização das atividades propostas, sendo tal fato natural no processo de aprendizagem. Por fim, ressalta-se que foram utilizados códigos para identificação das gravações de vídeos e áudios, com o objetivo de preservar a identidade dos participantes. Tais gravações foram ouvidas unicamente pelo pesquisador responsável.

5.6. REFERENCIAL METODOLÓGICO

5.6.1. ENGENHARIA DIDÁTICA

A noção de Engenharia Didática surgiu no início da década de 1980, em um contexto de desenvolvimento formal de pesquisas que contribuíram para a constituição do campo teórico da Didática da Matemática francesa. Nesse cenário de produção científica, e tendo como fruto uma série de investigações relacionando o processo de ensino e aprendizagem da Matemática à condução de sequências didáticas em sala de aula, o referencial teórico da Engenharia Didática é desenvolvido especialmente a partir da contribuição teórica da pesquisadora francesa Michele Artigue. De acordo com Artigue (1988), o termo foi cunhado com o objetivo de comparar o fazer didático ao trabalho de um engenheiro que, para realizar um determinado projeto, se apoia no conhecimento científico de sua área, concorda em submeter o trabalho a um controle de tipo científico, mas, ao mesmo tempo, é obrigado a lidar com uma série de situações de grande complexidade, que extrapolam o puro fazer científico. Dessa forma, seria necessário atacar, de forma prática e com os meios que se encontram à disposição, problemas dos quais a Ciência pode ainda não dar conta.

Segundo Bittar (2017), no início da década de 1980 ainda não existiam metodologias bem definidas que auxiliassem pesquisadores no preparo e análise de sequências didáticas, uma vez que as metodologias oriundas do campo da Educação não atendiam às especificidades que emergiam dos trabalhos em Educação Matemática. Assim, tanto em uma perspectiva metodológica quanto científica, era necessário estabelecer relações mais concretas entre a especificidade do conteúdo matemático e a

natureza das questões didáticas. Nesse sentido, a Engenharia Didática emerge objetivando explicitar as relações entre pesquisa e ação no sistema educacional, em especial no que se refere ao ensino e aprendizagem da Matemática, compreendendo o lugar reservado para as realizações didáticas no contexto das metodologias de pesquisa (Guedes et al., 2019; Carneiro, 2005; Artigue, 1988).

Destacando a importância da articulação entre pesquisa e ação, Artigue (1988) cita Chevalard (1982) ao fazer uma crítica a uma “ideologia da inovação”, que teria ganho destaque somente por conta da inexistência de uma cultura científica no campo educacional à época. A autora questiona, assim, uma obsessão relacionada à elaboração de práticas pedagógicas ditas inovadoras que, no entanto, careciam de bases científicas para sua criação e execução. Para ela, a Engenharia Didática surge em contraposição, como uma possibilidade de ação racional com base em conhecimentos didáticos pré-estabelecidos, destacando, ainda, a importância da produção didática como prática de pesquisa.

Enquanto metodologia de pesquisa, a Engenharia Didática se caracteriza em primeiro lugar por um esquema experimental baseado em produções didáticas conduzidas em sala de aula, tendo por base a concepção, realização, observação e análise de sequências de ensino. Além disso, a metodologia se difere de outros tipos de pesquisa baseados em experimentos em sala de aula em virtude do modo de validação utilizado para as sequências didáticas elaboradas. Em geral, esses outros tipos de pesquisa adotam uma abordagem com validação externa baseada na comparação de performance de grupos experimentais com grupos controle. Esse não é o caso da Engenharia Didática, que tem a especificidade de tal validação ser realizada internamente, através da confrontação de uma análise a priori com uma a posteriori, não havendo a necessidade de aplicação de um pré-teste ou de um pós-teste (Almouloud, Coutinho, 2008; Artigue, 1988).

A abordagem metodológica da Engenharia Didática foi aprimorada ao longo das décadas seguintes, tendo sido usada na pesquisa em diferentes níveis educacionais, e também em pesquisas relacionadas à formação de professores. Nesse sentido, Almouloud e Silva (2012) demarcam a existência de duas gerações da Engenharia Didática. A primeira, que emerge da Didática da Matemática na década de 1980, teria sido apresentada como uma metodologia de pesquisa que possibilitaria o surgimento e análise de fenômenos didáticos nas condições mais próximas possíveis do

funcionamento de uma sala de aula clássica. Já a Engenharia de segunda geração teria por primeiro objetivo o desenvolvimento de recursos para o ensino regular ou para a formação de professores.

Por fim, Artigue (1988) destaca que a metodologia da Engenharia Didática se divide em quatro fases: as análises preliminares, a fase de concepção, a fase de experimentação e a fase de análise a posteriori e avaliação. A seguir, são apresentados aspectos de cada uma das etapas mencionadas.

Análises preliminares

De acordo com Artigue (1988), a fase de concepção da sequência didática deve se apoiar em um conjunto de análises preliminares acerca do contexto educacional em que a pesquisa se insere. Como colocam Zborowski e Pigatto (2018), é indicado que o pesquisador faça esta análise a partir de três dimensões: a epistemológica, que está associada às características do saber em questão; a didática, que se refere ao modo como se estrutura o ensino do tema na atualidade; e a cognitiva, que diz respeito à análise de concepções comuns dos estudantes sobre o tema a ser trabalhado, bem como suas dificuldades e obstáculos.

Na dimensão epistemológica das análises preliminares, o conteúdo que se quer ensinar é caracterizado de forma ampla, de modo que os conceitos envolvidos são abordados considerando suas diferentes possibilidades de representação, e a diversidade de problemas em que eles são utilizados como ferramenta é considerada de maneira abrangente. A dimensão didática compreende uma análise do ensino atual e de seus efeitos, caracterizando a forma como o conteúdo é usualmente desenvolvido, sua evolução com as mudanças curriculares e as consequências dessa abordagem na atuação do professor, em práticas avaliativas, e outros aspectos do ensino. Para tal, propõe-se um estudo dos documentos orientadores da educação nacional e uma análise dos livros didáticos que tratam sobre esse assunto. Por sua vez, a dimensão cognitiva preconiza um levantamento de condutas de estudantes e a identificação de problemas por eles enfrentados, erros cometidos e outros comportamentos observados.

Por fim, os elementos reunidos a partir das análises conduzidas permitirão a formulação de hipóteses de diferentes naturezas (epistemológica, didática, cognitiva, afetiva, interacional) em relação ao saber em questão. Munido de tais informações o

pesquisador pode, devidamente embasado, dar seguimento ao processo de Engenharia Didática, partindo para a fase de criação da sequência em si.

Concepção e análise a priori

A segunda fase da Engenharia Didática é caracterizada pela concepção da sequência e realização da análise a priori. Nesta etapa a sequência didática que se pretende desenvolver é de fato elaborada, considerando as informações reunidas a partir das análises preliminares, os conhecimentos que os estudantes já possuem e as competências e habilidades que precisam desenvolver. Desse modo, trata-se do momento em que as situações e objetos de aprendizagem são construídos, a quantidade de encontros necessários para a condução da sequência didática, bem como a duração de cada um deles, é definida, e os passos que se pretende seguir na condução da sequência didática são descritos (Zborowski; Pigatto, 2018).

Nesta etapa também é realizada a análise a priori da sequência didática, processo inseparável da concepção que, como esta última, se baseia na teoria das situações didáticas de Brousseau (Artigue, Perrin-Glorian, 1991). Para Artigue (1988), se a teoria construtivista destaca a necessidade de envolvimento do estudante na construção de seu conhecimento através da interação com o ambiente, a teoria das situações didáticas, referência teórica para a metodologia da Engenharia Didática, teve desde o princípio a ambição de se constituir como uma teoria do controle da relação entre situação e significado. Nesse contexto, para a autora, a análise a priori deve ser concebida, de modo análogo, como um mecanismo de controle de significados indissociável do processo de concepção da sequência didática.

De maneira geral, Artigue (1988) aponta que o objetivo da análise a priori é determinar como as escolhas realizadas no momento de concepção da sequência didática permitem prever e controlar os comportamentos dos estudantes, bem como seus significados. Assim, esta análise comporta uma parte descritiva, na qual o pesquisador descreve as escolhas realizadas, e outra preditiva, na qual são realizadas hipóteses em relação aos possíveis comportamentos que o aluno terá frente à atividade proposta (Zborowski; Pigatto, 2018). A validação das hipóteses aqui construídas se dará a partir de uma confrontação entre as análises a priori e a posteriori, esta última realizada na quarta fase da Engenharia Didática.

Experimentação

A fase de experimentação é o momento de colocar em prática as ações elaboradas na fase de concepção. Desse modo, nessa etapa a sequência construída é aplicada junto aos estudantes e são coletados os dados que serão analisados na próxima etapa do processo de engenharia. Tais dados podem ser coletados a partir da observação direta das sessões de ensino, bem como através do registro dos comportamentos, interações e produções dos estudantes (Artigue, 1988).

Para Dolz (2016), esta etapa pode consistir em uma simples implementação da sequência desenvolvida através de um estudo de caso para ajustar as atividades e as inovações propostas à realidade em que a pesquisa se insere, bem como pode ser objeto de um estudo mais amplo, com um número maior de participantes tendo em vista uma possível generalização. Além disso, na medida em que as ações propostas são conduzidas, é possível realizar as correções que sejam necessárias com base nas análises locais do desenvolvimento experimental, retornando-se à análise a priori em um processo de constante complementação (Almouloud; Coutinho, 2008).

Análise a posteriori e validação

Na análise a posteriori os dados recolhidos durante a experimentação são retomados e analisados à luz da análise a priori. Desse modo, a partir de observações das sessões de ensino e das produções dos estudantes dentro ou fora de sala de aula, são obtidos resultados que possam contribuir para a melhoria dos conhecimentos didáticos existentes acerca das condições de ensino e aprendizagem dos saberes em jogo (Artigue, 1988; Almouloud; Coutinho 2008).

De maneira geral, na análise a posteriori se realiza uma confrontação entre os comportamentos previstos inicialmente e os de fato ocorridos no decorrer da sequência didática. Com isso, é possível validar ou não as hipóteses elaboradas no início da pesquisa. Desse modo, como colocam Almouloud e Coutinho (2008), a análise a posteriori depende das ferramentas técnicas ou teóricas utilizadas (vídeos, áudios, produções textuais, contrato didático), com as quais os dados da pesquisa foram coletados. Portanto, é necessário que tais ferramentas sejam adequadas, considerando a natureza da sequência didática elaborada na fase de concepção.

Finalmente, os dados mencionados são analisados profundamente pelo pesquisador e as informações obtidas são confrontadas com a análise a priori realizada. Nesse momento, o objetivo é relacionar as observações realizadas com os objetivos definidos a priori, validando as hipóteses construídas e avaliando a reprodutibilidade e a regularidade dos fenômenos didáticos identificados (Almouloud; Coutinho, 2008).

5.6.2. ANÁLISE DE CONTEÚDO

De acordo com Capelle, Melo e Gonçalves (2003, p. 2), os dados coletados, que inicialmente se encontram em estado bruto, devem ser transformados em resultados de pesquisa, o que envolve “a utilização de determinados procedimentos para sistematizar, categorizar e tornar possível sua análise por parte do pesquisador”. Tal processo demanda a utilização dos mecanismos apropriados para trazer à tona informações que ilustrem, expliquem ou ajudem a melhor compreender alguns fenômenos observados em dados obtidos por meio de entrevistas, mensagens e documentos em geral. Entre esses mecanismos encontra-se a análise de conteúdo.

Segundo Bardin (1977, p. 42), a análise de conteúdo é:

um conjunto de técnicas de análise das comunicações visando obter, por procedimentos sistemáticos e objetivos de descrição do conteúdo das mensagens, indicadores (quantitativos ou não) que permitam a inferência de conhecimentos relativos às condições de produção/recepção (variáveis inferidas) destas mensagens.

Assim, para além de uma simples descrição do conteúdo de uma mensagem, na análise de conteúdo busca-se identificar o que conduziu a um determinado enunciado, aspecto este que diz respeito às causas ou antecedentes da mensagem, e quais as consequências que determinado enunciado pode provocar, aspecto que se refere aos efeitos da mensagem. Desse modo, o conjunto de técnicas da análise de conteúdo é utilizado para identificar o que pode ser inferido a partir dos enunciados produzidos, quando os conteúdos de tais enunciados são tratados relativamente a outras coisas. Essas inferências, quando realizadas, podem ser de diferentes naturezas, como psicológica, sociológica, histórica e econômica, entre outras (Bardin, 1977).

Bardin (1977) argumenta que, na análise de conteúdo, objetiva-se explicitar, sistematizar e expressar o conteúdo de mensagens, com o contributo de índices passíveis ou não de quantificação. De acordo com a autora, a partir de um conjunto de

técnicas parciais, mas complementares, e utilizando um jogo de operações analíticas adaptadas à natureza do material analisado e da questão que se pretende resolver, o analista busca efetuar deduções lógicas e justificadas, referentes à origem das mensagens tomadas em consideração.

A análise de conteúdo é composta por três etapas: pré-análise, exploração do material, e tratamento dos resultados obtidos. Capelle, Melo e Gonçalves (2003) destacam que as etapas mencionadas se caracterizam pelos seguintes aspectos:

- pré-análise: organização e sistematização das ideias, escolha dos documentos analisados, retomada das hipóteses e objetivos iniciais da pesquisa em relação ao material coletado, e elaboração de indicadores que orientarão a interpretação final;
- exploração do material: codificação dos dados brutos do material, envolvendo procedimentos de recorte, contagem, classificação, desconto ou enumeração em função de regras previamente formuladas;
- tratamento dos resultados obtidos: submissão dos dados brutos a operações estatísticas, realização de inferências e interpretações de acordo com o quadro teórico e os objetivos propostos.

Por fim, Capelle, Melo e Gonçalves (2003) colocam que diferentes técnicas parciais podem ser utilizadas para promover a compreensão dos significados manifestos e latentes no material analisado (análise categorial, análise de avaliação, análise de enunciação, análise da expressão, análise das relações e análise do discurso). Mais especificamente, os autores destacam os seguintes aspectos:

- análise categorial: consiste em operações de desmembramento do texto em unidades (categorias), segundo reagrupamentos analógicos, visando a descobrir os núcleos de sentido que compõem uma comunicação e preocupando-se, ainda, com a frequência desses núcleos;
- análise de avaliação: visa a identificar atitudes do locutor quanto aos objetos de que ele fala, atendo-se basicamente à predisposição do emissor da mensagem para reagir sob a forma de opiniões (nível verbal), ou de atos (nível comportamental), em presença de objetos, de uma maneira determinada;
- análise de enunciação: se apoia na concepção da comunicação como um processo, desviando-se das estruturas e dos elementos formais presentes

no texto, considerando as condições de produção da palavra e as modalidades do discurso;

- análise da expressão: parte do princípio de que há uma correspondência entre o tipo de discurso e as características do locutor e de seu meio, sendo necessário conhecer o autor da fala, sua situação social e dados culturais que o moldam para se partir para a análise;
- análise das relações: busca extrair do texto as relações entre elementos da mensagem através da identificação de aparição associada de dois ou mais elementos do texto, atendo-se às relações que eles mantêm entre si e completando a análise frequencial simples;
- análise do discurso: tem como objetivo a inferência a partir dos efeitos de superfície de uma estrutura profunda, os processos de produção, procurando estabelecer relações entre a situação na qual o sujeito se encontra (condições de produção) e as manifestações semântico-sintáticas da superfície discursiva.

6. RESULTADOS E DISCUSSÃO

6.1. DESENVOLVIMENTO DA ENGENHARIA DIDÁTICA

6.1.1. ANÁLISES PRELIMINARES

Conforme mencionado anteriormente, a sequência didática deve ser concebida com base em um quadro teórico geral constituído a partir de conhecimentos didáticos previamente estabelecidos (Artigue, 1988). Para a elaboração de tal quadro, sugere-se a realização das análises preliminares, etapa inicial do processo de Engenharia Didática. Nesse momento, são obtidas informações importantes para a construção de uma intervenção adequada e bem fundamentada, considerando aspectos epistemológicos do campo de pesquisa, a dimensão didática do processo de ensino e aprendizagem, e aspectos cognitivos dos alunos. Os resultados obtidos nesta etapa preliminar são descritos a seguir.

A dimensão epistemológica

No contexto deste trabalho, a dimensão epistemológica se refere principalmente ao conceito de função na Matemática, que abrange múltiplas representações. Segundo Roque (2017), em geral, quando se pensa em função, duas coisas vêm à mente: a curva que a representa graficamente e sua expressão analítica. No entanto, a partir de um exercício mais formal, também seria possível relacionar o tema à ideia de correspondência, na figura de uma “máquina de entradas e saídas”. Ademais, é comum o uso de diagramas, esquemas e notação simbólica de conjuntos para a construção da ideia de função.

A existência de múltiplas representações para um mesmo conceito é comum na matemática, e tal diversidade é explorada a partir de diferentes construtos teóricos. De acordo com a Teoria dos Registros de Representação Semiótica, constituída a partir das contribuições do filósofo e psicólogo francês Raymond Duval (Duval, 1993, 2006, 2017), em virtude da natureza abstrata dos objetos matemáticos, é necessário recorrer a diferentes sistemas de expressão e representação para operar com os mesmos:

A particularidade da aprendizagem da Matemática faz com que as atividades cognitivas requeiram a utilização de sistemas de expressão e de representação

distintos da linguagem natural ou das imagens: variados sistemas de escrita para os números, notações simbólicas para os objetos, escritas algébricas e lógicas que tomam o status de linguagens paralelas à língua natural para expressar as relações e as operações, figuras geométricas, representações em perspectiva, gráficos cartesianos, redes, diagramas, esquemas, etc. (Duval, 2017, p.7, tradução nossa).

De maneira geral, Duval (2006) aponta que o estudo do papel das representações é um dos grandes quadros teóricos e metodológicos para a investigação e explicação da aquisição de conhecimentos matemáticos. Para o autor, as representações semióticas seriam produções constituídas pelo emprego de signos pertencentes a um sistema de representação que possui suas restrições próprias de significação e funcionamento (Duval, 1993). Nessa perspectiva, além de constituir um meio de exteriorização de representações mentais para fins de comunicação, as representações semióticas seriam essenciais para a própria atividade cognitiva do pensamento, na medida em que o próprio desenvolvimento de representações mentais depende da interiorização de representações semióticas.

Quanto ao papel das representações semióticas na atividade cognitiva e na aprendizagem, Duval põe em relevo dois pontos principais. Em primeiro lugar, coloca que além de preencher a função de objetivação, característica das representações mentais, as representações semióticas preenchem as funções de comunicação e de tratamento das informações. Em segundo lugar, o autor argumenta que não se pode conceber que um conteúdo representado esteja destacado da forma que o representa. Desse modo, não poderia haver formação e aquisição de conceitos sem a apreensão ou produção de uma representação semiótica e o recurso a uma pluralidade potencial de sistemas semióticos (Duval, 2017, p. 61).

A noção de imagem de conceito (Tall; Vinner, 1981; Vinner, 1991) também se relaciona com a relevância da multiplicidade de representações de um objeto matemático para sua aprendizagem conceitual. Tal noção é definida pelos autores como toda a estrutura cognitiva que está associada a um conceito, incluindo todas as imagens mentais, bem como propriedades e processos a elas associados. Além disso, tal imagem seria desenvolvida ao longo dos anos por meio de experiências de todos os tipos, se transformando na medida em que o indivíduo encontra novos estímulos e amadurece (Tall; Vinner, 1981). Assim, enquanto a definição de conceito seria uma formulação linguística tradicional que demarca os limites da palavra ou da frase que o representa, a imagem de um conceito compreenderia representações visuais, imagens mentais,

experiências e impressões evocadas pelo nome do conceito (Thompson, 1996). Nessa perspectiva, recorrer ao maior número possível de representações contribui para o enriquecimento da imagem que se constrói de um conceito.

O conceito de função se insere nesse contexto plural de abordagens, representações e significados. Do ponto de vista histórico, em seção em que busca elucidar as origens da noção de função, Roque (2018) argumenta que as tabelas babilônicas e egípcias já continham, de alguma forma, uma ideia do conceito, uma vez que se tratavam de registros de correspondências entre um número e o resultado de alguma operação que envolvia esse número. No entanto, a autora ressalta que um componente fundamental para a formalização do conceito de função ainda não havia surgido à época: a ideia de variação. Segundo ela, um passo fundamental para se chegar a esta noção foi o desenvolvimento da Física Matemática e da representação simbólica de uma quantidade desconhecida no século XVII.

De modo análogo, Ponte (1990) coloca que embora aspectos simples do conceito de função possam ser encontrados em épocas anteriores, como em operações elementares de contagem, o seu surgimento como conceito individualizado e objeto de estudo corrente na Matemática remonta ao final do século XVII. Segundo o autor, a noção de função se confunde com os primórdios do Cálculo Infinitesimal, na medida em que Newton (1642-1727) teria usado o termo “*relata quantitas*” para designar uma variável dependente, e o termo “*genita*” para se referir a uma quantidade obtida a partir de outras por meio de operações aritméticas fundamentais. Por sua vez, Leibniz (1646-1716) teria sido o primeiro a usar, em termos gerais, a palavra “função” (em 1673) para expressar a correspondência entre quantidades associadas a curvas da Geometria.

Ainda segundo Ponte (1990), com o desenvolvimento do estudo de curvas por meios algébricos, tornava-se necessário um termo para representar quantidades dependentes de uma variável através de uma expressão analítica. Nesse contexto, João Bernoulli (1667-1748) publica, em 1718, um artigo contendo sua definição de função de uma variável como uma quantidade que é composta de qualquer forma dessa variável e constantes. Para Bernoulli, cada função poderia ser representada por uma única expressão analítica, de modo que o conceito de função, em sua formulação, pode ser entendido como uma combinação de símbolos algébricos. Alguns anos mais tarde, em 1748, Euler (1707-1783), um antigo aluno de Bernoulli, após definir o significado de quantidade constante e quantidade variável, enuncia que uma função de uma

quantidade variável é uma expressão analítica composta de alguma maneira desta quantidade variável e números ou quantidades constantes (Botelho; Rezende, 2007).

Ponte (1990) observa que, a partir das contribuições de Bernoulli e Euler, a noção de função passava a ser identificada, na prática, com a de expressão analítica, fato que iria vigorar até o século XIX. Em 1822, no entanto, Fourier (1768-1830) publica a obra *Teoria Analítica do Calor*, que provoca uma revisão do conceito de função. O principal resultado matemático de Fourier nesta obra é que qualquer função poderia ser desenvolvida, em um intervalo, por uma série de senos e cossenos. Tal resultado não é completamente verdadeiro, e Dirichlet (1805-1859) formula, nos anos seguintes, as condições para que uma função possa ser representada através de uma série de Fourier (Botelho; Rezende, 2007; Ponte, 1990).

Botelho e Rezende (2007) destacam que a descoberta de Fourier contribuiu para esclarecer uma confusão entre duas noções aparentemente idênticas, mas diferentes: os conceitos de função e de sua representação analítica. Ponte (1990) coloca que Dirichlet, em 1837, separa essas duas noções, formulando o conceito de função em termos de uma relação arbitrária entre conjuntos numéricos. Assim, uma função passava a ser simplesmente uma correspondência entre duas variáveis, independente de relações algébricas, de modo que a todo valor da variável independente é associado um, e somente um valor da variável dependente.

Com o desenvolvimento da teoria dos conjuntos a partir da contribuição de Cantor (1845-1918), a noção de função passa a abranger correspondências arbitrárias entre quaisquer conjuntos, numéricos ou não. O próprio conceito de conjunto teve grande influência nas definições posteriores de funções. No entanto, a mais importante por conta de sua formalidade, rigor e generalidade, e que é usada atualmente nos meios matemáticos e científicos, foi dada em 1939 por um grupo de matemáticos que publicavam sob o pseudônimo de Nicolas Bourbaki⁷. De maneira geral, o grupo definiu função como uma tripla ordenada (X, Y, F) onde X e Y são conjuntos e F é um subconjunto de $X \times Y$, tal que para todo $x \in X$ existe $y \in Y$ tal que $(x, y) \in F$ e, se $(x, y) \in F$ e $(x, y') \in F$,

⁷ De acordo com Esquinalha (2012, p.28), Nicolas Bourbaki é o pseudônimo de “um grupo de matemáticos, quase todos franceses, que se reuniu para escrever um tratado de Análise e acabou por reorganizar boa parte da Matemática desenvolvida até então”.

então $y=y'$ (Silva, 2021; Barros; Silva; Silva, 2021; Botelho; Rezende, 2007; Silva; Rezende, 1999; Sierpiska, 1992).

Como visto, o conceito de função teve um longo processo de desenvolvimento histórico que culminou com as definições de Dirichlet e Bourbaki. No entanto, ao entrarem em contato com definições desse tipo, estudantes do Ensino Médio muitas vezes apresentam dificuldades na sua compreensão. Como colocam Brito e Almeida (2005), para o desenvolvimento de um conceito amplo como o de função, a utilização de uma linguagem matemática truncada e com objetivos em si mesma traz poucos benefícios em termos de construção de significados.

Lima (2003, p.44), nesse sentido, diz que “a conceituação, quando levada a extremos, pode colidir com os bons preceitos do ensino”. O autor critica a definição de função como um conjunto de pares ordenados sujeitos a determinadas condições (definição nos moldes de Dirichlet e Bourbaki), atribuindo tal formulação a uma “fixação conjuntista” advinda do movimento da Matemática Moderna”. Conceituações desse tipo objetivariam reduzir os conceitos fundamentais da Matemática à noção única de conjunto, evitando noções vagas ou intuitivas. Para o autor, embora logicamente corretas, definições desse tipo são excessivamente elaboradas e incompatíveis com a prática, não correspondendo à ideia que os matemáticos e os que utilizam a Matemática fazem de uma função em seus cotidianos. Para Lima (2003, p. 44), de maneira mais pragmática, dados os conjuntos A e B, uma função de domínio A e contradomínio B seria “uma regra, um conjunto de instruções, uma correspondência, uma lei que permite associar, sem restrições nem ambigüidade, a cada elemento x do conjunto A um elemento $f(x)$ do conjunto B”.

Assim como Lima (2003), entendemos que a ideia de função enquanto regra ou lei que permita obter $f(x)$ a partir de x não torna o conceito vago ou não matemático. Como argumentam Brito e Almeida (2005), é preciso levar em conta a possibilidade de conceituações intermediárias de função, valorizando os significados que advêm tanto de seu uso teórico quanto prático. Desse modo, no contexto deste trabalho, como Lima et al. (2006), entendemos uma função como uma regra que associa elementos entre dois conjuntos:

Dados os conjuntos X, Y, uma função $f: X \rightarrow Y$ (lê-se uma função de X em Y) é uma regra, ou um conjunto de instruções, que diz como associar a cada elemento $x \in X$ um elemento $y = f(x) \in Y$. O conjunto X chama-se o *domínio* e Y é o *contradomínio* da função f. Para cada $x \in X$, o elemento $f(x) \in Y$ chama-se a

imagem de x pela função f , ou o valor assumido pela função f no ponto $x \in X$. Escreve-se $x \mapsto f(x)$ para indicar que f transforma (ou leva) x em $f(x)$.

Mais especificamente, de acordo com Lima et al. (2006), “uma função $f: \mathbb{R} \rightarrow \mathbb{R}$ chama-se afim quando existem constantes $a, b \in \mathbb{R}$ tais que $f(x) = ax+b$ para todo $x \in \mathbb{R}$ ”. Nota-se, aqui, que alguns casos específicos emergem quando as constantes a e b em $y = ax+b$ assumem determinados valores. É o caso da função identidade, definida como $f: \mathbb{R} \rightarrow \mathbb{R} / f(x) = x$, para todo $x \in \mathbb{R}$. Os mesmos autores também destacam como casos particulares das funções afins as translações $f: \mathbb{R} \rightarrow \mathbb{R}, f(x) = x+b$, as funções lineares $f(x) = ax$ e as funções constantes $f(x) = b$.

Cabe ressaltar, no entanto, que por mais que as definições que tomaremos por base neste trabalho tenham caráter formal, as mesmas são mais adequadas ao contexto do Ensino Médio por não apresentarem uma linguagem excessivamente rebuscada e marcada pela simbologia de conjuntos. Ademais, a ideia de função enquanto uma relação ou lei que associa elementos entre dois conjuntos dialoga com os objetivos da pesquisa na medida em que se pretende, através da construção de programas de computador, resolver problemas concretos modeláveis por meio da função afim. Nesse cenário, encontrar a lei que determina como os elementos dos conjuntos são associados – em geral, a expressão analítica da função – é uma parte relevante do processo de construção do código do programa e de resolução de problema, o que inviabiliza uma abordagem “conjuntista” nos moldes de Bourbaki. Assim, a definição apresentada por Lima (2006) nos parece a mais adequada na medida em que formaliza o conceito em uma perspectiva teórica e, ao mesmo tempo, abre espaço para outras possibilidades de representação semiótica e aplicações em contextos concretos.

A dimensão didática

A dimensão didática das análises preliminares pressupõe um estudo acerca do funcionamento do sistema de ensino, com o objetivo de elucidar como o tema a ser desenvolvido na sequência didática é abordado na atualidade, tanto a nível de currículo quanto a nível de práticas conduzidas e recursos utilizados em sala de aula. No contexto deste trabalho, buscamos identificar o que se menciona em relação ao tema em documentos orientadores da educação nacional. Os resultados identificados são apresentados a seguir.

A Base Nacional Comum Curricular (BNCC) é um documento de caráter normativo, utilizado em todo o território nacional, que define o conjunto de aprendizagens essenciais que todos os alunos devem desenvolver ao longo das diferentes etapas e modalidades da Educação Básica (Brasil, 2018). Desse modo, trata-se do principal documento a ser tomado como base para elaboração dos currículos nas redes federal, estadual e municipal de ensino. Por esse motivo, tal documento foi utilizado como base para o estudo de como os temas centrais da sequência didática a ser elaborada são abordados em termos de legislação educacional.

De maneira geral, a BNCC destaca o desenvolvimento tecnológico como marca da contemporaneidade, em um cenário em que a computação e as tecnologias digitais de informação e comunicação (TDICs) estão cada vez mais presentes na vida de todos. Considerando este cenário, o documento coloca que é preciso assegurar aos estudantes “aprendizagens para atuar em uma sociedade em constante mudança, prepará-los para profissões que ainda não existem, para usar tecnologias que ainda não foram inventadas e para resolver problemas que ainda não conhecemos” (Brasil, 2018, p. 473). Além disso, se reforça a necessidade de compreender as novas formas de processar, transmitir e distribuir a informação de maneira segura e confiável em diferentes artefatos digitais, apropriando-se das linguagens da cultura digital e de ferramentas de *software* e aplicativos para explorar e produzir conteúdos em diversas mídias. Destaca-se, ainda, a necessidade de estimular a implementação de soluções envolvendo diferentes tecnologias para “identificar, analisar, modelar e solucionar problemas complexos em diversas áreas da vida cotidiana, explorando de forma efetiva o raciocínio lógico, o pensamento computacional, o espírito de investigação e a criatividade” (Brasil, 2018, p. 475).

Quanto à relação entre novas tecnologias e a aprendizagem da Matemática, o documento reforça a necessidade de integração de diferentes recursos didáticos, incluindo ferramentas tecnológicas digitais como calculadoras e planilhas eletrônicas. De acordo com a base curricular, tal fato deve ocorrer durante todo o processo de escolarização, de modo que seja possível a construção de situações de aprendizagem que promovam a reflexão e contribuam para a sistematização e formalização de conceitos matemáticos. Além disso, se aponta a importância do uso de ferramentas de *software* e aplicativos para compreensão e produção de conteúdos em diversas mídias, simulação de fenômenos e processos e elaboração e exploração dos diversos registros de representação matemática (Brasil, 2018).

O texto da BNCC menciona, ainda, o fato de que o ensino da Matemática não pode desvincular-se do contexto histórico e cultural em que se situa, na medida em que a disciplina é uma ciência humana, fruto das necessidades e preocupações de diferentes culturas em diferentes momentos históricos, contribuindo para solucionar problemas científicos e tecnológicos em um dado contexto (Brasil, 2018). Nesse sentido, considerando que na atualidade os estudantes estão imersos em uma cultura digital, na qual tecnologias digitais permeiam diferentes espaços sociais e reestruturam a sociedade, é importante que tal realidade se reflita na vivência escolar. Na medida em que esses artefatos digitais produzem transformações no âmbito comunicacional, informacional e interacional, construindo novas formas de relação entre o homem e as tecnologias e influenciando o processo de produção e construção do conhecimento, é essencial que essa temática adentre a escola e recursos dessa natureza sejam explorados em situações de aprendizagem (Souza; Tomanini; Santos, 2020).

De maneira mais específica, a BNCC destaca em diversos momentos a importância do desenvolvimento do pensamento computacional, especialmente em situações de aprendizagem da Matemática. De acordo com o texto do documento, na disciplina os estudantes devem ser capazes de formular e resolver problemas em contextos diversos, além de realizar transformações de situações em outras linguagens, como interpretar situações-problema apresentadas em língua materna e representá-las através de fórmulas, tabelas e gráficos (Brasil, 2018). Coloca-se, ainda, que a resolução de problemas, o desenvolvimento de projetos e a modelagem matemática são ricas oportunidades para o desenvolvimento do pensamento computacional:

Os processos matemáticos de resolução de problemas, de investigação, de desenvolvimento de projetos e da modelagem podem ser citados como formas privilegiadas da atividade matemática [...]. Esses processos de aprendizagem são potencialmente ricos para o desenvolvimento de competências fundamentais para o letramento matemático (raciocínio, representação, comunicação e argumentação) e para o desenvolvimento do pensamento computacional (Brasil, 2018, p. 267).

Em complemento à BNCC, em 2022 foi publicada a Resolução CNE/CEB 1/2022, definindo normas sobre a Computação na Educação Básica. O documento coloca que processos e aprendizagens referentes à Computação, no âmbito da Educação Básica, devem ser implementados considerando os textos da BNCC, das normas educacionais previstas na legislação, e também da referida resolução. Menciona-se, ainda, que a

formulação e o desenvolvimento dos currículos devem considerar um conjunto de competências e habilidades anexas à resolução (Brasil, 2022).

No que se refere à Educação Infantil e ao Ensino Fundamental, os objetivos de aprendizagem (Educação Infantil) e os objetos do conhecimento e habilidades a serem desenvolvidas (Ensino Fundamental) são organizados de acordo com três eixos: pensamento computacional, mundo digital e cultura digital. Além disso, em cada eixo as habilidades são detalhadas de acordo com os anos de escolarização, constituindo-se um percurso formativo com complexidade gradativamente maior, conforme o estudante avança em sua trajetória escolar.

Com relação ao Ensino Médio, por outro lado, são listadas sete competências específicas, cada uma se relacionando a uma série de habilidades a serem desenvolvidas ao longo dos anos de escolarização. No entanto, não se detalha em que ano cada habilidade poderia ser trabalhada, nem se menciona explicitamente o pensamento computacional em nenhuma parte do texto, embora se aborde a resolução de problemas em diversos momentos.

Já o conceito de função é mencionado em diferentes trechos na área da Matemática da BNCC. A noção surge já no texto referente aos anos iniciais do Ensino Fundamental, ainda que de maneira intuitiva, sem a utilização de letras para a representação de regularidades. De acordo com o documento, nesta etapa de escolarização “a noção intuitiva de função pode ser explorada por meio da resolução de problemas envolvendo a variação proporcional direta entre duas grandezas (sem utilizar a regra de três)” (Brasil, 2018, p. 270).

No texto referente aos anos finais do Ensino Fundamental, por outro lado, já descreve-se a necessidade de que os estudantes estabeleçam conexões entre variável e função e entre incógnita e equação. De maneira mais específica, o conceito passa a ser abordado formalmente a partir do nono ano (objeto de conhecimento “*Funções: representações numérica, algébrica e gráfica*” da unidade temática “*Álgebra*”). Em relação às habilidades a serem desenvolvidas, menciona-se a necessidade de “compreender as funções como relações de dependência unívoca entre duas variáveis e suas representações numérica, algébrica e gráfica e utilizar esse conceito para analisar situações que envolvam relações funcionais entre duas variáveis” (Brasil, 2018, p. 316).

No entanto, verifica-se que a noção de função é abordada majoritariamente nas diretrizes curriculares para o Ensino Médio. No Quadro 11 são apresentados todos os

momentos em que são mencionadas competências e habilidades relacionadas especificamente a funções.

Quadro 11: Habilidades da BNCC que envolvem o conceito de função.

Competência específica	Habilidade
<p>Competência específica 1: Utilizar estratégias, conceitos e procedimentos matemáticos para interpretar situações em diversos contextos, sejam atividades cotidianas, sejam fatos das Ciências da Natureza e Humanas, das questões socioeconômicas ou tecnológicas, divulgados por diferentes meios, de modo a contribuir para uma formação geral.</p>	<p>(EM13MAT101) Interpretar criticamente situações econômicas, sociais e fatos relativos às Ciências da Natureza que envolvam a variação de grandezas, pela análise dos gráficos das funções representadas e das taxas de variação, com ou sem apoio de tecnologias digitais.</p>
<p>Competência específica 3: Utilizar estratégias, conceitos, definições e procedimentos matemáticos para interpretar, construir modelos e resolver problemas em diversos contextos, analisando a plausibilidade dos resultados e a adequação das soluções propostas, de modo a construir argumentação consistente.</p>	<p>(EM13MAT302) Construir modelos empregando as funções polinomiais de 1º ou 2º grau, para resolver problemas em contextos diversos, com ou sem apoio de tecnologias digitais.</p>
	<p>(EM13MAT304) Resolver e elaborar problemas com funções exponenciais nos quais seja necessário compreender e interpretar a variação das grandezas envolvidas, em contextos como o da Matemática Financeira, entre outros.</p>
	<p>(EM13MAT305) Resolver e elaborar problemas com funções logarítmicas nos quais seja necessário compreender e interpretar a variação das grandezas envolvidas, em contextos como os de abalos sísmicos, pH, radioatividade, Matemática Financeira, entre outros.</p>
	<p>(EM13MAT306) Resolver e elaborar problemas em contextos que envolvem fenômenos periódicos reais (ondas sonoras, fases da lua, movimentos cíclicos, entre outros) e comparar suas representações com as funções seno e cosseno, no plano cartesiano, com ou sem apoio de aplicativos de álgebra e geometria.</p>
<p>Competência específica 4: Compreender e utilizar, com flexibilidade e precisão, diferentes registros de representação matemáticos (algébrico, geométrico, estatístico, computacional etc.), na busca de solução e comunicação de resultados de problemas.</p>	<p>(EM13MAT401) Converter representações algébricas de funções polinomiais de 1º grau em representações geométricas no plano cartesiano, distinguindo os casos nos quais o comportamento é proporcional, recorrendo ou não a softwares ou aplicativos de álgebra e geometria dinâmica.</p>

Competência específica	Habilidade
	<p>(EM13MAT402) Converter representações algébricas de funções polinomiais de 2º grau em representações geométricas no plano cartesiano, distinguindo os casos nos quais uma variável for diretamente proporcional ao quadrado da outra, recorrendo ou não a softwares ou aplicativos de álgebra e geometria dinâmica, entre outros materiais.</p>
	<p>(EM13MAT403) Analisar e estabelecer relações, com ou sem apoio de tecnologias digitais, entre as representações de funções exponencial e logarítmica expressas em tabelas e em plano cartesiano, para identificar as características fundamentais (domínio, imagem, crescimento) de cada função.</p>
	<p>(EM13MAT404) Analisar funções definidas por uma ou mais sentenças (tabela do Imposto de Renda, contas de luz, água, gás etc.), em suas representações algébrica e gráfica, identificando domínios de validade, imagem, crescimento e decrescimento, e convertendo essas representações de uma para outra, com ou sem apoio de tecnologias digitais.</p>
<p>Competência específica 5: Investigar e estabelecer conjecturas a respeito de diferentes conceitos e propriedades matemáticas, empregando estratégias e recursos, como observação de padrões, experimentações e diferentes tecnologias, identificando a necessidade, ou não, de uma demonstração cada vez mais formal na validação das referidas conjecturas.</p>	<p>(EM13MAT501) Investigar relações entre números expressos em tabelas para representá-los no plano cartesiano, identificando padrões e criando conjecturas para generalizar e expressar algebricamente essa generalização, reconhecendo quando essa representação é de função polinomial de 1º grau.</p>
	<p>(EM13MAT502) Investigar relações entre números expressos em tabelas para representá-los no plano cartesiano, identificando padrões e criando conjecturas para generalizar e expressar algebricamente essa generalização, reconhecendo quando essa representação é de função polinomial de 2º grau do tipo $y = ax^2$.</p>
	<p>(EM13MAT503) Investigar pontos de máximo ou de mínimo de funções quadráticas em contextos envolvendo superfícies, Matemática Financeira ou Cinemática, entre outros, com apoio de tecnologias digitais.</p>
	<p>(EM13MAT506) Representar graficamente a variação da área e do perímetro de um polígono regular quando os comprimentos de seus lados variam, analisando e classificando as funções envolvidas.</p>

Competência específica	Habilidade
	(EM13MAT507) Identificar e associar progressões aritméticas (PA) a funções afins de domínios discretos, para análise de propriedades, dedução de algumas fórmulas e resolução de problemas.
	(EM13MAT508) Identificar e associar progressões geométricas (PG) a funções exponenciais de domínios discretos, para análise de propriedades, dedução de algumas fórmulas e resolução de problemas.

Fonte: elaborado pelo autor (2024).

Observa-se, dessa forma, que a função afim é mencionada explicitamente apenas na habilidade *EM13MAT507*. No entanto, tal conceito também se relaciona com as habilidades *EM13MAT101*, *EM13MAT302*, *EM13MAT401*, *EM13MAT404*, *EM13MAT501* e *EM13MAT506*, na medida em que a função afim é um tipo de função polinomial do primeiro grau que pode ser utilizada para representar diferentes fenômenos. A partir da análise do Quadro 11, é possível notar que as habilidades que se relacionam com as funções afim envolvem, de maneira geral: a construção de modelos que empreguem funções afins para a resolução de problemas diversos; a construção de gráficos que representem geometricamente as funções estudadas; a identificação e análise de funções em suas diferentes representações, bem como a capacidade de conversão entre os distintos sistemas de representação; e a interpretação crítica de fenômenos naturais, econômicos e sociais a partir da análise de gráficos de funções.

Por fim, merece destaque o fato de que embora não se estabeleça a obrigatoriedade da utilização de recursos digitais para o desenvolvimento de tais habilidades, em todos os casos se menciona a possibilidade de utilização de *softwares* ou aplicativos de geometria dinâmica, ou demais tecnologias de natureza digital. Tal fato representa um avanço na medida em que tais possibilidades não eram mencionadas em nenhum momento nos Parâmetros Curriculares Nacionais (PCNs), que descreviam as diretrizes para os currículos da Educação Básica antes da publicação da BNCC.

A dimensão cognitiva

Como coloca Sierpinska (1992), as dificuldades apresentadas por estudantes no desenvolvimento do conceito de função são bem conhecidas e amplamente reportadas na literatura. Uma dificuldade comum, e talvez a mais significativa, é o estabelecimento de relações entre as diferentes formas de representação. De fato, o conceito de função pode ser desenvolvido a partir de múltiplas abordagens, movimentando diferentes registros semióticos e a necessidade de realizar operações de conversão entre eles. A relação entre grandezas distintas que uma função estabelece pode ser representada por tabelas, diagramas, relações algébricas e gráficos, entre outras formas de representação. Compreender esta pluralidade, bem como transitar entre estes diferentes formatos, muitas vezes traz problemas para os estudantes.

Outro problema comum no estudo de funções reside na ideia de generalização. Com o objetivo de investigar a descrição em língua natural das representações gráficas e dos procedimentos para a resolução de atividades envolvendo a função afim, Lorencini, Nogueira e Rezende (2020) propõem uma sequência de tarefas para 22 estudantes do segundo ano do Ensino Médio, entre eles uma com baixa visão. As autoras identificam que os participantes mobilizaram noções base como dependência, regularidade, variável e correspondência, mas não conseguiram mobilizar a ideia de generalização, mesmo em tarefas em que era solicitada explicitamente a forma geral da função.

Já Silva e Nogueira (2022), em estudo que objetivava encontrar razões para as dificuldades de estudantes de diferentes níveis de escolarização relacionadas ao processo de generalização da função afim, ressaltam a importância do desenvolvimento de sequências didáticas que abordem a lei de formação da função. Um ponto fundamental nesse sentido, segundo as autoras, é proporcionar aos estudantes uma pluralidade de situações nas quais seja possível identificar aspectos essenciais como: quais as informações relevantes no problema proposto; o que se está colocando em correspondência; quais variáveis estão em jogo; e quais delas são dependentes e independentes.

Na mesma linha, outros estudos também evidenciam as dificuldades apresentadas pelos estudantes no que se refere à ideia de generalização, essencial para o desenvolvimento do conceito de função (Calado; Rezende, 2022; Rezende; Nogueira; Calado, 2020). Particularmente em situações de modelagem matemática, nas quais é necessário definir as variáveis de interesse e os parâmetros do problema para, em

seguida, construir uma relação capaz de representar adequadamente um fenômeno observado, a capacidade de generalização adquire importância ainda maior.

Por sua vez, Dornelas (2007) argumenta que o estudo da função $f(x) = ax + b$ é muitas vezes vinculado às ideias da Geometria Analítica, onde o número real a é associado à tangente do ângulo que a reta correspondente ao gráfico da função faz com o eixo das abscissas no plano cartesiano, e o número real b representa o ponto onde tal reta intercepta o eixo das ordenadas. Para a autora, a partir de tal enfoque o estudo da função afim adquire um caráter estático, na medida em que os estudantes associam uma função à figura de uma reta, e não a uma relação entre grandezas.

Os aspectos cognitivos relacionados ao desenvolvimento do pensamento computacional também devem ser considerados. Segundo Lodi e Martini (2021), a ideia não gira em torno de transformar crianças em desenvolvedores de *softwares*, mas sim fornecer ferramentas que permitam ao estudante compreender e atuar na cultura digital na qual estão inseridos. Na perspectiva da educação inclusiva, o fornecimento dessas ferramentas passa pelo planejamento e condução de atividades de aprendizagem que atendam às especificidades educacionais dos alunos em sua individualidade, bem como às múltiplas formas de interação entre os alunos, professores e conteúdo (Camargo; Nardi, 2008). Nesse contexto, ao planejar situações de aprendizagem para o desenvolvimento de habilidades de programação de computadores para grupos onde haja pessoas com deficiência visual, a escolha do ambiente de programação que será utilizado é crucial, na medida em que os recursos utilizados devem possibilitar a participação e interação de todos os envolvidos. Assim, considerando os problemas de acessibilidade comumente apresentados por linguagens de programação baseadas em blocos, é importante levar em conta as alternativas disponíveis.

Utreras e Pontelli (2021) argumentam que se por um lado a utilização de linguagens de programação textuais não seja comum em uma abordagem introdutória junto a jovens estudantes, por outro, novos recursos têm sido desenvolvidos no sentido de facilitar o processo de programação neste contexto. Dessa forma, se ler, analisar e construir códigos de programação pode ser difícil em linguagens textuais mais tradicionais como FORTRAN, C ou Perl, em contrapartida novas linguagens de programação, como Java e Python, foram desenvolvidas com esforços no sentido de promover maior facilidade na leitura e escrita de códigos (Humble; Mozelius; Sällvin, 2019). Nesta perspectiva, como

mencionado anteriormente, a utilização de linguagens algorítmicas baseadas em português estruturado constitui uma alternativa viável.

Por fim, em revisão sistemática de literatura, Barcelos et al. (2015) concluem que há um crescente interesse da comunidade científica em explorar as relações entre o pensamento computacional e a matemática, especialmente através de experiências didáticas cujo público-alvo são alunos da Educação Básica. Outros trabalhos de revisão de literatura (Bordini et al., 2017; Nascimento; Santos; Tanzi, 2018; Werlich; Kemczinski; Gasparini, 2018) contribuem para a construção de um panorama sobre as principais abordagens e metodologias utilizadas em estudos sobre o tema. No entanto, embora sejam identificados diversos trabalhos apresentando diferentes abordagens, em nenhum dos mapeamentos foi apontado qualquer estudo associando resolução de problemas, pensamento computacional e deficiência visual na Educação Básica.

6.1.2. A FASE DE CONCEPÇÃO E ANÁLISE A PRIORI

Tomando como base o conjunto de informações reunidas através das análises preliminares, iniciou-se o processo de concepção da sequência didática. Para tal, partimos do fato de que o objetivo da proposta é o desenvolvimento de habilidades do pensamento computacional através da construção de programas de computador. Em segundo lugar, consideramos que trabalhos relacionados ao tema apontam vantagens na utilização de pseudolinguagens de programação baseadas na linguagem natural do estudante (Cambranes, 2013; Oliveira; Monteiro; Roman, 2013; Souza; Batista; Barbosa, 2016), e que o ILA é uma pseudolinguagem de programação baseada em português estruturado cuja estrutura possibilita a construção de programas que modelem determinados problemas relacionados à função afim. Desse modo, optamos pela utilização desta pseudolinguagem como recurso educacional para o desenvolvimento da proposta. Os roteiros elaborados para todos os encontros da sequência encontram-se no Apêndice D. A seguir, tais roteiros são detalhados e é realizada a análise a priori da intervenção.

Encontro 1

Para o primeiro encontro da sequência didática planeja-se, inicialmente, um diálogo introdutório em relação ao processo de programação de computadores, às linguagens

de programação e, mais especificamente, ao ILA. Além disso, deseja-se desenvolver as noções de entrada, processamento e saída de dados. Assim, os objetivos do encontro, os recursos utilizados e a avaliação foram planejados neste sentido (Figura 23).

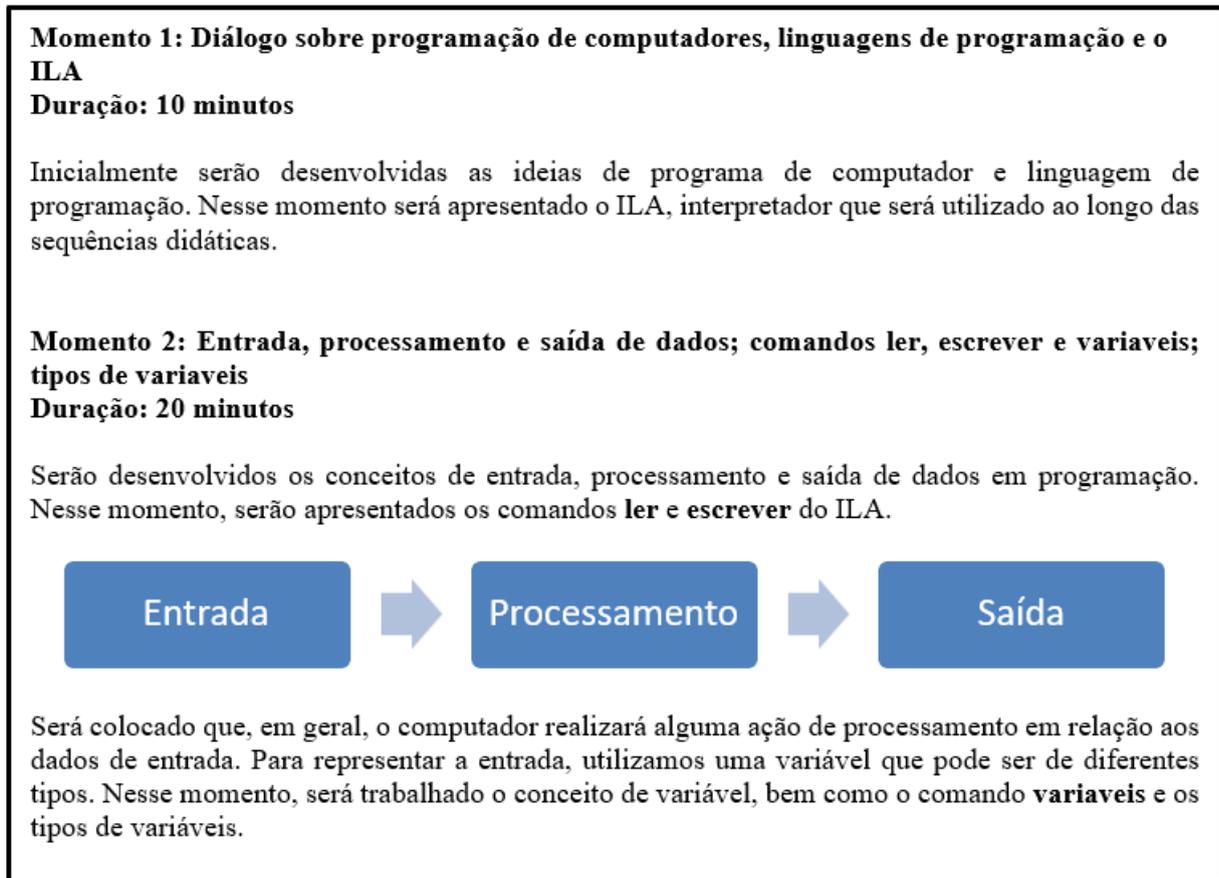
Figura 23: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 1.

<u>ENCONTRO 1</u>	
Temas:	<ul style="list-style-type: none"> • Programação de computadores e o ILA; • Entrada, processamento e saída de dados.
Objetivos:	<ul style="list-style-type: none"> • Dialogar sobre os conceitos de programação de computadores e linguagens de programação; • Apresentar o ILA; • Desenvolver as ideias de entrada, processamento e saída de dados; • Apresentar os comandos início, fim, variáveis, ler e escrever do ILA.
Habilidades do pensamento computacional priorizadas:	<ul style="list-style-type: none"> • Reconhecimento de padrões, decomposição, algoritmos.
Recursos:	<ul style="list-style-type: none"> • Computadores com ILA e NVDA instalados.
Avaliação:	<ul style="list-style-type: none"> • Construção de um programa de computador que receba uma informação e apresente na tela uma frase que contenha a informação recebida.

Fonte: elaborado pelo autor (2024).

O primeiro momento do encontro compreende um diálogo sobre os conceitos de programação de computadores, linguagens de programação e o ILA. Com isso, pretende-se identificar experiências e conhecimentos prévios dos estudantes no que se refere ao processo de programação de computadores, bem como construir uma definição para tais conceitos. Além disso, pretende-se introduzir os participantes ao funcionamento do ILA. Na medida em que, em geral, os estudantes da Educação Básica não possuem componentes curriculares na educação formal em que são exploradas habilidades de programação de computadores, estima-se que a maior parte dos participantes não tenha participado de experiências anteriores neste sentido.

Figura 24: Momentos 1 e 2 do encontro 1.



Fonte: elaborado pelo autor (2024).

No segundo momento do encontro objetiva-se discutir as ideias de entrada, processamento e saída de dados. Aqui pretende-se colocar que, em geral, o computador realiza alguma ação de processamento a partir de dados que são introduzidos pelo usuário (entrada). Em seguida, retorna dados processados através de algum dispositivo (saída). Cabe colocar que tal modelo é uma generalização, sendo possível a construção de programas que não apresentem dados de entrada ou saída. Em seguida, serão apresentados os comandos utilizados para representar entradas e saídas no ILA, “ler” e “escrever”. Aqui, espera-se que os estudantes relacionem a palavra/comando “ler” com a ideia de entrada, na medida em que o computador realizaria uma “leitura” dos dados que são introduzidos pelo usuário. Da mesma forma, é esperado que os participantes relacionem a palavra/comando “escrever” com a ideia de saída, pelo fato de que o computador “escreveria” uma informação na tela.

Figura 25: Momento 3 do encontro 1.

Momento 3: estrutura de um programa em ILA; comandos inicio e fim; bloco de instruções.
Duração: 20 minutos

Parte 1: Apresentação da estrutura de um programa em ILA. Nesse momento serão apresentados os comandos **inicio** e **fim**. Colocar que no ILA os programas são constituídos, em geral, utilizando a seguinte estrutura:

- Declaração de variáveis
- Comando inicio
- Blocos de instruções
- Comando fim

Parte 2: Análise de dois programas:

Programa 1:
 inicio
 escrever "Eu sou estudante do Colegio Pedro II"
 fim

Programa 2:
 variaveis
 caracter nome
 inicio
 escrever "Digite seu nome"
 ler nome
 escrever nome, "e estudante do Colegio Pedro II"
 fim

Fonte: elaborado pelo autor (2024).

No terceiro momento do encontro, inicialmente será apresentada a estrutura básica de um programa na pseudolinguagem ILA. Aqui, além de serem introduzidos os comandos "inicio" e "fim", utilizados para iniciar e finalizar o bloco de instruções no programa, serão apresentadas as noções de variável, declaração de variáveis e tipo de variável.

Em seguida, os participantes serão convidados a fazerem a leitura dos códigos de dois programas constituídos utilizando o ILA. O objetivo é que os estudantes sejam familiarizados com a estrutura da pseudolinguagem e identifiquem os comandos mencionados anteriormente. Além disso, pretende-se que, a partir de uma análise dos símbolos utilizados em português estruturado, os participantes reflitam acerca das ações que são realizadas pelo programa. Nesse sentido, o Quadro 12 e o Quadro 13 mostram as perguntas a serem conduzidas.

Quadro 12: Perguntas elaboradas e ações esperadas em relação ao programa 1.

Pergunta prevista	Ação esperada
Esse programa tem dados de entrada, de saída ou os dois?	<i>Espera-se que o participante compreenda que o programa tem apenas dados de saída.</i>
O que você acredita que acontecerá nesse programa?	<i>Espera-se que o participante compreenda que será exibida a frase “Eu sou estudante do Colégio Pedro II” na tela do computador.</i>

Fonte: elaborado pelo autor (2024).

Quadro 13: Perguntas elaboradas e ações esperadas em relação ao programa 2.

Pergunta prevista	Ação esperada
Esse programa tem dados de entrada, de saída ou as duas?	<i>Espera-se que o participante compreenda que o programa tem dados de entrada e de saída.</i>
O que acontece nas duas primeiras linhas do programa?	<i>Espera-se que o participante compreenda que nas duas primeiras linhas é realizada a declaração das variáveis que serão utilizadas no programa.</i>
Por que foi utilizado o termo caracter na segunda linha?	<i>Espera-se que o participante compreenda que o termo caracter foi utilizado para definir o tipo da variável declarada.</i>
E o termo nome da segunda linha, o que representa no programa?	<i>Espera-se que o participante compreenda que o termo nome representa a nomenclatura utilizada para a variável.</i>
Tanto nesse quanto no outro programa, que símbolo veio logo após a palavra escrever?	<i>Espera-se que o participante identifique que após o comando escrever, utiliza-se as aspas para delimitar a frase ou conjunto de símbolos que serão exibidos como saída.</i>
O que você acredita que acontecerá nesse programa?	<i>Espera-se que o participante compreenda que o nome do usuário será solicitado na tela do computador e, em seguida, será exibida a frase “[Nome] e estudante do Colégio Pedro II”, onde [Nome] representa o que foi digitado pelo usuário.</i>

Fonte: elaborado pelo autor (2024).

Cabe destacar, aqui, a crescente complexidade dos questionamentos colocados aos participantes. No que se refere ao primeiro programa, optou-se por uma estrutura simples, de modo que além da identificação da existência do comando de saída (escrever) e da inexistência do comando de entrada (ler), fosse possível que o participante realizasse inferências acerca do funcionamento e do objetivo do programa a partir de uma análise dos termos utilizados em português estruturado. Já em relação

ao segundo programa é possível que, dada a novidade na estrutura de representação do código, alguns participantes apresentem dificuldades na compreensão do funcionamento do algoritmo, especialmente no que se refere à natureza dos dados de saída. Nesse sentido, é possível que haja a compreensão de que a saída exibida na tela do computador seja do tipo “Nome e estudante do Colegio Pedro II”, onde o termo “Nome” representa a palavra “nome” em si, e não o termo digitado pelo usuário.

O quarto momento se refere à avaliação do encontro, consistindo na construção de um programa de computador utilizando o ILA. Tal programa deve ser capaz de solicitar um dado de entrada representado pela idade do usuário, e exibir um dado de saída na forma da frase “A idade do usuário e ‘tanto’”, onde o termo “tanto” representa o valor digitado pelo usuário. Espera-se que, nesse caso, o participante seja capaz de identificar que a variável a ser declarada no programa é idealmente do tipo numérico, embora seja possível a construção de tal programa declarando-se uma variável do tipo caracter.

Após a elaboração do código pelos participantes, no último momento do encontro será discutida a necessidade de avaliação do programa construído. Nesse sentido, os estudantes serão orientados a primeiramente verificar se o programa roda quando o arquivo é executado no prompt de comando. Caso o programa não rode, é necessário analisar a mensagem de erro retornada, realizando as alterações cabíveis no código do programa com o objetivo de solucionar o problema identificado. Tal operação deve ser realizada até que o programa rode.

Por fim, será discutido com os participantes que mesmo que o programa rode, é possível que seu funcionamento não esteja ocorrendo de forma adequada. Desse modo, é necessário realizar diferentes testes de modo que seja possível analisar se o programa funciona corretamente em diferentes cenários. Aqui, será reforçado aos estudantes a importância de determinar quais são estes possíveis cenários, de acordo com as características do programa construído.

Figura 26: Momentos 4 e 5 do encontro 1.

<p>Momento 4 (avaliação): construção de um programa em ILA Duração: 30 minutos</p> <p>Tomando por base as estruturas observadas nos programas analisados pelos estudantes, estes serão estimulados a construir um programa que cumpra as seguintes instruções:</p> <ul style="list-style-type: none"> • Perguntar ao usuário qual a sua idade. • Apresentar na tela do computador a seguinte frase “A idade do usuário e ‘tanto’”, onde “tanto” é a idade do usuário. <p>Momento 5: desenvolvendo a ideia de avaliação do programa construído (habilidade metacognitiva)</p> <p>Nesse momento, o estudante será estimulado a refletir sobre o programa construído, avaliando se o mesmo funciona adequadamente. Nesse sentido, o estudante deverá avaliar se:</p> <p>1- O programa roda?</p> <p>Caso não rode, verificar onde está o problema. Nesse momento, o estudante deverá refletir sobre a mensagem de erro retornada.</p> <p>2- O programa funciona adequadamente em diferentes casos?</p> <p>Para tal, deverão ser realizados diferentes testes de modo que se verifique se o retorno do programa é adequado em diferentes cenários. Nesse momento, perguntar aos estudantes que testes eles poderiam realizar para avaliar se o programa funciona adequadamente.</p>
--

Fonte: elaborado pelo autor (2024).

Quadro 14: Avaliação do encontro 1 e produção esperada dos participantes.

Avaliação	Produção esperada dos participantes
<p>Construir um programa que solicite a idade do usuário e retorne a frase “A idade do usuário e ‘tanto’”, onde ‘tanto’ é o valor digitado pelo usuário.</p>	<pre> variaveis numerico a inicio escrever "Digite a sua idade: " ler a escrever "A idade do usuário e: ", a fim </pre>

Fonte: elaborado pelo autor (2024).

Encontro 2

Realizado o encontro introdutório, espera-se que os participantes apresentem uma compreensão inicial em relação ao funcionamento do ILA, bem como estejam mais

familiarizados com a estrutura de um código simples e comandos como “início”, “fim”, “variáveis”, “ler” e “escrever”. Nesse cenário, para o segundo encontro objetiva-se apresentar determinados símbolos utilizados para representar operadores aritméticos no ILA. Além disso, pretende-se apresentar e discutir a estrutura de um programa onde tais operadores são utilizados. Por fim, espera-se que os estudantes sejam capazes de construir um código simples para resolver um problema que esteja relacionado com uma função afim.

Figura 27: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 2.

<p style="text-align: center;"><u>ENCONTRO 2</u></p> <p style="text-align: center;">Operadores matemáticos</p> <p>Objetivos:</p> <ul style="list-style-type: none">• Apresentar a representação de alguns operadores matemáticos no ILA;• Construir programas que realizem operações simples utilizando operadores matemáticos. <p>Habilidades do pensamento computacional priorizadas:</p> <ul style="list-style-type: none">• Decomposição, reconhecimento de padrões, algoritmos <p>Recursos:</p> <ul style="list-style-type: none">• Computadores com ILA e NVDA instalados <p>Avaliação:</p> <ul style="list-style-type: none">• Construção de um programa de computador que realize uma operação aritmética simples.

Fonte: elaborado pelo autor (2024).

No primeiro momento, após breve retomada dos conceitos desenvolvidos no encontro anterior, seguida de uma explanação inicial em relação aos objetivos e à organização do segundo encontro, serão apresentados alguns operadores matemáticos utilizados para representação de operações aritméticas no ILA. Aqui, optou-se por privilegiar os operadores da soma (+), subtração (-), multiplicação (*), divisão (/) e exponenciação (^), símbolos mais utilizados na representação algébrica de funções afins.

Quadro 15: Operadores e operações associadas no ILA.

Operador	Operação	Resultado
=	Atribui um valor a uma variável	Atribuição
+	Manutenção de sinal	Positivo
-	Inversão de sinal	Negativo
^	Exponenciação	Numérico
raiz(base, índice)	Raiz de índice qualquer	Numérico
resto(dividendo, divisor)	Resto de divisão	Numérico
/	Divisão com quociente real	Numérico
*	Multiplicação	Numérico
+	Adição	Numérico
-	Subtração	Numérico

Fonte: Manzano (2017).

No segundo momento do encontro será apresentado aos participantes o código de um programa construído para realizar uma operação de soma entre dois números. Espera-se que, analisando a estrutura do código, os participantes sejam capazes de realizar inferências em relação ao funcionamento do programa.

Figura 28: Momentos 1 e 2 do encontro 2.

Momento 1: apresentação dos principais operadores matemáticos e suas expressões no ILA
Duração: 10 minutos

Após apresentação dos objetivos do encontro, no primeiro momento serão apresentados os operadores matemáticos utilizados no ILA para realização de operações de soma, subtração, multiplicação, divisão e exponenciação.

Momento 2: estrutura de um programa que envolva operações aritméticas
Duração: 40 minutos

Nesse momento será apresentado aos estudantes o seguinte programa:

```

variaveis
numerico a, b, soma
inicio
escrever "Digite o primeiro numero:"
ler a
escrever "Digite o segundo numero:"
ler b
soma = a + b
escrever "O resultado equivale a: ", soma
fim

```

Fonte: elaborado pelo autor (2024).

Com o objetivo de auxiliar os estudantes na compreensão da estrutura do código e na identificação das ações realizadas pelo programa, será realizado o conjunto de perguntas apresentado no Quadro 16.

Quadro 16: Perguntas elaboradas e ações esperadas em relação ao programa.

Pergunta prevista	Ação esperada
Esse programa apresenta comandos de entrada, de saída ou os dois?	<i>Espera-se que o participante compreenda que o programa tem dados de entrada e de saída.</i>
O que isso significa?	<i>Espera-se que o participante compreenda que, pelo fato de existirem dados de entrada e de saída, serão solicitadas informações que deverão ser digitadas pelo usuário e serão exibidos dados processados na tela do computador.</i>
O que significam as duas primeiras linhas do programa?	<i>Espera-se que o participante compreenda que nas duas primeiras linhas é realizada a declaração das variáveis que serão utilizadas no programa.</i>
O que você acha que esse programa faz?	<i>Espera-se que o participante identifique que o programa solicita que o usuário digite dois números e realiza a operação de soma entre os mesmos, exibindo na tela o resultado da soma.</i>
O que você acha que acontece na linha 8 do programa?	<i>Espera-se que o participante infira que na linha 8 atribui-se à variável soma o resultado da operação de soma entre os dois números digitados pelo usuário.</i>
O que você acha que acontece na linha 9 do programa?	<i>Espera-se que o participante compreenda que na linha 9 determina-se que seja exibida na tela do computador a frase "O resultado equivale a 'soma'", onde 'soma' representa o resultado da adição entre os números digitados pelo usuário.</i>

Fonte: elaborado pelo autor (2024).

A partir das respostas dadas pelos participantes será discutido o algoritmo utilizado para a construção do programa, reforçando o significado de comandos abordados anteriormente e desenvolvendo novos conceitos e habilidades relacionados aos objetivos do encontro. Em especial, deseja-se discutir o que acontece nas linhas 8 e 9 do código, onde é definida a operação de processamento a ser realizada pelo computador, além de ser determinado de que modo os dados de saída serão apresentados ao usuário.

Nesse momento, pretende-se reforçar que o programa realiza uma operação de processamento a partir de dados de entrada que são inseridos pelo usuário, e que na construção do código devemos definir claramente a operação que deve ser feita,

atribuindo o resultado a uma nova variável. Além disso, deseja-se estimular que os estudantes reflitam acerca das mudanças que devem ser feitas no código para que outras operações sejam feitas pelo programa. Para tal, serão realizadas as perguntas exibidas no Quadro 17.

Quadro 17: Perguntas elaboradas em relação a mudanças no programa proposto e ações esperadas.

Pergunta prevista	Ação esperada
O que deveríamos modificar para criar um programa que multiplica dois números?	<i>Espera-se que o participante identifique que o sinal de adição na linha 8 do código deve ser alterado por um asterisco (*).</i>
O que deveríamos modificar para criar um programa que somasse três números?	<i>Espera-se que o participante identifique que deve ser declarada uma nova variável do tipo numérico, e que na linha 9 do código tal variável deve ser adicionada às duas já existentes.</i>

Fonte: elaborado pelo autor (2024).

Por mais que se espere que os participantes sejam capazes de realizar as alterações necessárias em ambos os casos, é possível que ocorram enganos em dois pontos que possivelmente representam novidades para os participantes. Primeiramente, como o símbolo asterisco não é usualmente utilizado para representar a operação de multiplicação na Educação Básica, é possível que ocorram erros nesse sentido. Considerando tal cenário, pretende-se neste momento reforçar quais os símbolos utilizados para representação das operações aritméticas no ILA. Por outro lado, em relação à segunda pergunta, é possível que participantes não se recordem de declarar, no início do programa, a nova variável a ser somada às duas anteriores. Tal possibilidade se deve ao fato esperado de os participantes ainda não estarem familiarizados com o ato de declarar variáveis na construção de um código.

Figura 29: Momento 3 do encontro 2.

<p>Momento 3: construção de um programa que realize uma operação aritmética simples Duração: 30 minutos</p> <p>Nesse momento, os estudantes serão estimulados a construírem um programa que calcule o quadrado de um número. Reforçar que os estudantes devem avaliar o programa construindo, criando uma série de testes para verificar se o programa funciona adequadamente.</p>

Fonte: elaborado pelo autor (2024).

Por fim, no momento de avaliação do encontro 2, os participantes serão estimulados a criarem um programa que realize uma operação de potenciação. Para isso, deverão realizar as alterações necessárias no programa elaborado para a soma de dois números inicialmente apresentado no encontro. Nesse caso, espera-se que os participantes recorram a uma das seguintes estratégias na construção do código: declaração de duas variáveis e utilização da operação de potenciação, através do símbolo “^”, para o cálculo do quadrado do número digitado; ou declaração de duas variáveis e utilização da operação de multiplicação, através do símbolo “*”, para a realização do produto do número digitado por ele mesmo.

Quadro 18: Avaliação do encontro 2 e produção esperada dos participantes.

Avaliação	Produção esperada dos participantes
<p>Construir um programa que solicite um número ao usuário e retorne o quadrado do número digitado</p>	<p>Primeira possibilidade:</p> <pre> variaveis numerico a, b inicio escrever "Digite o numero:" ler a b = a^2 escrever "O resultado equivale a: ", b fim </pre> <p>Segunda possibilidade:</p> <pre> variaveis numerico a, b inicio escrever "Digite o numero:" ler a b = a*a escrever "O resultado equivale a: ", b fim </pre>

Fonte: elaborado pelo autor (2024).

Encontro 3

No terceiro encontro da sequência didática pretende-se retomar o conceito de função afim e introduzir a noção de algoritmo. Como todos os participantes são estudantes do Ensino Médio, é esperado que já tenham começado a desenvolver a noção de função de uma variável na matemática escolar. Além disso, como o objetivo deste trabalho não é desenvolver o conceito de função, mas sim as habilidades do pensamento computacional através da resolução de problemas por meio da programação de computadores, pretende-se nesse momento apenas uma retomada de

elementos e significados relacionados ao conceito de função e, mais especificamente, da função afim. Por outro lado, é esperado que, em geral, os participantes não estejam familiarizados com a noção de algoritmo e, portanto, será conduzido um diálogo introdutório acerca do tema.

Figura 30: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 3.

ENCONTRO 3

Desenvolvendo a ideia de algoritmo e retomando o conceito de função

Objetivos:

- Apresentar a representação de novos operadores matemáticos no ILA;
- Rever o conceito de função afim;
- Construir um programa que envolva o conceito de função afim.

Habilidades do pensamento computacional priorizadas:

- Algoritmos, reconhecimento de padrões, abstração, decomposição

Recursos:

- Computadores com ILA e NVDA instalados

Avaliação:

- Construção de um programa de computador que envolva o conceito de função afim

Fonte: elaborado pelo autor (2024).

No primeiro momento do encontro, inicialmente serão lembrados os operadores e o programa construído no encontro anterior. Em seguida, será realizada uma conversa com os estudantes com o objetivo de identificar seus conhecimentos prévios em relação a funções de uma variável. Com base nas interações aqui conduzidas, será reforçada a ideia de função enquanto relação entre duas variáveis, bem como será caracterizada a função afim como uma relação do tipo $y = ax + b$, com $a, b \in \mathbb{R}$.

Em seguida, será trabalhada a noção de algoritmo enquanto sequência de passos que devem ser seguidos para solucionar um problema ou atingir um objetivo. Aqui, será realizada uma analogia com a ideia de uma “receita de bolo”, e os participantes serão estimulados a construírem um algoritmo que represente uma “ida à escola”.

Figura 31: Momentos 1, 2 e 3 do encontro 3.

<p>Momento 1: revendo o conceito de função Duração: 15 minutos</p> <p>Nesse momento será retomado junto aos estudantes o conceito de função. Inicialmente, buscaremos identificar os conhecimentos prévios dos estudantes em relação ao conteúdo. A partir das interações conduzidas, o objetivo é reforçar a ideia de função enquanto relação entre duas ou mais variáveis. Em seguida, será caracterizada a função do primeiro grau, bem como sua forma padrão $f(x)=ax+ b$, ou $y=ax+b$.</p> <p>Momento 2: desenvolvendo a ideia de algoritmo Duração: 15 minutos</p> <p>Nesse momento será apresentado o conceito de algoritmo aos estudantes. Aqui, buscaremos relacionar a ideia de algoritmo à noção de “receita”, uma sequência de passos que deverão ser seguidos para solucionar um problema ou atingir um objetivo. Inicialmente, será discutido o algoritmo de uma “ida à escola”.</p> <p>Momento 3: construindo um algoritmo para calcular o dobro de um número Duração: 20 minutos</p> <p>Estabelecendo uma relação entre as atividades anteriores, nesse momento os estudantes serão estimulados a construir um algoritmo para a criação de um programa que solicite um número ao usuário um número e retorne o dobro do mesmo. O objetivo é que os mesmos construam uma sequência similar à seguinte:</p> <ul style="list-style-type: none"> • Solicitar o número ao usuário • Armazenar o número • Realizar a operação de dobrar o número, atribuindo o resultado a uma variável • Exibir o valor do resultado na tela do computador
--

Fonte: elaborado pelo autor (2024).

No terceiro momento, objetivando criar um cenário que relacione os conceitos trabalhados anteriormente no encontro, os participantes serão estimulados a construir um algoritmo para a criação do código de um programa que solicite um número ao usuário e retorne o dobro do mesmo, o que retoma a ideia de função afim. Nesse ponto, cabe ressaltar que um mesmo problema pode ser resolvido de diferentes maneiras e, com isso, a criação do algoritmo em questão é subjetiva. Desse modo, é esperado que os participantes realizem produções que se diferenciem umas das outras. De maneira geral, uma possível resposta é ilustrada no Quadro 19.

Quadro 19: Algoritmo esperado dos participantes.

Proposta	Produção esperada
<p>Construção de um algoritmo para a criação de um programa que solicite um número ao usuário e retorne o dobro dele.</p>	<ul style="list-style-type: none"> • <i>Solicitar o número ao usuário;</i> • <i>Armazenar o número;</i> • <i>Realizar a operação de dobrar o número, atribuindo o resultado a uma variável;</i> • <i>Exibir o valor do resultado na tela do computador.</i>

Fonte: elaborado pelo autor (2024).

Por fim, na avaliação do encontro os participantes deverão converter o algoritmo elaborado para a pseudolinguagem ILA, construindo o código de um programa que solicite um número ao usuário e retorne o dobro do número digitado. Neste momento, será reforçada a necessidade de avaliação do programa construído através da realização de um conjunto de testes de acordo com o código elaborado.

Figura 32: Momento 4 do encontro 3.

<p>Momento 4 (construção do programa/avaliação): realizando a operação de conversão do algoritmo construído para a linguagem ILA Duração: 30 minutos</p> <p>Nesse momento, os estudantes serão estimulados a representarem o algoritmo desenvolvido utilizando a linguagem ILA. O objetivo é que os estudantes construam um programa que retorne o dobro de um número fornecido pelo usuário. Na mediação do processo, podem ser realizadas as seguintes perguntas aos estudantes: i) o que é o dobro de um número? ii) que operação devemos realizar para ter como resultado o dobro de um número? iii) você consegue pensar em alguma função que represente essa situação?</p> <p>Reforçar a ideia de que os estudantes devem testar o programa construído, avaliando se o mesmo roda adequadamente.</p>

Fonte: elaborado pelo autor (2024).

Quadro 20: Avaliação do encontro 3 e produção esperada dos participantes.

Avaliação	Produção esperada dos participantes
<p>Construir um programa que solicite um número ao usuário e retorne o seu dobro.</p>	<pre> variaveis numerico a, b inicio escrever "Digite o numero: " ler a b=2*a escrever "O dobro do numero digitado e: ", b fim </pre>

Fonte: elaborado pelo autor (2024).

Encontro 4

No quarto encontro da sequência será introduzida a noção de programação condicional. Optou-se por trabalhar estruturas que envolvem tomada de decisão pelo fato de possibilitarem a construção de programas mais elaborados, proporcionando situações ricas para o desenvolvimento do raciocínio lógico e de habilidades do pensamento computacional.

Figura 33: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 4.

<u>ENCONTRO 4</u>	
Programação condicional: a instrução se então	
Objetivos:	<ul style="list-style-type: none"> • Apresentar a representação de alguns operadores relacionais no ILA • Apresentar a instrução se então • Construir programas utilizando a instrução se então no ILA
Habilidades do pensamento computacional priorizadas:	<ul style="list-style-type: none"> • Abstração, decomposição, reconhecimento de padrões, algoritmos, avaliação
Recursos:	<ul style="list-style-type: none"> • Computadores com ILA e NVDA instalados
Avaliação:	<ul style="list-style-type: none"> • Construção de um programa de computador que utilize a instrução se então no ILA

Fonte: elaborado pelo autor (2024).

Após apresentação dos objetivos e da organização do encontro, e de uma breve retomada de conceitos trabalhados no encontro anterior, será realizada uma apresentação de alguns operadores relacionais comumente utilizados na construção de códigos que envolvem tomada de decisões no ILA, a saber: igual (=), maior (>), menor (<), maior ou igual (>=) e menor ou igual (<=). Nesse momento pretende-se verificar se os participantes conhecem o posicionamento de tais teclas no teclado e, em caso negativo, auxiliá-los no sentido de localizá-las.

Em seguida, será apresentada a instrução “se então” no ILA. Tal instrução é utilizada quando se deseja realizar um desvio condicional que, por sua vez, está associado à realização de decisões em um programa (Manzano, 2017). Para a introdução dos comandos, inicialmente será apresentado um código (Figura 34) que os participantes deverão analisar.

Figura 34: Momentos 1 e 2 do encontro 4.

<p>Momento 1: apresentação dos principais operadores relacionais e suas expressões no ILA Duração: 15 minutos</p> <p>Nesse momento serão apresentados os principais operadores utilizados para estabelecimento de uma relação entre elementos (>, <, =, >=, <=).</p> <p>Momento 2: apresentando a instrução se então Duração: 40 minutos</p> <p>Nesse momento será apresentada a instrução se então aos participantes. Reforçar que tal instrução será utilizada quando for necessário tomar uma decisão para realizar uma ação. Para tal, inicialmente será apresentado o seguinte programa:</p> <pre> variaveis numerico a inicio escrever "Digite o numero" ler a se (a<10) entao escrever "O numero digitado e menor que 10" fim_se fim </pre>
--

Fonte: elaborado pelo autor (2024).

Após leitura do código apresentado, objetivando estimular a reflexão e a produção de inferências em relação ao funcionamento do programa, serão feitas as perguntas exibidas no Quadro 21 aos participantes.

Quadro 21: Perguntas em relação ao programa e ações esperadas.

Pergunta prevista	Ação esperada
Nesse programa, em quais linhas há comandos que você ainda não conhecia?	<i>Espera-se que o participante identifique no código os comandos "se", "entao" e "fim_se", que ainda não haviam sido trabalhados em encontros anteriores.</i>
O que você acha que esse programa faz?	<i>Espera-se que o participante infira que o programa retornará a mensagem "O numero digitado e menor que 10" caso o número digitado pelo usuário seja menor que 10.</i>
O que você acha que aparecerá na tela se o usuário digitar o número 7?	<i>Espera-se que, nesse caso, o participante infira que o programa retornará a mensagem "O numero digitado e menor que 10".</i>
Considerando a maneira como o programa está escrito, o que você acha que irá acontecer se o usuário digitar o número 15?	<i>Espera-se que o participante perceba que, nesse caso, o programa não retornará nenhuma mensagem, já que o número digitado é maior que 10.</i>

Fonte: elaborado pelo autor (2024).

Pelo fato de o código apresentar uma composição baseada em português estruturado, sendo possível encontrar de maneira explícita a instrução “se então”, espera-se que os participantes compreendam, ainda que de maneira intuitiva, que o programa realizará uma atividade de processamento que dependerá da informação fornecida pelo usuário. Por outro lado, é possível que os participantes não percebam diretamente que, levando em conta a forma como o código foi construído, o programa não fornece dados de saída para valores digitados que sejam maiores ou iguais a 10. Caso o participante perceba tal fato, pretende-se questioná-lo em relação ao que poderia ser feito para contornar tal problema; ou seja, como o código poderia ser alterado de modo que o programa fornecesse uma saída adequada para valores digitados maiores ou iguais a 10. A partir do problema levantado e das interações conduzidas através da última questão realizada, um segundo código será apresentado aos participantes (Figura 35).

Figura 35: Segunda parte do momento 2 do encontro 4.

A partir das interações com os estudantes, perguntar se eles têm alguma sugestão para solucionar o problema ocorrido quando o usuário digitou o número 15. Em seguida, pedir que os estudantes observem o seguinte programa:

```

variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
fim_se
se (a>10) entao
escrever "O numero digitado e maior que 10"
fim_se
fim

```

Fonte: elaborado pelo autor (2024).

No segundo código é introduzida uma alteração em relação ao primeiro. Com ela, o programa passa a retornar uma saída quando são digitados valores maiores que 10. Objetivando que os participantes identifiquem tal alteração e reflitam sobre seu significado, serão realizadas as perguntas exibidas no Quadro 22.

Quadro 22: Perguntas em relação ao segundo programa e ações esperadas.

Pergunta prevista	Ação esperada
O que você acha que esse programa faz?	<i>Espera-se que o participante infira que o programa retornará a mensagem “O numero digitado e menor que 10” caso o número digitado pelo usuário seja menor que 10, ou a mensagem “O numero digitado e maior que 10” caso o número digitado pelo usuário seja maior que 10</i>
Qual a diferença desse programa em relação ao anterior?	<i>Espera-se que o participante perceba que no segundo programa há três linhas adicionais que contemplam os casos em que o número digitado pelo usuário é maior que 10.</i>
Você acha que agora o programa está construído da melhor maneira possível? Ou seja, como ele está escrito, existe uma resposta adequada para todos os possíveis valores digitados pelo usuário?	<i>Espera-se que o participante perceba que ainda é necessário considerar o caso em que o número digitado pelo usuário é o 10.</i>

Fonte: elaborado pelo autor (2024).

Por mais que se espere que os participantes identifiquem as alterações realizadas no código, bem como seus significados, é possível que haja dúvidas em relação à terceira pergunta. Considerando que estão cobertos os casos em que o número digitado é maior ou menor que 10, pode ocorrer que o participante conclua que todos os valores possíveis a serem digitados pelo usuário já estão contemplados no código. Sendo assim, são previstos os desdobramentos ilustrados no Quadro 23.

Quadro 23: Desdobramentos possíveis da terceira pergunta e ações a serem realizadas.

Desdobramento possível	Ação a ser realizada	
O estudante conclui equivocadamente que existe uma resposta adequada para todos os possíveis valores digitados pelo usuário	<i>Perguntar: “E se o usuário digitar o número 10?”</i>	
	Possibilidade 1: O participante percebe que o código não considera tal possibilidade.	Possibilidade 2: O participante não percebe que o código não considera tal possibilidade.
	<i>Ação: Solicitar que o participante faça a alteração necessária no programa de modo que seja considerada essa possibilidade.</i>	<i>Ação: Dialogar com o participante no sentido de conduzi-lo à compreensão de que tal possibilidade ainda não é contemplada no código, e solicitá-lo que faça a alteração necessária no programa de modo que seja considerada essa possibilidade.</i>
O estudante conclui adequadamente que não se leva em conta a possibilidade de digitação do número 10	<i>Solicitar que o participante faça a alteração necessária no programa de modo que seja considerada essa possibilidade.</i>	

Fonte: elaborado pelo autor (2024).

Partindo das interações conduzidas com base na análise dos dois programas apresentados, pretende-se reforçar que a instrução “se entao” segue a estrutura:

se (*proposição cuja veracidade será avaliada*) **entao**
ação a ser realizada se a proposição for verdadeira
fim_se

Será colocado que na estrutura observada, a proposição cuja veracidade será avaliada é representada entre parênteses entre os comandos “se” e “entao”. Além disso, a ação que será realizada se a condição atribuída for satisfeita - isto é, se a proposição criada for verdadeira – aparece após o comando “entao”, podendo ser de vários tipos. Nos códigos utilizados como exemplo a ação era representada pela exibição de uma informação na tela do computador, operacionalizada através do uso do comando “escrever”. No entanto, é possível que tal ação seja construída de diferentes formas, como determinando-se uma ação de processamento ou criando-se um bloco de instruções.

Por fim, será reforçada a importância de se considerar a variedade de cenários que podem emergir a partir de diferentes dados de entrada inseridos pelo usuário. Nos códigos ilustrados anteriormente, por exemplo, seria importante prever os diferentes valores que poderiam ser digitados pelo usuário e verificar se o programa apresenta um dado de saída adequado para cada caso. Desse modo, é essencial realizar uma bateria de testes ao fim do processo de construção do código, com vistas a avaliar se o programa funciona adequadamente em diferentes cenários. Tal prática representa, ainda, um momento potencial para o desenvolvimento de habilidades metacognitivas e de avaliação.

Figura 36: Momento 3 do encontro 4.

Momento 3 (avaliação): construindo um programa utilizando a instrução se entao
Duração: 25 minutos

Neste momento, os estudantes serão estimulados a construir um programa que remeta ao conceito de função afim, utilizando a instrução se entao. Para isso, apresentar a seguinte situação:

Em uma loja onde são vendidos materiais escolares, cada caderno custa R\$8,00. No entanto, a gerência criou uma promoção em que se o cliente compra a partir de 4 cadernos, cada um sai por R\$6,50.

Construa um programa que retorne o preço que o usuário irá pagar a partir da quantidade de cadernos comprada.

Fonte: elaborado pelo autor (2024).

O momento final do encontro corresponde à avaliação, que consiste na construção de um programa que envolva o conceito de função afim e apresente a instrução “se então”. Nesse sentido, os participantes deverão elaborar um código para um programa que modele o problema exposto na Figura 36. Para orientar os estudantes no processo de construção do algoritmo, poderão ser realizadas as seguintes perguntas orientadoras: *i) o valor a ser pago por cada caderno depende da quantidade de cadernos que serão comprados?; ii) nesse caso, podemos pensar em uma única função que poderia representar o valor total a ser pago pelo cliente?; iii) poderíamos dividir a situação descrita em duas ou mais partes? iv) qual seria a função que representaria o valor a ser pago em cada um desses casos?*

A partir das interações realizadas anteriormente, será discutida com os participantes a possibilidade de decompor um determinado problema em etapas menores. É o caso, por exemplo, do problema apresentado na avaliação do encontro, na medida em que é possível dividir a situação em dois cenários que deverão ser tratados de formas diferentes: um primeiro, que representa o caso em que o número de cadernos comprados é menor que 4; e um segundo, onde o número de itens adquiridos é maior ou igual a 4. Com isso, pretende-se introduzir a noção de decomposição, um dos pilares do pensamento computacional, reforçando a sua importância e estimulando seu desenvolvimento. Por fim, no Quadro 24 é apresentada uma possível produção esperada dos participantes.

Quadro 24: Avaliação do encontro 4 e produção esperada dos participantes.

Avaliação	Produção esperada dos participantes
<p>Em uma loja onde são vendidos materiais escolares, cada caderno custa R\$8,00. No entanto, a gerência criou uma promoção em que se o cliente compra a partir de 4 cadernos, cada um sai por R\$6,50.</p> <p>Construa um programa que retorne o preço que o usuário irá pagar a partir da quantidade de cadernos comprada.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de cadernos" ler a se (a<4) entao b=8*a fim_se se (a>=4) entao b=6.5*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>

Fonte: elaborado pelo autor (2024).

Encontro 5

O quinto encontro da sequência é o último onde são apresentados novos comandos. Dando seguimento ao desenvolvimento de conceitos relacionados à programação condicional, será introduzido o comando “senao”, que é utilizado para fazer uso de um desvio condicional composto, quando existem duas ações que podem ser tomadas de acordo com a veracidade ou não da proposição avaliada.

Figura 37: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 5.

<u>ENCONTRO 5</u>	
Programação condicional (continuação)	
Objetivos:	<ul style="list-style-type: none"> • Apresentar o comando senao no ILA • Construir programas utilizando a instrução senao no ILA
Habilidades do pensamento computacional priorizadas:	<ul style="list-style-type: none"> • Decomposição, reconhecimento de padrões, algoritmos
Recursos:	<ul style="list-style-type: none"> • Computadores com ILA e NVDA instalados
Avaliação:	<ul style="list-style-type: none"> • Construção de um programa de computador que utilize a instrução se então no ILA

Fonte: elaborado pelo autor (2024).

No primeiro momento do encontro, como habitual, primeiramente será realizada uma breve retomada dos conceitos e do programa trabalhados anteriormente. Em seguida, será introduzido o comando “senao” aos participantes. Para tal, o código exibido na Figura 38 será apresentado para análise dos estudantes.

Figura 38: Momento 1 do encontro 5.

Momento 1: apresentando a instrução `senao`
Duração: 40 minutos

Nesse momento, será apresentada a instrução `senao` no ILA. Para tal, inicialmente será retomado um programa trabalhado no encontro anterior, com uma pequena alteração incluindo um operador relacional. Nesse sentido, pedir que os estudantes observem o seguinte código:

```

variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
fim_se
se (a>=10) entao
escrever "O numero digitado não e menor que 10"
fim_se
fim

```

Fonte: elaborado pelo autor (2024).

O programa em questão se assemelha aos discutidos no encontro anterior, com a novidade da introdução de um operador relacional diferente e novos dados de saída. Para estimular a reflexão dos participantes, serão realizadas as perguntas exibidas no Quadro 23.

Quadro 25: Perguntas elaboradas e ações esperadas dos participantes.

Pergunta prevista	Ação esperada
Existe algum símbolo ou comando nesse programa que você ainda não tinha utilizado? Se sim, qual você acha que é o significado dele?	<i>Espera-se que o participante perceba que não é utilizado nenhum símbolo ou comando que ainda não havia sido mencionado.</i>
O que você acha que esse programa faz?	<i>Espera-se que o participante infira que o programa solicita um número ao usuário e retorna a mensagem "O numero digitado e menor que 10", caso o número digitado seja menor que 10, ou a mensagem "O numero digitado nao e menor que 10", caso o número digitado seja maior ou igual a 10.</i>

Fonte: elaborado pelo autor (2024).

Como o código exibido não apresenta novidades em termos de conceitos ou comandos utilizados, espera-se que os participantes não demonstrem maiores dificuldades em sua compreensão. Desse modo, dando seguimento à situação de aprendizagem, será apresentado o código de um programa análogo ao anterior. No entanto, o novo algoritmo utiliza o comando `senao` (Figura 39).

Figura 39: Segunda parte do momento 1 do encontro 5.

Em seguida, pedir que os estudantes observem o seguinte programa:

```

variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
senao
escrever "O numero digitado nao e menor que 10"
fim_se
fim

```

Fonte: elaborado pelo autor (2024).

Assim como na exploração do programa anterior, perguntas serão realizadas com o objetivo de estimular a reflexão dos participantes (Quadro 26).

Quadro 26: Perguntas elaboradas e ações esperadas dos participantes.

Pergunta prevista	Ação esperada
Existe algum símbolo ou comando nesse programa que você ainda não tinha utilizado? Se sim, qual você acha que é o significado dele?	<i>Espera-se que o participante perceba que o programa apresenta o comando <code>senao</code>, que ainda não havia sido trabalhado. Espera-se, também, que se infira que o comando é utilizado para determinar uma ação a ser executada caso uma condição inicial não seja satisfeita.</i>
O que você acha que esse programa faz?	<i>Espera-se que o participante infira que por mais que tenha sido construído de forma diferente, o programa, assim como o anterior, solicita um número ao usuário e retorna a mensagem "O numero digitado e menor que 10", caso o número digitado seja menor que 10, ou a mensagem "O numero digitado nao e menor que 10", caso o número digitado seja maior ou igual a 10.</i>

Fonte: elaborado pelo autor (2024).

Após a discussão dos dois programas apresentados, espera-se que os participantes infiram que é possível utilizar o comando `senao` como complemento à

instrução “se entao”, determinando uma ação a ser executada caso a condição inicial não seja satisfeita. Nesse momento, pretende-se reforçar que a instrução “se entao senao” segue a estrutura:

se (*proposição cuja veracidade será avaliada*) **entao**
 ação a ser realizada se a proposição for verdadeira
senao
 ação a ser realizada se a proposição for falsa
fim_se

Será reforçado que na estrutura observada, a proposição cuja veracidade será avaliada é representada entre parênteses entre os comandos “se” e “entao”, e a ação que será realizada se a proposição for verdadeira continua aparecendo logo após o comando “entao”, como visto anteriormente. No entanto, antes da utilização do comando “fim_se”, que finalizaria o bloco de tomada de decisão, é possível utilizar o comando “senao”, que permite determinar uma ação a ser realizada caso a proposição analisada seja falsa.

Será colocado, ainda, que podemos construir programas com funcionamentos similares a partir de códigos diferentes. Nos programas apresentados, por exemplo, embora tenham sido utilizadas estruturas diferentes nos códigos, são retornados os mesmos dados de saída para um dado valor digitado pelo usuário. Desse modo, um mesmo problema pode ser solucionado de diferentes modos, podendo-se recorrer a estratégias variadas para se chegar à solução. Nesse contexto, como exercício, será solicitado aos participantes que façam as alterações necessárias no último código apresentado de modo que os dados de saída sejam os representados no Quadro 27.

Quadro 27: Funcionamento do novo programa e produção esperada dos participantes.

Ações a serem realizadas pelo programa	Produção esperada dos participantes
<p>Pedir que o usuário digite um número. Se o número for menor ou igual a 100, deve aparecer na tela “O numero nao e maior que 100”. Se o número for maior que 100, deve aparecer na tela “O numero e maior que 100”</p>	<pre> variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numero nao e maior que 100" senao escrever "O numero e maior que 100" fim_se fim ou variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numero nao e maior que 100" fim_se se (a>100) escrever "O numero e maior que 100" fim_se fim </pre>

Fonte: elaborado pelo autor (2024).

Após a construção do código, será reforçado junto aos participantes que o programa poderia ter sido estruturado tanto utilizando a instrução “se entao senao”, como usando-se somente os comandos “se” e “entao”. Por fim, o terceiro momento do encontro corresponde à avaliação, que consiste na construção de um programa que deve realizar ações diferentes de acordo com a entrada fornecida pelo usuário (Figura 40).

Figura 40: Momento 3 do encontro 5.

<p>Momento 3 (avaliação): construção de um programa onde pode ser utilizada a instrução senao Duração: 20 minutos</p> <p>Nesse momento, os estudantes serão estimulados a construírem um programa que resolva à seguinte situação problema:</p> <p>Uma companhia telefônica realiza a seguinte promoção para seus clientes:</p> <ul style="list-style-type: none"> - Se o cliente utilizar até 40MB é cobrado um valor fixo de R\$50, além de R\$2 por megabite utilizado. - Se o cliente utilizar mais que 40MB, segue sendo cobrado R\$2 por cada megabite utilizado, mas o valor fixo cai pra R\$40. <p>Construa um programa que solicite ao usuário a quantidade de megabites utilizada no mês e retorne o valor a ser pago pelo cliente.</p>
--

Fonte: elaborado pelo autor (2024).

Quadro 28: Avaliação do encontro 5 e produção esperada dos participantes.

Avaliação	Produção esperada dos participantes
<p>Uma companhia telefônica realiza a seguinte promoção para seus clientes:</p> <ul style="list-style-type: none"> - Se o cliente utilizar até 40MB é cobrado um valor fixo de R\$50, além de R\$2 por megabite utilizado. - Se o cliente utilizar mais que 40MB, segue sendo cobrado R\$2 por cada megabite utilizado, mas o valor fixo cai pra R\$40. <p>Construa um programa que solicite ao usuário a quantidade de megabites utilizada no mês e retorne o valor a ser pago pelo cliente.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de megabites" ler a se (a<=40) entao b=50+2*a senao b=40+2*a fim_se escrever "O valor a ser pago e: R\$", b fim ou variaveis numerico a, b inicio escrever "Digite a quantidade de megabites" ler a se (a<=40) entao b=50+2*a fim_se se (a>40) entao b=40+2*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>

Fonte: elaborado pelo autor (2024).

Terminada a construção e discussão do programa elaborado, finaliza-se a etapa de desenvolvimento de conceitos e aprendizagem de novos comandos. Os dois últimos

encontros consistem em momentos avaliativos, nos quais os participantes deverão construir programas para resolver problemas específicos.

Encontro 6

No sexto encontro, primeiro dos dois momentos finais de avaliação, os participantes deverão construir programas com o objetivo de solucionar duas situações problema específicas. Os programas devem ser construídos desde o princípio – ou seja, não haverá nenhuma linha de código previamente elaborada.

Figura 41: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 6.

<u>ENCONTRO 6</u>	
Avaliação (com possibilidade de consulta)	
Objetivos:	<ul style="list-style-type: none"> • Resolver situações problema que se relacionam com a função afim através da construção de programas no ILA
Habilidades do pensamento computacional priorizadas:	<ul style="list-style-type: none"> • Abstração, decomposição, reconhecimento de padrões, algoritmos
Recursos:	<ul style="list-style-type: none"> • Computadores com ILA e NVDA instalados
Avaliação:	<ul style="list-style-type: none"> • Construção de programas de computador com o objetivo de resolver duas situações problema

Fonte: elaborado pelo autor (2024).

Inicialmente, os estudantes serão informados que terão 1h para a elaboração dos códigos e que poderão recorrer a um material para consulta, que consiste em um conjunto de códigos que foram trabalhados em encontros anteriores. A organização do material de consulta pode ser verificada nos Anexos deste trabalho.

Em seguida, os participantes serão informados que podem iniciar a construção dos códigos. Nos encontros avaliativos não haverá mediação do pesquisador, de modo que os participantes deverão construir o código de maneira completamente autônoma. Caso

sejam realizadas perguntas relacionadas ao código em si, será colocado que não é possível respondê-las.

Figura 42: Momento 1 do encontro 6.

Momento 1: apresentação da dinâmica de avaliação

Duração: 5 minutos

Neste momento, será apresentada aos participantes a dinâmica da atividade. Será colocado que os participantes terão 1h para a construção de dois programas de computador, com o objetivo de modelar duas situações problema apresentadas. Os participantes poderão consultar um arquivo auxiliar que contém todos os programas trabalhados nas sequências didáticas anteriores.

Fonte: elaborado pelo autor (2024).

Figura 43: Momento 2 do encontro 6.

Momento 2: construção dos programas

Duração: 1h

Nesse momento os participantes deverão construir programas que modelem as seguintes situações problema:

Situação 1

Nos últimos anos, aplicativos para transportes de passageiros em automóveis privados têm se multiplicado, oferecendo serviços voltados tanto para o público geral, quanto para grupos específicos. Ainda assim, em virtude de problemas como excesso de cancelamento de viagens e aumento de tarifas, os táxis por vezes continuam representando uma opção atrativa no transporte de passageiros. Na cidade de Taxilandia, por exemplo, a corrida de táxi é calculada da seguinte maneira:

- É cobrada a taxa de R\$6 referente à bandeirada (valor fixo inicial de qualquer corrida), independente do horário em que seja realizada a corrida;

- É cobrado o valor adicional de R\$3 por quilômetro rodado.

Nessas condições, construa um programa que determine o valor a ser pago pelo usuário de acordo com a quantidade de quilômetros rodados.

Situação 2

A comissão é um benefício muitas vezes utilizado no setor comercial para motivar os empregados e, conseqüentemente, aumentar os lucros da empresa. Tulio é proprietário de uma loja de tapetes, e para tentar impulsionar as vendas no período de Natal, decidiu conceder, além das comissões, um bônus aos funcionários que atingissem a meta estabelecida para a quantidade de tapetes vendidos. Assim, a renda mensal dos vendedores de sua loja passou a ser composta por uma parte fixa de R\$1500, e uma parte variável que dependia da quantidade de tapetes que o vendedor conseguisse vender. Para o cálculo da parte variável, ficou estabelecido que cada vendedor ganharia uma comissão de R\$ 16 por tapete vendido e, além disso, se o vendedor conseguisse vender mais de 100 tapetes no mês, ganharia adicionalmente um bônus de R\$300.

Nessas condições, construa um programa que Tulio poderia utilizar para calcular o salário dos vendedores da loja de acordo com a quantidade de tapetes vendidos.

Fonte: elaborado pelo autor (2024).

Quadro 29: Situações problema e produção esperada dos participantes.

Situação problema	Produção esperada dos participantes
Situação 1 (Figura 43)	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de quilometros" ler a b=6+3*a escrever "O valor total e: R\$", b fim </pre>
Situação 2 (Figura 43)	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de tapetes" ler a se (a>100) entao b=1800+16*a senao b=1500+16*a fim_se escrever "O valor a ser pago e: R\$", b fim ou variaveis numerico a, b inicio escrever "Digite a quantidade de tapetes" ler a se (a>100) entao b=1800+16*a fim_se se (a<=100) entao b=1500+16*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>

Fonte: elaborado pelo autor (2024).

Encontro 7

No sétimo encontro, último da sequência didática, os participantes também deverão construir programas com o objetivo de solucionar duas situações problema específicas. No entanto, neste caso não será possível recorrer a nenhum material de consulta. Assim, os programas devem ser construídos desde o princípio, não havendo nenhum trecho de código a ser utilizado como referência.

Figura 44: Objetivos, habilidades do pensamento computacional priorizadas, recursos e formato da avaliação no encontro 7.

<u>ENCONTRO 7</u>	
Avaliação (sem possibilidade de consulta)	
Objetivos:	<ul style="list-style-type: none"> • Resolver situações problema que se relacionam com a função afim através da construção de programas no ILA
Habilidades do pensamento computacional priorizadas:	<ul style="list-style-type: none"> • Abstração, decomposição, reconhecimento de padrões, algoritmos
Recursos:	<ul style="list-style-type: none"> • Computadores com ILA e NVDA instalados
Avaliação:	<ul style="list-style-type: none"> • Construção de programas de computador com o objetivo de resolver duas situações problema

Fonte: elaborado pelo autor (2024).

Inicialmente, os estudantes serão informados que terão 1h para a elaboração dos códigos e que, desta vez, não poderão recorrer a nenhum material para consulta.

Figura 45: Momento 1 do encontro 7.

<p>Momento 1: apresentação da dinâmica de avaliação Duração: 5 minutos</p> <p>Neste momento, será apresentada aos participantes a dinâmica da atividade. Será colocado que os participantes terão 1h para a construção de dois programas de computador, com o objetivo de modelar duas situações problema apresentadas. Os participantes não terão nenhuma fonte de consulta.</p>
--

Fonte: elaborado pelo autor (2024).

Em seguida, os participantes serão informados que podem iniciar a construção dos códigos. Assim como no encontro avaliativo anterior, não haverá mediação do pesquisador, de modo que os participantes deverão construir o código de maneira completamente autônoma. Caso sejam realizadas perguntas relacionadas ao código em si, será colocado que não é possível respondê-las.

Figura 46: Momento 2 do encontro 7.**Momento 2: construção dos programas****Duração: 1h**

Nesse momento os participantes deverão construir programas que modelem as seguintes situações problema:

Situação 3

Mariana ganhou de sua mãe como presente de Natal um cofre que já continha uma nota de R\$200. Animada com o presente, decidiu, a partir do mês seguinte, depositar R\$35 no cofre a cada mês. Construa um programa que determine quantos reais haverá no cofre de acordo com a quantidade de meses passados.

Situação 4

Academias de ginástica e musculação têm se espalhado por todo o país, passando a fazer parte do cotidiano de muitos brasileiros. No entanto, é comum encontrar pessoas que inicialmente se engajam na realização de atividades físicas mas, com o passar o tempo, passam a se sentir desmotivadas e abandonam a prática. Pensando em atrair novos clientes e estimular a realização de exercícios físicos entre eles, uma academia lançou a seguinte promoção:

- Se o cliente frequentar a academia menos que 10 horas no mês, deve pagar apenas um valor fixo de R\$80.

- Se o cliente frequentar a academia 10 ou mais horas no mês, o valor fixo cai para R\$30, e é cobrado um adicional de R\$2 por cada hora utilizada na academia.

Construa um programa que determine o valor a ser pago pelo cliente de acordo com a quantidade de horas frequentadas na academia.

Fonte: elaborado pelo autor (2024).

Quadro 30: Situações problema e produção esperada dos participantes

Situação problema	Produção esperada dos participantes
Situação 3 (Figura 46)	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de meses" ler a b=200+35*a escrever "O valor a ser pago e: R\$", b fim </pre>
Situação 4 (Figura 46)	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de horas" ler a se (a<10) entao b=80 senao b=30+2*a fim_se escrever "O valor a ser pago e: R\$", b fim ou variaveis numerico a, b inicio escrever "Digite a quantidade de tapetes" ler a se (a<10) entao b=80 fim_se se (a>=10) entao b=30+2*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>

Fonte: elaborado pelo autor (2024).

6.1.3. EXPERIMENTAÇÃO

A fase de experimentação é o momento em que a sequência didática é de fato conduzida junto aos estudantes, e são coletados os dados para a análise a posteriori. Neste trabalho, a sequência didática foi aplicada junto a quatro grupos de participantes, entre 24 de setembro de 2022 e 1 de setembro de 2023. Os quatro grupos mencionados foram formados de acordo com as disponibilidades dos estudantes e do pesquisador, e tinham a seguinte composição: grupo 1: Adriana (campus A); grupo 2: Marcos (campus A); grupo 3: Sara e Renata (campus B); e grupo 4: Ana, Fabio e Helena (campus C).

No caso de todos os grupos, optou-se pela realização de um encontro inicial (Encontro 0), no qual a pesquisa foi apresentada aos participantes, bem como foram mencionadas algumas características do ILA. A partir de então, os encontros passaram a ocorrer semanalmente com cada grupo, sempre em um mesmo dia da semana previamente acordado com os participantes do grupo. As transcrições dos áudios dos encontros podem ser acessadas integralmente em https://osf.io/jpgts/?view_only=bc33dcf7ff8649508a27208b1c2ae767.

Cabe ressaltar que no decorrer da sequência, uma série de fatores fizeram com que alguns encontros não pudessem ocorrer no dia previsto inicialmente. O fato de o período de experimentação ter coincido com o período da Copa do Mundo de 2022, por exemplo, demandou o adiamento de alguns encontros que ocorreriam em dias de jogos da seleção brasileira (quando, em geral, era decretado ponto facultativo).

Outro fator que teve impacto, nesse sentido, foi o fato de alguns participantes não terem podido comparecer, por algum motivo, em determinados dias. Alguns estudantes, por exemplo, precisaram se ausentar algumas semanas para participar de práticas de laboratório, ou para estudar para avaliações. Outros, em algumas ocasiões, eram impossibilitados de comparecerem à escola por questões de segurança pública nas localidades de suas residências. Cabe destacar, ainda, o caso do participante Marcos, que por ser atleta precisou em diversos momentos se ausentar para participar de treinamentos, eventos esportivos, ou sessões de fisioterapia. Tal fato fez com que o período de experimentação se estendesse em relação ao previsto inicialmente.

Quadro 31: Datas de realização dos encontros com cada grupo de participantes, e frequência dos estudantes em cada encontro (P: presente; A: ausente).

CAMPUS	PARTICIPANTE	Encontro 0		Encontro 1		Encontro 2		Encontro 3		Encontro 4		Encontro 5		Encontro 6		Encontro 7	
		Data	Freq.														
A	Adriana	22-set-2022	P	10-out-2022	P	17-out-2022	P	7-nov-2022	P	21-dez-2022	P	21-dez-2022	P	21-dez-2022	P	6-jan-2023	P
A	Marcos	4-out-2022	P	4-out-2022	P	8-nov-2022	P	8-nov-2022	P	13-dez-2022	P	13-dez-2022	P	1-set-2023	P	1-set-2023	P
B	Sara	26-set-2022	p	3-out-2022	p	10-out-2022	A	17-out-2022	p	7-nov-2022	A	12-dez-2022	A	19-dez-2022	A	13-fev-2023	A
	Renata		P		P		P		P		P		P				
C	Ana	24-set-2022	P	1-out-2022	P	7-out-2022	P	14-out-2022	P	4-nov-2022	A	11-nov-2022	A	16-dez-2022	A	3-mar-2023	A
	Fabio		P		A		A		A								
	Helena		p		p		p		p		p		p				

Fonte: elaborado pelo autor (2024).

No Quadro 31 são exibidos, de maneira geral, os dias em que cada um dos encontros ocorreu. Além disso, também são demarcadas as presenças e ausências de participantes a cada encontro. Cabe colocar, ainda, que ao fim do encontro 7 ocorreu uma entrevista final com os participantes presentes naquele momento.

Além da extensão do período de condução da sequência didática, outro aspecto que teve impacto no processo de experimentação foi o fato que alguns estudantes, a partir de algum momento, deixaram de comparecer aos encontros. Foi o caso dos participantes Fabio, Ana e Sara. No Quadro 31 se nota até que ponto da sequência cada um deles esteve presente. Cabe ressaltar que nenhum dos participantes mencionados registrou formalmente o interesse de interromper a participação na pesquisa.

Por fim, durante toda a fase de experimentação, buscou-se conduzir os encontros de acordo com o previsto na fase de concepção. No entanto, em algumas situações notava-se, na prática, a inadequação de alguma proposta planejada. Conforme destacam Almouloud e Coutinho (2008), este não é um problema no processo de Engenharia Didática, na medida em que é possível realizar correções que sejam necessárias com base em conclusões advindas do desenvolvimento experimental. Com isso, retorna-se à análise a priori em um processo de constante complementação. Em alguns momentos, por exemplo, notava-se que, por conta de limitações temporais, seria inviável que alguma atividade inicialmente prevista fosse de fato colocada em prática. Desse modo, tal atividade era retirada do planejamento da sequência, tampouco sendo realizada nos outros grupos.

6.1.4. ANÁLISE A POSTERIORI

A análise a posteriori da sequência didática é o momento em que os resultados obtidos na experimentação são analisados e confrontados com aquilo que se esperava no momento de concepção da sequência. Desse modo, é feita uma comparação com os comportamentos previstos inicialmente na pesquisa, registrados na análise a priori da sequência. A seguir, são apresentadas as análises referentes a cada um dos encontros da sequência.

Encontro 1

No primeiro encontro da sequência didática realizou-se um diálogo inicial em relação ao processo de programação de computadores, às linguagens de programação e, mais especificamente, ao ILA e, além disso, foram desenvolvidas noções básicas de entrada, processamento e saída de dados. No momento inicial de diálogo, como já havia sido previsto na análise a priori, observou-se que a maior parte dos participantes não havia experiências prévias com programação de computadores. Questionados sobre o que seria um programa de computador, a maior parte respondeu de forma confusa, de modo que nenhum dos estudantes apresentou uma definição clara em relação ao que foi perguntado. Os trechos a seguir ilustram algumas das respostas dos participantes.

Adriana: Um programa de computador, ele é composto de várias partes. O software em si, que ele serve para facilitar ou explicar pra pessoa, ou facilitar, ou reproduzir uma... algo que o ser humano demoraria mais tempo fazendo.

Marcos: Um programa de computador é um arquivo que passou por algumas modificações para servir a algum... para algumas coisas... alguns mecanismos, algumas contas, enfim...

Sara: Acho que é como um sistema, que tem várias coisas incluídas em uma só, que possa organizar um único objetivo.

Renata: Tipo alguma coisa correlacionada a um script?

Fabio: Ah, seria um recurso... é... como eu posso dizer... que pode variar de diversas maneiras (...) Porque ele pode ter... esse recurso pode ter várias finalidades.

Ana: Menor ideia.

Já no segundo momento do encontro, objetivou-se discutir as ideias de entrada, processamento e saída de dados. Aqui, o pesquisador realizou, em todos os grupos, uma breve explanação sobre as diferenças entre operações de inserção de dados pelo usuário (entrada), processamento de dados pelo computador (processamento) e fornecimento de dados processados ao usuário (saída). Além disso, foram apresentados os comandos “ler” e “escrever”, utilizados para instruções de entrada e saída de dados no ILA, respectivamente.

No terceiro momento do encontro, com o objetivo de ilustrar o funcionamento dos comandos de entrada e saída de dados, os participantes foram convidados a fazerem a leitura de dois códigos construídos utilizando a pseudolinguagem ILA. Nos Quadros Quadro 32 e Quadro 33 são comparados os comportamentos esperados na análise a priori com aqueles de fato observados no momento da experimentação.

O Quadro 32 revela que, em um primeiro momento, apenas os participantes Adriana e Marcos compreenderam que no programa apresentado há somente comandos de saída de dados. Já em relação ao funcionamento do programa, Adriana e Helena responderam corretamente. Por outro lado, Marcos afirma corretamente que irá aparecer uma saída sem, no entanto, explicitar como ela seria. Além disso, o estudante acredita que poderia ocorrer algum erro no processamento do programa pelo fato de não existirem comandos de entrada no código. Sara, por sua vez, fornece uma resposta confusa, sugerindo falta de clareza em relação ao significado do programa. Por fim, Renata, Ana e Fabio não responderam ao questionamento.

Quadro 32: Análise a posteriori do momento 3 do encontro 1 (primeiro programa).

Pergunta prevista	Ação esperada	Ação observada
Esse programa tem dados de entrada, de saída ou os dois?	<i>Espera-se que o participante compreenda que o programa tem apenas dados de saída.</i>	<p>Adriana: Tem atividades de... peraí... de saída, eu acho que é saída de dados.</p> <p>Marcos: “Pô”, cara... ali ele não tem... não tá programado com nenhuma delas (...) O que eu vi ali só tem... meio que “tipo assim”, só saída né...</p> <p>Sara: Ele tem atividade de entrada e de saída.</p> <p>Renata: Eu acho que tem as duas coisas. (...) É que eu fiquei pensando que talvez teria só entrada, porque só tem uma frase ali, mas tem início e fim.</p> <p>Ana: A última opção.</p> <p>Helena: Entrada?</p> <p>Fabio: De entrada.</p>
O que você acredita que acontecerá nesse programa?	<i>Espera-se que o participante compreenda que será exibida a frase “Eu sou estudante do Colégio Pedro II” na tela do computador.</i>	<p>Adriana: Vai escrever eu sou um aluno do Colégio II pro usuário.</p> <p>Marcos: Bom, só vai aparecer a saída e provavelmente pode ser que dê algum erro nele por causa que ele não tem entrada e nem o meio lá do processador.</p> <p>Sara: Acho que... é como se... escrever, mas a gente tivesse que escrever para o comando, e esse comando iria algo por nós...</p> <p>Helena: Vai aparecer a frase que tá escrita.</p>

Fonte: elaborado pelo autor (2024).

Já a análise do Quadro 33 revela que todos os participantes compreenderam que o segundo programa apresenta comandos de entrada e saída. A diferença em relação ao ocorrido no primeiro programa, quando apenas uma minoria de participantes identificou a existência exclusiva de comandos de saída, pode se dever à devolutiva dada pelo pesquisador após os erros cometidos.

Por outro lado, no que se refere às perguntas relacionadas à declaração de variáveis e de seus tipos, a maior parte das respostas dos estudantes não correspondeu ao esperado na análise a priori. Em primeiro lugar, apenas Adriana mencionou que nas duas primeiras linhas do programa é realizada a declaração de variáveis. Marcos, por sua vez, parece argumentar, de maneira confusa, que nas linhas mencionadas é informada a “variante” que será utilizada, o “nome”. Já Sara coloca que é caracterizado o que será escrito posteriormente, sem mencionar a declaração de variáveis. Por fim, as respostas dos outros participantes sugerem que o questionamento pode não ter sido compreendido como esperado pelo pesquisador, na medida em que os estudantes apenas apontam os comandos que são observados nas primeiras linhas, sem discorrer sobre seus significados.

Já questionados sobre o termo “character” na segunda linha do código, mais uma vez apenas Adriana responde claramente que a palavra se refere ao tipo da variável que será utilizada. Renata e Helena, por outro lado, parecem querer argumentar que o termo character está associado à utilização de letras, possivelmente em uma menção ao caráter alfanumérico desse tipo de variável. Já os outros participantes não respondem ou dão respostas que divergem do esperado na análise a priori.

Em relação ao termo “nome” na segunda linha do código, novamente apenas Adriana responde de maneira correta que se trata da variável utilizada no programa. Renata, por sua vez, embora não tenha mencionado que o termo representa a variável usada, parece ter compreendido que se trata de um símbolo utilizado para o reconhecimento, pelo computador, do que for digitado pelo usuário (*É pra ele reconhecer as letras que a gente escrever?*). Por outro lado, Marcos e Helena mencionam que o termo estaria associado ao “nome de uma pessoa” ou à “aparição de um nome”, sem esclarecer, no entanto, que se trata da variável em si. Já os outros participantes não responderam ao questionamento.

No que se refere ao símbolo que aparece após o termo “escrever”, apenas dois participantes identificaram o uso das aspas, ao passo que outros dois mencionam a

palavra ou a frase que se observa após as aspas (“digite” ou “digite seu nome”). Tal fato pode ser explicado pelo fato de o NVDA, leitor de telas utilizado pelos participantes, não fazer a leitura imediata de alguns símbolos, entre eles as aspas. Desse modo, para que o usuário do recurso consiga identificar esses símbolos, é necessário fazer uma leitura por carácter utilizando as setas do teclado, ao invés da leitura direta da linha de código.

Por fim, questionados em relação ao funcionamento do programa, embora a maior parte dos participantes tenha mencionado que seria solicitado e/ou informado o nome do usuário, nenhum respondeu de forma clara e completa como, de fato, funcionava o programa. Assim, foi necessária a mediação do pesquisador de modo que os estudantes compreendessem que inicialmente seria exibida uma mensagem solicitando o nome do usuário e, após a inserção deste dado, seria retornada uma frase contendo o nome digitado.

Quadro 33: Análise a posteriori do momento 3 do encontro 1 (segundo programa).

Pergunta prevista	Ação esperada	
<p>Esse programa tem dados de entrada, de saída ou as duas?</p>	<p><i>Espera-se que o participante compreenda que o programa tem dados de entrada e de saída.</i></p>	<p>Adriana: Tem entrada e saída. “Escrever” e “ler”.</p> <p>Marcos: Tem saída. Tem entrada. Tem saída. Tá, ok?</p> <p>Sara: As duas coisas.</p> <p>Renata: É que teve uma parte ali que eu não vi início, mas... eu acho que é só... não sei, só o fim. (...) Ah, tá. Tem, tem, tem. (...) As duas.</p> <p>Helena: Os dois.</p> <p>Ana: Os dois.</p> <p>Fabio: Os dois.</p>
<p>O que acontece nas duas primeiras linhas do programa?</p>	<p><i>Espera-se que o participante compreenda que nas duas primeiras linhas é realizada a declaração das variáveis que serão utilizadas no programa.</i></p>	<p>Adriana: Eh, apresentação de uma variável, declaração da variável.</p> <p>Marcos: Tem a variável, e tem a... o nome da... da carácter, que é o... que é a variante que você vai usar... que é o nome... o nome? eu acho que é (...) É o nome.</p> <p>Sara: É... nas duas primeiras linhas ele está caracterizando o que vai escrever embaixo.</p> <p>Renata: Tá escrito carácter nome.</p> <p>Helena: Variáveis [risos]... carácter nome...</p>

		<p>Fabio: Variáveis (...) Caracter nome...</p> <p>Ana: Ah, variáveis!</p>
Por que foi utilizado o termo caracter na segunda linha?	<p><i>Espera-se que o participante compreenda que o termo caracter foi utilizado para definir o tipo da variável declarada.</i></p>	<p>Adriana: Porque é o tipo da variável.</p> <p>Marcos: Porque vai ter... pra escrever sobre alguma coisa.</p> <p>Renata: Ah, que é letra?</p> <p>Helena: Isso aí é pra dizer a forma que vai ser feito o programa... tem letra...</p> <p>Fabio: Quais as características que aquele... que aquele programa vai obter.</p>
E o termo nome da segunda linha, o que representa no programa?	<p><i>Espera-se que o participante compreenda que o termo nome representa a nomenclatura utilizada para a variável.</i></p>	<p>Adriana: A variável.</p> <p>Marcos: Vai ser o nome de alguma pessoa, no caso?</p> <p>Renata: É pra ele reconhecer as letras que a gente escrever?</p> <p>Helena: Que vai aparecer o nome? Ou é o comando que vai aparecer escrito? Essa aí eu não sei não.</p>
Tanto nesse quanto no outro programa, que símbolo veio logo após a palavra escrever?	<p><i>Espera-se que o participante identifique que após o comando escrever, utiliza-se as aspas para delimitar a frase ou conjunto de símbolos que serão exibidos como saída.</i></p>	<p>Adriana: Aspas.</p> <p>Marcos: É o “digite”?</p> <p>Renata: Aspas?</p> <p>Fabio: Escrever digite seu nome...</p>
O que você acredita que acontecerá nesse programa?	<p><i>Espera-se que o participante compreenda que o nome do usuário será solicitado na tela do computador e, em seguida, será exibida a frase “[Nome] e estudante do Colégio Pedro II”, onde [Nome] representa o que foi digitado pelo usuário.</i></p>	<p>Adriana: Eh, ele vai perguntar pro... o nome da pessoa na tela, vai aparecer essa informação, a pessoa vai digitar, e depois vai aparecer a informação do nome da pessoa.</p> <p>Marcos: Cara, não sei...</p> <p>Renata: Eu acho que vai aparecer pra gente digitar o nosso nome, e aí depois ele vai ler o nosso nome, e aí depois...</p> <p>Sara: Talvez, eu não sei mas, vai aparecer de uma forma mais legível na tela (...) o “digite seu nome”. (...) Mas vai fazer todas as funções assim, de ler e aparecer novamente a frase. (...) Digite seu nome.</p> <p>Helena: Ele vai aparecer pra você escrever seu nome?</p> <p>Fabio: Vai aparecer seu nome escrito.</p>

Fonte: elaborado pelo autor (2024).

Finalmente, o último momento do encontro foi destinado a uma atividade de avaliação, na qual os participantes deveriam fazer as alterações necessárias em um código de modo que, após a inserção de um dado pelo usuário, fosse exibida uma determinada frase na tela do computador. No Quadro 34 são apresentados os códigos construídos pelos participantes, em comparação com aqueles esperados na análise a priori.

Quadro 34: Análise a posteriori da avaliação do encontro 1.

Avaliação	Produção esperada dos participantes
<p>Construir um programa que solicite a idade do usuário e retorne a frase "A idade do usuario e 'tanto'", onde 'tanto' é o valor digitado pelo usuário.</p>	<pre>variaveis numerico a inicio escrever "Digite a sua idade: " ler a escrever "A idade do usuário e: ", a fim</pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre>variaveis numerico idade inicio escrever "digite sua idade" ler idade escrever " Voce tem", idade fim</pre>	<p style="text-align: center;">Marcos</p> <pre>variaveis caracter idade inicio escrever "Digite sua idade" ler idade escrever "minha idade e ", idade fim</pre>
<p style="text-align: center;">Renata</p> <pre>variaveis numerico numero inicio escrever "Digite sua idade" ler numero escrever "A idade do usuario e ", numero fim</pre>	<p style="text-align: center;">Sara</p> <pre>variaveis numerico numero inicio escrever "Digite sua idade" ler numero escrever "A idade do usuario e ", numero fim</pre>
<p style="text-align: center;">Helena</p> <pre>variaveis numerico idade inicio escrever "sua idade" ler idade escrever "sua idade e", idade fim</pre>	<p style="text-align: center;">Fabio</p> <pre>variaveis numerico idade inicio escrever "sua idade é" escrever "digite a idade" fim</pre>
<p style="text-align: center;">Ana</p> <p style="text-align: center;">Não realizou a atividade</p>	

Fonte: elaborado pelo autor (2024).

O quadro mostra que, de formas diferentes, Adriana, Marcos, Renata, Sara e Helena construíram programas que atendiam ao que foi solicitado. Ressalta-se, aqui, que as participantes Renata e Sara apresentaram o mesmo código pois, nesta atividade, trabalharam colaborativamente. Cabe destacar, ainda, que Marcos, ao contrário das outras participantes mencionadas, utiliza o tipo caracter para a variável idade, o que no caso deste programa não representa um problema na medida em que a idade é simplesmente apresentada novamente em uma frase, não sendo realizada nenhuma operação aritmética com o valor.

Além disso, pode-se notar que alguns dos participantes cometem pequenos equívocos que não impedem o programa de funcionar, mas que poderiam ser evitados. Entre os problemas identificados estão a não utilização de letra maiúscula no início da frase (“*digite sua idade*”, “*sua idade*”, “*sua idade e*”), o uso excessivo ou indevido de espaços (“*Digite sua idade*”, “*Voce tem*”), e a escolha inadequada de frases para solicitação ou fornecimento de informações. Helena, por exemplo, não utiliza um verbo (digite, insira, escreva) na frase “*sua idade*”, o que poderia ocasionar dúvida para o usuário. Já na construção de Adriana, o uso da frase “*Voce tem*”, na penúltima linha de código, faria com que a saída fosse visualizada no formato “*Voce tem[idade]*” (Voce tem15, como exemplo). Ainda assim, como mencionado anteriormente, tais pontos não representam impeditivos ao funcionamento do programa.

Por outro lado, dois participantes não atenderam ao que foi solicitado para a atividade. Em primeiro lugar, embora Fabio tenha construído um código que não contém erros sintáticos, seu programa não apresenta a funcionalidade solicitada. Na medida em que o participante não utiliza um comando de entrada em nenhum momento, seu programa apenas exibe duas frases na tela do computador. Por fim, Ana iniciou a atividade mas, apresentando dificuldades e inquietação ao longo de sua realização, não quis dar continuidade, mesmo após incentivo do pesquisador.

Encontro 2

Os objetivos do segundo encontro da sequência didática se centraram no reconhecimento e utilização de operadores aritméticos no ILA, bem como na análise e construção de programas onde tais símbolos fossem utilizados. Nesse cenário, no primeiro momento do encontro, após breve retomada dos conceitos desenvolvidos no encontro anterior, o pesquisador conduziu um diálogo sobre operações aritméticas e os

símbolos utilizados para a representação de tais operações. Mais especificamente, foram apresentados os operadores utilizados para a realização de operações de soma (+), subtração (-), multiplicação (*), divisão (/) e exponenciação (^) no ILA, bem como as localizações de tais símbolos no teclado do computador.

Já no segundo momento do encontro, o pesquisador solicitou que os participantes fizessem a leitura de um código construído para fazer uma operação de soma entre dois números e, em seguida, foi realizado um conjunto de perguntas aos estudantes. No Quadro 35 são exibidas as respostas obtidas e, também, as esperadas na análise a priori.

Quadro 35: Análise a posteriori da primeira parte do momento 2 do encontro 2.

Pergunta prevista	Ação esperada	Ação observada
Esse programa apresenta comandos de entrada, de saída ou os dois?	<i>Espera-se que o participante compreenda que o programa tem dados de entrada e de saída.</i>	<p>Adriana: As duas coisas.</p> <p>Marcos: As duas.</p> <p>Renata: Hum, eu acho que as duas coisas.</p> <p>Helena: Os dois.</p> <p>Ana: Os dois.</p>
O que isso significa?	<i>Espera-se que o participante compreenda que, pelo fato de existirem dados de entrada e de saída, serão solicitadas informações que deverão ser digitadas pelo usuário e serão exibidos dados processados na tela do computador.</i>	<p>Adriana: Que você depende de um usuário pra você mover esse programa, sem um usuário você não consegue fazer nada nesse programa porque só vai estar a variável aqui.</p> <p>Marcos: Que tem os dois comandos... o escrever... e o de ler.</p> <p>Renata: O de entrada vai colocar as coisas na tela do computador, e o de saída a pessoa tem que escrever as coisas.</p> <p>Helena: Que você... que você vai dar o comando e ele vai me devolver.</p>
O que significam as duas primeiras linhas do programa?	<i>Espera-se que o participante compreenda que nas duas primeiras linhas é realizada a declaração das variáveis que serão utilizadas no programa.</i>	<p>Adriana: São declarações de variáveis. São no caso a variável “a” e a variável “b”, que são numéricas.</p> <p>Marcos: Ele dá a acusação de variável, que tem... E depois fala que a variável tal é a soma. No caso, a variável que está sendo usada ali pra somar os dois números é a soma.</p>

		<p>Renata: Tá dando um comando dizendo que é pra somar “a” com “b”.</p> <p>Helena: Variáveis, tá dizendo qual é o tipo de variável.</p>
O que você acha que esse programa faz?	<i>Espera-se que o participante identifique que o programa solicita que o usuário digite dois números e realiza a operação de soma entre os mesmos, exibindo na tela o resultado da soma.</i>	<p>Adriana: Ele vai somar a mais b, igual ele tá falando aqui... (...) E vai dar a tal da soma.</p> <p>Renata: Vai parecer na tela pra somar... digitar o número... (...) E depois vai aparecer o resultado.</p> <p>Helena: Então, vai aparecer pra ele digitar o número (...) Aí você coloca o número (...) E aí depois ele vai aparecer o resultado. Não, ele vai aparecer escreva o outro número (...) Aí depois ele vai aparecer o resultado.</p>
O que você acha que acontece na linha 8 do programa?	<i>Espera-se que o participante infira que na linha 8 atribui-se à variável soma o resultado da operação de soma entre os dois números digitados pelo usuário.</i>	<p>Adriana: Tá pegando as informações da variável “a” e da variável “b”, e somando as duas pra dar a variável “soma”.</p> <p>Marcos: Soma é igual a “a” mais “b”, no caso ele está somando.</p> <p>Renata: Tá somando “a” mais “b”.</p> <p>Helena: Hum... ali é... ele tá, os números que a gente colocou... (...) Vai dar o resultado mais b. É isso?</p>
O que você acha que acontece na linha 9 do programa?	<i>Espera-se que o participante compreenda que na linha 9 determina-se que seja exibida na tela do computador a frase “O resultado equivale a ‘soma’”, onde ‘soma’ representa o resultado da adição entre os números digitados pelo usuário.</i>	<p>Adriana: Vai aparecer o resultado na tela.</p> <p>Marcos: No caso ele está executando aqui um programa... um comando de saída, que fala que o resultado é igual a... o resultado equivale... é igual à soma, eu acho... o resultado equivale... entre aspas tá a... não, fora das aspas tem “a”, vírgula... dentro das aspas tá “soma”.</p> <p>Renata: Ah, tá! Ah, é pra você escrever o resultado. Não! É... é pra você, ele ler o resultado.</p> <p>Helena: Escrever o resultado equivale a soma (...) Ele vai mostrar o resultado equivalente à soma feita anteriormente.</p> <p>Ana: Escrever o resultado da soma.</p>

Fonte: elaborado pelo autor (2024).

O quadro mostra, inicialmente, que todos os participantes identificaram corretamente que no código apresentado se observam comandos de entrada e de saída. Já em relação ao significado desse fato, embora nenhum dos estudantes tenha dado uma resposta de acordo com o esperado, Adriana coloca corretamente que é necessária uma ação do usuário para que o programa possa funcionar adequadamente, e Helena argumenta que o usuário daria um “comando” para, em seguida, obter uma resposta do computador. Renata, por sua vez, troca os significados de entrada e saída, na medida em que responde que um comando de entrada seria usado para a exibição de elementos na tela do computador, e um de saída para o fornecimento de dados pelo usuário. Já Marcos menciona que são utilizados os comandos “escrever” e “ler” sem, no entanto, dar de fato um significado para a existência de tais comandos. A participante Ana, por fim, não respondeu ao questionamento.

Em seguida, perguntados em relação ao significado das duas primeiras linhas do código, nota-se que Adriana e Marcos mencionam, como esperado, a ocorrência da declaração de variáveis. Helena, por sua vez, coloca que é definido o tipo da variável. Aqui, cabe destacar que, embora Adriana não identifique em um primeiro momento a existência da variável “soma”, após mediação do pesquisador a participante chega a tal conclusão, como mostra o trecho a seguir:

Adriana: São declarações de variáveis. São no caso a variável “a” e a variável “b”, que são numéricas.

Pesquisador: Perfeito. Tem só essas duas ou tem alguma outra variável?

Adriana: Eu acho que... espera aí, eu acho... eu entendi só essas duas, calma aí... eu acho que tem mais uma sim. E soma!

Pesquisador: Perfeito. Então, quantas variáveis a gente tem no total?

Adriana: Três.

Ainda no que se refere às duas primeiras linhas do programa, Marcos chega à conclusão de que a variável soma é utilizada para somar os dois números representados pelas variáveis “a” e “b”. No entanto, a atribuição do resultado da operação $a + b$ à variável “soma” é realizada apenas posteriormente no código. Desse modo, pode-se inferir que Marcos chegou a essa conclusão a partir do significado da palavra soma, o que tem sentido em uma perspectiva lógica. De modo análogo, Renata chega a uma conclusão semelhante, na medida em que menciona que seria dado um comando para “somar ‘a’ com ‘b’”. No entanto, como o fato de a variável ter sido denominada “soma” não implica, necessariamente, que a ela será atribuído o resultado de uma soma, entende-se que este é um aspecto que merece atenção, na medida em que pode causar

confusão conceitual. A participante Adriana também menciona esse fato, como mostra o trecho a seguir:

Pesquisador: Então, o que que... o que o computador está fazendo nesse caso?

Adriana: Tá pegando as informações da variável “a” e da variável “b”, e somando as duas pra dar a variável “soma”.

Pesquisador: Perfeito.

Adriana: Deixa eu só te dar uma “ideia”, em vez de você botar “soma”, que vai ficar confuso, você poderia ter botado “c”.

Pesquisador: O “c”, né? Pois é.

Adriana: Porque fica meio confuso, fica parecendo que é uma função.

Em relação à funcionalidade do programa, embora não argumentem com clareza, Adriana, Renata e Helena parecem ter compreendido que será realizada uma soma de números digitados pelo usuário. Já no que se refere à linha 8 do programa ($soma = a + b$), Adriana argumenta corretamente que as informações das variáveis “a” e “b” são somadas “para dar” a variável “soma”. Renata e Marcos colocam que está ocorrendo uma soma entre “a” e “b”, sem, no entanto, mencionar a atribuição do resultado à variável “soma”. Helena, por sua vez, parece ter tido outro entendimento, na medida em que coloca que o resultado seria somado à variável “b”. Uma parte mais ampla do trecho em que a participante chega a tal conclusão é exposta a seguir:

Pesquisador: Perfeito. Na linha 8 do programa, vai lá na linha 8.

Helena: Soma é igual a mais b.

Pesquisador: O que vocês acham que tá acontecendo nessa linha?

Helena: Hum... ali é... ele tá, os números que a gente colocou...

Pesquisador: Uhum.

Helena: Vai dar o resultado mais b. É isso?

Analisando as falas da participante, uma possível explicação para a sua conclusão seria uma interpretação equivocada da transcrição realizada pelo leitor de telas. De fato, para o trecho “ $soma = a + b$ ” o NVDA retorna o áudio “soma igual a mais b”, ao invés de “soma igual a ‘a’ mais ‘b’”. Desse modo, a participante pode ter interpretado a linha 8 como “ $soma = +b$ ”, o que justificaria a resposta dada.

Por fim, em relação à linha 9 do programa (*escrever "O resultado equivale a: ", soma*), todos os participantes parecem compreender que se trata da exibição do resultado da soma na tela do computador, embora se expressem de maneiras diferentes. No que se seguiu, a partir das respostas dadas pelos participantes, buscou-se reforçar o significado dos comandos abordados no programa, com atenção especial para as

linhas 8 e 9 do código, nas quais o resultado da operação $a + b$ é atribuído à variável “soma, e é determinado o modo como os dados de saída são apresentados ao usuário. Além disso, com o objetivo de promover a reflexão dos participantes, foram realizadas duas perguntas acerca das alterações que deveriam ser feitas no código de modo que se pudesse fazer outras operações aritméticas. As respostas apresentadas são exibidas no Quadro 36.

Quadro 36: Análise a posteriori da segunda parte do momento 2 do encontro 2.

Pergunta prevista	Ação esperada	Ação observada
O que deveríamos modificar para criar um programa que multiplica dois números?	<i>Espera-se que o participante identifique que o sinal de adição na linha 8 do código deve ser alterado por um asterisco (*).</i>	<p>Adriana: O sinal. (...) É o asterisco.</p> <p>Marcos: O “mais” ali pra no caso ser o asterisco.</p> <p>Renata: O comando de somar, que tá ali escrito “somar”. (...) E aí eu teria que escrever... escrever “multiplicar”. (...) E o sinal também. (...) Asterisco.</p> <p>Helena: O sinal. Colocar o asterisco.</p>
O que deveríamos modificar para criar um programa que somasse três números?	<i>Espera-se que o participante identifique que deve ser declarada uma nova variável do tipo numérico, e que na linha 9 do código tal variável deve ser adicionada às duas já existentes.</i>	<p>Adriana: É... você teria que acrescentar mais uma variável. (...) Seria “c”. (...) E faria o resultado ser essa soma aqui.</p> <p>Marcos: Seria... tinha que colocar mais uma variável, c eu acho... e mudar pra $a+b+c$.</p> <p>Renata: Hum... teria que acrescentar mais asterisco... e colocar mais um... mais um caracter na... na linha (...) Que tá escrito... “a” mais “b”, no caso. (...) Aí teria que mudar e botar... por exemplo seria “a” vezes “b” vezes “c”.</p> <p>Helena: Ia ter que botar mais um terceiro número.</p>

Fonte: elaborado pelo autor (2024).

O Quadro 36 mostra que todos os participantes identificaram que para realizar uma operação de multiplicação entre dois números, bastaria trocar o símbolo da adição (+) pelo da multiplicação no ILA, o asterisco (*). Tal nível de acerto vai além do esperado na análise a priori, na medida em que havia sido prevista a possibilidade de dúvidas e equívocos na utilização do asterisco em lugar dos sinais habitualmente utilizados para a representação da multiplicação (. ou x).

Já para a criação de um programa que somasse três números, embora Adriana, Marcos e Helena tenham identificado que seria necessário adicionar mais um número à soma, apenas os dois primeiros mencionaram, explicitamente e de imediato, a necessidade de também declarar esta nova variável. Tal fato está de acordo com o esperado na análise a priori, pelo fato de os participantes possivelmente ainda não estarem familiarizados com o ato de declarar variáveis na construção de um código. Renata, por sua vez, aparentou não ter compreendido adequadamente a solicitação, apresentando modificações que deveriam ser feitas para, na realidade, multiplicar três números. Após mediação do pesquisador a participante chegou à conclusão correta, inclusive sobre a necessidade de declarar a nova variável no início do programa, conforme ilustra o trecho a seguir:

Renata: Hum... teria que acrescentar mais asterisco... e colocar mais um... mais um caracter na... na linha?

Pesquisador: Em qual linha?

Renata: Que tá escrito... “a” mais “b”, no caso.

Pesquisador: Aham.

Renata: Aí teria que mudar e botar... por exemplo seria “a” vezes “b” vezes “c”.

Pesquisador: Mas aí eu queria que ele somasse três números. Então seria...

Renata: Ah, então teria que ser “a” mais “b” mais “c”.

Pesquisador: Perfeito. Então você teria que mudar isso lá embaixo, né, na linha da soma, e aí você teria que fazer mais alguma coisa aqui em cima também. O que seria essa coisa que você teria que fazer aqui em cima?

Renata: Mudar aqui.

Pesquisador: Uhum.

Renata: E aí seria numerico a vírgula b, aí teria que botar vírgula c.

Por fim, no momento de avaliação do encontro, os participantes foram estimulados a criarem um programa que realizasse uma operação de potenciação. No Quadro 37 os programas construídos são confrontados com as produções esperadas na análise a priori.

Quadro 37: Análise a posteriori da avaliação do encontro 2.

Avaliação	Produção esperada dos participantes
<p>Construir um programa que solicite um número ao usuário e retorne o quadrado do número digitado</p>	<pre> variaveis numerico a, b inicio escrever "Digite o numero:" ler a b = a^2 escrever "O resultado equivale a: ", b fim ou variaveis numerico a, b inicio escrever "Digite o numero:" ler a b = a*a escrever "O resultado equivale a: ", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a, b inicio escrever "Digite o numero:" ler a b=a^2 escrever "O resultado equivale a: ", b fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, soma inicio escrever "Digite um numero:" ler a soma = a ^2 escrever "O resultado equivale a: ", soma fim </pre>
<p style="text-align: center;">Renata</p> <pre> variaveis numerico a, elevado inicio escrever "Digite o numero:" ler a elevado = a^2 escrever "O resultado equivale a: ", elevado fim </pre>	<p style="text-align: center;">Helena</p> <pre> variaveis numerico a, quadrado inicio escrever "digite o numero" ler a quadrado = a^2 escrever "O ressaltado equivale : ", quadrado fim </pre>
<p style="text-align: center;">Ana</p> <p style="text-align: center;">Não realizou a atividade</p>	

Fonte: elaborado pelo autor (2024).

A análise do Quadro 37 mostra que Adriana, Marcos, Renata e Helena construíram programas que atendiam ao solicitado. Pode-se notar que os quatro participantes optaram por utilizar o símbolo usado para a exponenciação (^), formando a expressão "a^2" em lugar da multiplicação da variável "a" por si própria (a*a). Além disso, um ponto notável é que cada participante utilizou um nome diferente para a variável que determinaria o resultado da exponenciação. Curiosamente, Marcos utilizou o termo

“soma” para se referir ao quadrado de um número, o que pode indicar que o participante ainda não tem total clareza em relação ao significado das variáveis em um programa.

Por fim, assim como no primeiro encontro, Ana iniciou a realização da atividade mas não a concluiu. Foi possível notar que a participante apresentou inquietação e uma série de dificuldades ao longo do encontro, o que pode ter contribuído para sua desmotivação.

Encontro 3

No terceiro encontro da sequência pretendia-se retomar o conceito de função afim e introduzir a noção de algoritmo junto aos participantes. Após relembrar o programa construído no encontro anterior e reforçar a ideia de função enquanto relação entre duas variáveis, o pesquisador passou à noção de algoritmo como um conjunto de passos para se alcançar um objetivo. A partir de uma analogia com a ideia de uma “receita de bolo”, pediu-se que os participantes construíssem um algoritmo que representasse uma “ida à escola”. Todos os estudantes atenderam de maneira satisfatória à solicitação.

Em seguida, com o objetivo de relacionar os conceitos de função afim e algoritmo, solicitou-se que os participantes elaborassem um algoritmo para a criação do código de um programa que solicitasse um número ao usuário e retornasse o seu dobro. Os algoritmos elaborados pelos participantes são apresentados no Quadro 38. Cabe colocar que a participante Sara não realizou a atividade por não tê-la compreendido com clareza, na medida em que a mesma tentou partir diretamente para a construção do código do programa.

A análise do quadro mostra que Adriana foi a participante que elaborou um conjunto de passos com maior detalhamento, sendo a estudante que mais se aproximou daquilo que era esperado na análise a priori. Ainda assim, todos os demais realizaram algoritmos que contemplavam a maior parte das etapas esperadas na produção dos estudantes.

Quadro 38: Análise a posteriori dos algoritmos elaborados pelos participantes.

Proposta	Produção esperada
<p>Construção de um algoritmo para a criação de um programa que solicite um número ao usuário e retorne o dobro dele.</p>	<ul style="list-style-type: none"> • <i>Solicitar o número ao usuário;</i> • <i>Armazenar o número;</i> • <i>Realizar a operação de dobrar o número, atribuindo o resultado a uma variável;</i> • <i>Exibir o valor do resultado na tela do computador.</i>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <p>1 o sistema pede um valor ao usuario. 2 usuario digita um valor. 3 O sistemasauva o valor. 4 O sistema mut'pçlica o valor por 2. 5O sistema retorma o resutado ao usuario.</p>	<p style="text-align: center;">Marcos</p> <p>digitar o número recocimento do número dobrar o número retornar o número mutiplicado em dois</p>
<p style="text-align: center;">Renata</p> <p>Primeiro, escolhemos um numero Segundo, depois pegamos outro numero que vai ser multiplicado Terceiro, por fim multiplicamos o numero pelo outro e obtemos o resultado</p>	<p style="text-align: center;">Sara</p> <p>Não realizou a atividade</p>
<p style="text-align: center;">Ana</p> <p>usuario coloque um valor computador o vallor em dobro coloque o valor do dobro</p>	<p style="text-align: center;">Helena</p> <p>A primeira coisa, o computador vai pedir um número O usuario coloca o numero O computador dobra o numero</p>

Fonte: elaborado pelo autor (2024).

Por fim, no último momento do encontro solicitou-se que os participantes construíssem, a partir do algoritmo elaborado na atividade anterior, o código de um programa que solicitasse um número ao usuário e retornasse o dobro desse número. Os códigos construídos pelos participantes são apresentados no Quadro 39.

Quadro 39: Análise a posteriori da avaliação do encontro 3.

Avaliação	Produção esperada dos participantes
<p>Construir um programa que solicite um número ao usuário e retorne o seu dobro.</p>	<pre> variaveis numerico a, b inicio escrever "Digite o numero: " ler a b=2*a escrever "O dobro do numero digitado e: ", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico A,B inicio escrever "digiti um numero" ler A B=A*2 escrever"A multiplicacao e",b fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, b inicio escrever "digite o numero" ler a b=a *2 escrever "a mutiplicacao desse numero e", b fim </pre>
<p style="text-align: center;">Renata</p> <pre> variaveis numerico a, b inicio escrever "digite o numero que queira multiplicaro" ler a b=2*a escrever b fim </pre>	<p style="text-align: center;">Helena</p> <pre> variaveis numerico dobro, numero inicio escrever "digite um numero" ler numero dobro = numero + numero escrever dobro fim </pre>
<p style="text-align: center;">Ana</p> <pre> variaveis numerico a, d inicio escrever "digite um numero" ler a d =a +a escrever d fim </pre>	

Fonte: elaborado pelo autor (2024).

O Quadro 39 mostra que todos os participantes construíram códigos que atendiam ao solicitado. Um ponto de destaque é o fato de que os participantes recorreram a estratégias diferentes para a resolução do problema proposto. Enquanto Adriana, Marcos e Renata representaram o dobro do valor inserido pelo usuário como o resultado da multiplicação deste número por 2, as participantes Helena e Ana optaram por

representar a ação de dobrar como uma operação de soma de um número por ele mesmo.

Encontro 4

No primeiro momento do encontro, objetivava-se apresentar alguns operadores relacionais aos estudantes, identificando se os mesmos já conseguiam, de antemão, localizar suas posições no teclado. De maneira geral, os estudantes já conheciam as localizações dos símbolos apresentados. Em particular, a participante Adriana fez menção à grafia a tinta para o símbolo “maior ou igual”, fato que se verifica no trecho a seguir. Tal fato sugere que a estudante, que apresenta baixa visão, já conhecia o aspecto visual do símbolo (\geq), como ilustrado no trecho a seguir.

Pesquisador: (...) um seguido do outro: maior, igual. O que isso significa? Que vão ser todos os números que sejam ou maiores ou iguais àquele número. Por exemplo...

Adriana: Não vai ser mais um em cima do outro, vai ser um do lado do outro.

Pesquisador: Do lado do outro, isso (...)

Após apresentação dos operadores relacionais aos participantes, procedeu-se à análise de um programa inicial que contivesse comandos utilizados na programação condicional. Nesse momento, foram realizados estímulos de modo que os estudantes pudessem estabelecer conclusões em relação às características dos novos comandos apresentados e ao funcionamento do programa. O Quadro 40 apresenta o comparativo entre análise a priori e respostas obtidas na experimentação.

Quadro 40: Análise a posteriori do momento 2 do encontro 4.

Pergunta prevista	Ação esperada	Respostas observadas
Nesse programa, em quais linhas há comandos que você ainda não conhecia?	<i>Espera-se que o participante identifique no código os comandos “se”, “entao” e “fim_se”, que ainda não haviam sido trabalhados em encontros anteriores.</i>	<p>Adriana: É... a linha que tá o se, é... então “a” é menor do que 10.</p> <p>Marcos: Só o a menor 10 então.</p> <p>Renata: Essa daqui... se a menor que 10 então. (...) E daqui... “oxi”... fim se.. ãhn?</p> <p>Helena: Uma tá escrito se a é... maior ou menor... não, maior ou igual... não, menor ou igual que... não... se a... é porque ela lê estranho... se a menor 10, então.</p>

<p>O que você acha que esse programa faz?</p>	<p><i>Espera-se que o participante infira que o programa retornará a mensagem “O numero digitado e menor que 10” caso o número digitado pelo usuário seja menor que 10.</i></p>	<p>Adriana: Ele vai ver se o número é menor do que 10 e vai imprimir uma mensagem... escrever uma mensagem.</p> <p>Marcos: Ah, cara... eu realmente não sei (...) eu acho que ele simplesmente vai... é, vai pedir para “mim” digitar... dar algum comando ou algum número específico (...) ele vai avaliar o número com base no que tá ali pra ele fazer. Depois ele vai dar um comando e o número vai ter que ser menor do que 10. Por algum motivo, eu não sei qual.</p> <p>Renata: Ele tá pegando o número que a pessoa digita (...) e tá transformando ele num número menor que 10?</p> <p>Helena: (...) se o a for menor que 10, ele vai escrever de novo na tela (...) que o número que eu escrevi é menor que 10.</p>
<p>O que você acha que aparecerá na tela se o usuário digitar o número 7?</p>	<p><i>Espera-se que, nesse caso, o participante infira que o programa retornará a mensagem “O numero digitado e menor que 10”.</i></p>	<p>Adriana: Que ele é um número menor do que 10.</p> <p>Marcos: Ah, meio difícil de dizer. Que eu realmente nunca mexi com uma dessa.</p> <p>Renata: Que o número 7 é menor do que 10.</p> <p>Helena: Ué, vai aparecer que o número 7 é menor que 10.</p>
<p>Considerando a maneira como o programa está escrito, o que você acha que irá acontecer se o usuário digitar o número 15?</p>	<p><i>Espera-se que o participante perceba que, nesse caso, o programa não retornará nenhuma mensagem, já que o número digitado é maior que 10.</i></p>	<p>Adriana: Não vai aparecer nada.</p> <p>Marcos: Talvez ele teria falado que o número é maior do que 10? Será?</p> <p>Renata: Vai terminar o programa.</p> <p>Helena: Não vai aparecer nada!</p>

Fonte: elaborado pelo autor (2024).

De maneira geral, os participantes conseguiram identificar parte dos comandos que ainda não haviam sido trabalhados. Nesse aspecto, os comandos “se” e “entao” foram os mais percebidos de maneira imediata pelos estudantes que participaram do Encontro 4, tendo sido identificados por três participantes e quatro participantes, respectivamente. Por outro lado, apenas uma estudante identificou imediatamente o comando “fim_se”. Cabe colocar, ainda, que alguns participantes não identificaram comandos de maneira imediata, mas o fizeram após estímulos do pesquisador, como ilustram as falas a seguir.

Pesquisador: Tem uma outra coisa que você estranhou também ali no final, que foi o quê?

Marcos: O fim se.

Pesquisador: E a próxima linha?

Helena: Fim se.

Pesquisador: Esse fim se você já conhece?

Helena: Não.

No que se refere à realização de inferências quanto ao funcionamento do programa, os participantes demonstraram maior índice de erro, o que sugere maior grau de dificuldade nesse ponto. Esperava-se que os estudantes, considerando a existência dos comandos “se” e “entao”, imaginassem que o programa retornaria a mensagem “O número digitado e menor que 10” caso o número digitado pelo usuário fosse menor que 10. No entanto, apenas a participante Adriana chegou à conclusão imediata de que seria exibida uma mensagem caso tal fato ocorresse, não explicitando o teor da mensagem. Por outro lado, por mais que a participante Helena também tenha chegado a essa conclusão, foi necessária a mediação do pesquisador para tal, como mostra o trecho da transcrição do encontro a seguir.

Pesquisador: (...) O que você acredita que esse programa vai fazer?

Helena: Hum... ele vai...

Pesquisador: Se quiser ir vindo de novo, também não tem problema.

Helena: Eu acho que ele vai...

Pesquisador: Qual é a primeira coisa que vai acontecer? O que vai aparecer na tela?

Helena: Deixa eu ver...

Pesquisador: Aham.

Helena: Se a... se o número a for menor que 10...

Pesquisador: Que número a é esse?

Helena: Aí é... como assim, que número a é esse?

Pesquisador: [risos]

Helena: [risos] Como assim que número a é esse? É o número a, que tá lá em cima! A variável!

Pesquisador: Aham.

Helena: Aí se o a... ele escreve o a...

Pesquisador: Aham.

Helena: Se o a for menor que 10, ele vai escrever de novo na tela?

Pesquisador: O que ele vai escrever na tela?

Helena: Que o número que eu escrevi é menor que 10.

As dificuldades apresentadas pelos participantes na compreensão do funcionamento do programa podem estar relacionadas ao fato de ser o primeiro momento em que estão sendo explorados comandos associados à tomada de decisões.

Nos casos em que tais dificuldades foram mais evidentes, o pesquisador realizou uma estímulos no sentido de levá-los à compreensão de que a instrução “se entao” é utilizada nos casos em que uma ação deve ser realizada quando uma determinada condição for satisfeita.

As ações seguintes tinham como objetivo avaliar a compreensão dos estudantes em relação à condição observada no programa (se $a < 10$), e a saída fornecida pelo computador em casos de satisfação ou não de tal condição. Observou-se que três dos quatro participantes identificaram que, ao ser digitado o número 7, o programa retornará uma mensagem informando que o número digitado é menor que 7. Os mesmos três participantes conseguiram perceber, ainda, que caso fosse digitado o número 15, o programa não retornaria nenhuma mensagem. Marcos, por sua vez, inferiu que seria exibida uma mensagem informando que o número digitado é maior que 10, possibilidade que havia sido prevista na análise a priori.

Questionados em relação ao que poderia ser feito para solucionar o problema de inexistência de resposta do programa em caso de digitação de número maior que 10, nenhum dos estudantes mencionou explicitamente a possibilidade de criação de uma nova condição envolvendo a instrução “se entao” (se $a > 10$ entao ou se $a \geq 10$ entao). No entanto, algumas das participantes demonstraram ter a noção, mesmo que primitiva, da necessidade de adição de novas linhas de código que contemplassem a possibilidade do valor digitado ser maior que 10. Em particular, Adriana sugeriu o uso da instrução “senão”, ainda não trabalhada com os participantes. Tal fato permite concluir que a estudante já havia tido contato com algum comando similar ao “entao” do ILA. Além disso, a partir da fala “*calma aí, eu tô tentando lembrar o nome em português, calma aí... senão, acho que seria senão*”, expressa pela estudante, infere-se que tal contato possivelmente se deu em um contexto de aprendizagem de uma linguagem de programação cujos comandos são descritos em outra língua. Os trechos a seguir ilustram os fatos mencionados.

Pesquisador: (...) Pensando em como esse programa tá escrito, você tem alguma sugestão pra solucionar esse problema? (...)

Adriana: Eu... é... eu escrevia: se o número... aqui tá aqui, se o número é maior, menor do que 10...

Pesquisador: Uhum.

Adriana: Agora... é... então... não, “peraí”... calma aí, eu tô tentando lembrar o nome em português, calma aí... senão, acho que seria senão.

Pesquisador: Uhum.

Adriana: Senão... é... número maior do que 10 digite... é... número maior do que 10, alguma coisa assim.

Pesquisador: (...) Você tem alguma sugestão do que a gente poderia fazer pra considerar esse... pra consertar esse problema?

Helena: É... acrescentar mais uma linha...

Pesquisador: Uhum...

Helena: Onde vai dizer que esse número não é menor que 10.

Dando seguimento ao encontro, o pesquisador apresentou um novo programa aos participantes dos quatro grupos, conforme previsto na análise a priori. Tal programa consistia em uma pequena alteração do código apresentado anteriormente, com o acréscimo de um bloco de instruções que contemplava o caso em que o número digitado era maior que 10. Os estudantes foram convidados a lerem tal código, sobre o qual foram realizadas algumas perguntas posteriormente. O Quadro 41 apresenta a comparação entre análise a priori e resultados da experimentação.

Quadro 41: Análise a posteriori da segunda parte do momento 2 do encontro 4.

Pergunta prevista	Ação esperada	Respostas observadas
O que você acha que esse programa faz?	<i>Espera-se que o participante infira que o programa retornará a mensagem "O numero digitado e menor que 10" caso o número digitado pelo usuário seja menor que 10, ou a mensagem "O numero digitado e maior que 10" caso o número digitado pelo usuário seja maior que 10</i>	<p>Adriana: Escrever o número digitado menor do que 10, que ele falou. Se o número for maior do que 10, não tá nada dito também. É um programa parecido.</p> <p>Marcos: Bom. Com base naquele número, se eu colocar um sete vai ser reconhecido como menor, e se eu colocar um 15, ou um 16, vai ser reconhecido como maior do que 10.</p> <p>Renata: Se eu digitar um número menor que 10, ele vai dizer que é menor que 10. Se eu digitar um maior que 10, tipo, se eu digitar, sei lá, 14, aí vai dizer que é maior que 10.</p> <p>Helena: Quando o número for maior que 10 ele vai mostrar que o número é maior que 10, já vai aparecer agora.</p>
Qual a diferença desse programa em relação ao anterior?	<i>Espera-se que o participante perceba que no segundo programa há três linhas adicionais que contemplam os casos em que o número digitado pelo usuário é maior que 10.</i>	<p>Adriana: Aqui tem o segundo "se" aqui que é o se o segundo número for maior do que 10, a gente vai... vai aparecer o... "esse número é maior do que 10".</p> <p>Marcos: É porque ele vai reconhecer tanto se o número for menor ou maior.</p> <p>Renata: Que o outro, quando a gente digitava um número maior que 10, ele terminava a execução.</p>

		Helena: Acrescentou as linhas (...) de comando pra o número, se ele for maior, aparecer na tela também.
Você acha que agora o programa está construído da melhor maneira possível? Ou seja, como ele está escrito, existe uma resposta adequada para todos os possíveis valores digitados pelo usuário?	<i>Espera-se que o participante perceba que ainda é necessário considerar o caso em que o número digitado pelo usuário é o 10.</i>	Adriana: Sim, se o número for menor do que 10... menor do que 10, vai aparecer. Se for maior do que 10, vai aparecer. Marcos: Não, necessariamente. (...) Porque aqui ele está com 2 comandos (...) Em locais diferentes. Pode ser que ele atenda a 25% de todas as estimativas. Renata: Sim. Helena: Eu acho que sim.

Fonte: elaborado pelo autor (2024).

O Quadro 41 permite observar que três dos participantes perceberam de imediato que as alterações inseridas no código fazem com que o programa passe a fornecer uma resposta também para o caso em que o número digitado for maior que 10. No caso da participante Helena, por mais que sua resposta exponha apenas o que ocorre quando o número digitado é maior que 10, infere-se que existe o entendimento de que também é exibida uma mensagem para ocasiões em que são digitados números menores que 10, como se percebe na continuidade da transcrição. Por sua vez, Adriana inicialmente cometeu um equívoco na leitura do código, o que impediu uma compreensão adequada do funcionamento do programa. No entanto, após estímulo do pesquisador, consegue identificar seu engano. Os trechos a seguir ilustram o ocorrido com Helena e Adriana.

Adriana: Escrever o número digitado menor do que 10, que ele falou. Se o número for maior do que 10, não tá nada dito também. É um programa parecido.

Pesquisador: Hum, acho que você não leu até o final.

Adriana: Li... tá um fim se e um fim normal.

Pesquisador: Uhum. Perfeito, e é a mesma coisa que o outro? Ou esse programa vai ser diferente? Qual a diferença desse programa em relação ao anterior?

Adriana: Então, tá um fim se aqui no final, eu acho que você abriu um se então dentro de um se então, só que...

Pesquisador: Mas ele só tem uma condição como o outro programa?

Adriana: Só tem uma condição. Escrever o número digitado é menor do que 10. Você não botou a segunda condição.

Pesquisador: Vai para a linha de cima. Essa.

(...)

Adriana: Ah, tá.

Pesquisador: Vai vendo pra cima, tem mais coisas pra cima também.

Adriana: Ah, sim. Entendi, entendi.

Pesquisador: Entendeu?

Adriana: Entendi.

Pesquisador: Então qual é a diferença desse programa em relação ao anterior?

Adriana: Aqui tem o segundo “se” aqui que é o se o segundo número for maior do que 10, a gente vai... vai aparecer o... “esse número é maior do que 10”.

Helena: Quando o número for maior que 10 ele vai mostrar que o número é maior que 10, já vai aparecer agora.

Pesquisador: Uhum, e só isso?

Helena: Só isso o quê? É!

Pesquisador: E se for menor que 10?

Helena: Se for menor que 10 vai aparecer que ele é menor que 10.

O questionamento seguinte se referiu à diferença do segundo programa em relação ao anterior. Por mais que o teor das respostas concedidas pelos participantes tenha sido diverso, nota-se que todos identificaram que o novo código fornecia uma resposta para os casos em que o número digitado pelo usuário fosse maior que 10, em lugar de simplesmente terminar a execução. Em contrapartida, ao observar as respostas dos estudantes à terceira pergunta ilustrada no quadro, nota-se que nenhum deles identificou em um primeiro momento que o programa, considerando a forma como o código estava construído, não fornecia uma saída caso o usuário digitasse o próprio número 10. Tal possibilidade havia sido prevista na análise a priori e, com isso, foram elaborados desdobramentos para a terceira pergunta. Indagados sobre o que ocorreria se fosse digitado o número 10, todos os participantes concluíram que não seria dada uma resposta pelo computador, finalizando-se a execução do programa. Além disso, todos inferiram que, para resolver o problema, seria necessário acrescentar um novo bloco de instruções que considerasse o caso de digitação do número 10. Os trechos a seguir ilustram as respostas dos participantes.

Pesquisador: Beleza. E se o usuário digitar o número 10? O que vai acontecer?

Adriana: Ué, vai aparecer. Não?

Pesquisador: O que que vai aparecer?

Adriana: Vai... ele vai entrar nesses dois casos, esse é um problema. Só se... aconteceria se você escrevesse algo... se o número for igual... se o número fosse igual a 10, então digite tal coisa.

Pesquisador: Perfeito. Então você acha que ainda teria que ser feita essa nova alteração, pra ele ficar escrito da melhor forma possível?

Adriana: Sim. Tá falando que o número é maior do que 10, ou o número é menor do que 10. Mas e o número que seja 10?

Pesquisador: Você consegue pensar em algum número que se você digitar, o computador não vai dar uma resposta?

Marcos: O próprio?

Pesquisador: Que próprio?

Marcos: O próprio 10, no caso.

(...)

Marcos: Então, acho que dá pra colocar aqui, ó, maior ou menor. Uai, mas aí vai ter que abrir outra função ali, cara. Vou ter que abrir outra função aqui embaixo.

Pesquisador: Pode ser. Como é que seria essa outra função que você abriria?

Marcos: Então, acho que dá pra colocar aqui, ó, maior ou menor. Uai, mas aí vai ter que abrir outra função ali, cara. Vou ter que abrir outra função aqui embaixo.

Pesquisador: Pode ser. Como é que seria essa outra função que você abriria?

Marcos: Seria aqui se igual a 10.

Renata: Nada? [risos]

Pesquisador: Nada [risos].

Renata: Ah!

Pesquisador: Então o que tá faltando nesse programa?

Renata: Dizer que ele é igual a 10.

Pesquisador: Dizer que ele é igual a 10? Como assim?

Renata: O computador tem que dizer que ele é igual a 10... só que eu não me lembro como é que é, se é isso mesmo...

Pesquisador: E tá, beleza. Então deixa eu te perguntar, e se o usuário digitar o número 10 nesse programa, o que que vai acontecer?

Helena: Ah, por que você só complica? Não vai aparecer nada!

Pesquisador: [risos] Ah, não vai aparecer nada!

Helena: Ah, você só complica!

Pesquisador: Então, pensando nessa estrutura do programa, o que você poderia fazer então pra resolver esse problema?

Helena: [risos] Ai, por que você faz isso? Se a for igual a 10...

Por fim, no momento de avaliação, os participantes foram estimulados a construir um programa que contivesse a instrução “se entao”. O Quadro 42 ilustra as soluções apresentadas pelos estudantes. Pode-se observar que, à exceção do fato de a participante Renata não ter considerado o caso onde a quantidade de cadernos comprados é exatamente igual a 4, todos os participantes construíram programas que atendiam à solicitação. Por outro lado, foram observados pequenos equívocos no uso da língua portuguesa em pelo menos uma das frases elaboradas por cada um dos estudantes, como em “O valor da coiiompra e:”, “*quantos cadernos deseja compprar*” “*Digite o numero de cadenos*” e “*escreva a quatidade*”.

Quadro 42: Análise a posteriori da avaliação do encontro 4.

Avaliação	Produção esperada dos participantes
<p>Em uma loja onde são vendidos materiais escolares, cada caderno custa R\$8,00. No entanto, a gerência criou uma promoção em que se o cliente compra a partir de 4 cadernos, cada um sai por R\$6,50.</p> <p>Construa um programa que retorne o preço que o usuário irá pagar a partir da quantidade de cadernos comprada.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de cadernos" ler a se (a<4) entao b=8*a fim_se se (a>=4) entao b=6.5*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a, b inicio escrever "Digite o numero de cadernos" ler a se (a<4) entao b=a*8 escrever "O valor da compra e: ", b fim_se se (a>=4) entao b=a*6.50 escrever "O valor da coiiompra e:", b fim_se fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, b inicio escrever "quantos cadernos deseja compprar" ler a se (a>=4) entao b=a*6.50 escrever "o valor total e: ", b fim_se se (a<4) entao b=a*8 escrever "o valor total e:", b fim_se fim </pre>
<p style="text-align: center;">Renata</p> <pre> variaveis numerico a, b inicio escrever "Digite o numero de cadenos" ler a se (a<4) entao b=a*8 escrever "O resultado e igual: ", b fim_se se (a>4) entao b=a*6.50 escrever "o resultado e:", b fim_se fim </pre>	<p style="text-align: center;">Helena</p> <pre> variaveis numerico a, b inicio escrever "escreva a quatidade" ler a se (a<4) entao b=8*a escrever b fim_se se (a>=4) entao b=6.50*a escrever b fim_se fim </pre>

Fonte: elaborado pelo autor (2024).

Encontro 5

No quinto encontro da sequência didática, objetivou-se dar continuidade ao desenvolvimento de habilidades relacionadas à programação condicional, desta vez através da apresentação do comando “senao”. Após breve retomada do programa construído no encontro anterior, os participantes dos quatro grupos foram convidados a

fazerem a leitura de um código onde era utilizada a instrução “se entao” para tomada de decisão. Em seguida, os estudantes foram questionados em relação ao funcionamento do programa. As respostas apresentadas são exibidas no Quadro 43.

Quadro 43: Análise a posteriori do momento 1 do encontro 5.

Pergunta prevista	Ação esperada	Respostas observadas
Existe algum símbolo ou comando nesse programa que você ainda não tinha utilizado? Se sim, qual você acha que é o significado dele?	<i>Espera-se que o participante perceba que não é utilizado nenhum símbolo ou comando que ainda não havia sido mencionado.</i>	<p>Adriana: Tá, é... ele usou o se. (...) Aí depois ele usa se a igual maior do que 10, então... não é menor do que 10. (...) Ah, não tô vendo nada de diferente não.</p> <p>Marcos: Deixa eu só dar uma olhada aqui, nessa última linha aqui. Bom, acho que não. Maior ou igual a 10... (...) Não.</p> <p>Renata: Igual.</p> <p>Helena: Poxa! Acho que não.</p>
O que você acha que esse programa faz?	<i>Espera-se que o participante infira que o programa solicita um número ao usuário e retorna a mensagem “O numero digitado e menor que 10”, caso o número digitado seja menor que 10, ou a mensagem “O numero digitado nao e menor que 10”, caso o número digitado seja maior ou igual a 10.</i>	<p>Adriana: Ele vai digitar o número e vai botar menor do que 10, maior do que 10, não é menor do que 10, quer dizer...</p> <p>Marcos: Ele vai pedir um número. (...) Aí a pessoa vai mandar um número, e vai reconhecer como um número a. (...) Se o a for menor que 10, ele vai avisar. (...) Aí ele vai falar, o número digitado é menor que 10. Se o a for maior ou igual a 10, ele também vai falar.</p> <p>Renata: Eu vou escrever um número e ele vai dizer se é maior que 10. Ou menor.</p> <p>Helena: Ele vai colocar na tela se o número é maior ou menor que 10.</p>

Fonte: elaborado pelo autor (2024).

Nota-se que três dos quatro participantes observaram que o código não apresentava nenhum comando que ainda não havia sido trabalhado nos encontros anteriores. A participante Renata, por sua vez, identifica o sinal de igual como um novo símbolo. Isso pode ter ocorrido pelo fato de embora tal sinal já tenha sido apresentado e utilizado na construção de programas anteriores, nunca tinha sido usado em conjunto com o sinal de maior, formando o operador \geq (\geq).

Em relação à funcionalidade do programa, embora todos os participantes tenham identificado que seria tomada uma decisão, retornando-se uma mensagem diferente de acordo com o fato de o número digitado ser maior ou menor que 10, as respostas diferiram em relação ao teor da mensagem que seria retornada pelo computador. Para Renata e Helena, o programa informaria se o número digitado era maior ou menor que 10. Já Marcos percebeu que uma das condições exibidas no código englobava valores maiores ou iguais a 10 (em lugar de maiores que 10), o que se evidencia na fala “se o a for maior ou igual a 10, ele também vai falar”. No entanto, o participante não menciona qual frase seria exibida na tela nesse caso, apenas colocando que ele (o programa) “também vai falar”. Adriana, por sua vez, foi a única que mencionou que uma das possíveis mensagens exibidas como saída seria de que o número digitado não é menor que 10 (em lugar de o número digitado é maior que 10).

Dando continuidade ao encontro, foi apresentado um segundo programa com código similar ao anterior. No entanto, em lugar de utilizar dois blocos com a instrução “se entao”, um para valores menores que 10, e outro para valores maiores ou iguais a 10, utilizou-se apenas um bloco com a instrução “se entao, senao”. Após a leitura do código pelos participantes, foi realizado um conjunto de perguntas. As respostas esperadas e as observadas são apresentadas no Quadro 44.

Quadro 44: Análise a posteriori da segunda parte do momento 1 do encontro 5.

Pergunta prevista	Ação esperada	Respostas observadas
Existe algum símbolo ou comando nesse programa que você ainda não tinha utilizado? Se sim, qual você acha que é o significado dele?	<i>Espera-se que o participante perceba que o programa apresenta o comando senao, que ainda não havia sido trabalhado. Espera-se, também, que se infira que o comando é utilizado para determinar uma ação a ser executada caso uma condição inicial não seja satisfeita.</i>	<p>Pesquisador: (...) Então tem algum símbolo ou comando que você não tinha utilizado ainda, que você não conhece?</p> <p>Adriana: Senão.</p> <p>Pesquisador: (...) Qual que você acha que é o significado desse comando? O que você acha que vai acontecer no programa?</p> <p>Adriana: O senão é como se fosse uma alternativa, se não... se não acontecer tal coisa, vai ter que acontecer... você vai fazer tal coisa. Seria... uma segunda possibilidade.</p> <p>Pesquisador: (...) Existe algum comando novo nesse programa?</p> <p>Marcos: [risos] ai ai... vamos ver! Hum, tem um comandinho novo aqui no final... senão.</p> <p>Pesquisador: (...) Qual o significado desse senão pra você?</p> <p>Marcos: Hum... senão... alguma coisa se o número não for menor que 10.</p>

		<p>Pesquisador: (...) esse programa tem algum comando que seja novo pra você, que você não tinha visto ainda?</p> <p>Renata: Tem. (...) Senão.</p> <p>Pesquisador: (...) O que você acha que esse comando vai fazer?</p> <p>Renata: Hum... é como uma outra possibilidade.</p> <p>Pesquisador: (...) Tem algum símbolo ou algum comando nesse programa que você ainda não usou?</p> <p>Helena: Tem o senão.</p> <p>Pesquisador: O senão. O que você acha que ele significa?</p> <p>Helena: Uai, o nome já diz, um senão... senão uma coisa acontece.</p>
<p>O que você acha que esse programa faz?</p>	<p><i>Espera-se que o participante infira que por mais que tenha sido construído de forma diferente, o programa, assim como o anterior, solicita um número ao usuário e retorna a mensagem “O numero digitado e menor que 10”, caso o número digitado seja menor que 10, ou a mensagem “O numero digitado nao e menor que 10”, caso o número digitado seja maior ou igual a 10.</i></p>	<p>Adriana: Ele vai... deixa eu ler aqui. Ele vai escrever que o número digitado é menor que 10 ou não é menor do que 10... é menor do que... não é menor do que 10.</p> <p>Marcos: Hum, tá! Ele não está acusado como variável, neste caso. Só que ele tá trabalhando aqui... é... no caso dele ali é se não for menor que 10.</p> <p>Renata: Ele vai dizer que não é maior ou que não é menor do que 10.</p> <p>Helena: Ele vai ser visto com todo aquele processo que a gente faz de se, “na na na na”, ele só vai colocar um senão pra poder diminuir o programa.</p>

Fonte: elaborado pelo autor (2024).

O Quadro 44 mostra que todos os participantes identificaram um comando que ainda não havia sido trabalhado nos programas anteriores, o “senao”. Em relação ao significado do comando, os participantes atribuíram as ideias de “alternativa”, “outra possibilidade” ou “acontecimento de outra coisa”. Por outro lado, apenas a participante Adriana expressou sua compreensão de maneira mais elaborada, argumentando, a seu modo, que se uma determinada condição não é satisfeita, uma segunda possibilidade de saída será fornecida pelo computador.

Já em relação ao funcionamento do programa as respostas foram variadas. Adriana respondeu corretamente, embora de maneira pouco elaborada, que será retornado pelo computador que o número é menor que 10 ou não é menor que 10, de acordo com a

entrada fornecida pelo usuário. Por sua vez, Renata parece ter tido uma compreensão semelhante; no entanto, a estudante colocou que a resposta fornecida seria que o número digitado “não é maior” ou “não é menor do que 10”. O participante Marcos concluiu corretamente que o comando não era uma variável, inferindo que o mesmo estaria relacionado ao caso de o número digitado não ser menor que 10. No entanto, não apresentou nenhuma conclusão em relação ao funcionamento do programa em si. Por fim, Helena foi a única que relacionou o programa com o trabalhado anteriormente, concluindo que o comando “senao” foi utilizado para simplificar o código.

Após a apresentação e discussão dos dois programas foi reforçado, como previsto na análise a priori, que é possível resolver um mesmo problema de diferentes maneiras, utilizando-se comandos distintos e construindo códigos diferentes que, ao fim, vão atender a um mesmo objetivo. Em seguida, pediu-se que os participantes fizessem as modificações necessárias no programa de modo que se o número digitado pelo usuário fosse menor ou igual a 100, deveria aparecer na tela a frase “*O numero nao e maior que 100*”. Em contrapartida, se o número fosse maior que 100, a frase a ser retornada deveria ser “*O numero e maior que 100*”. O Quadro 45 exhibe os códigos construídos pelos participantes.

Quadro 45: Análise a posteriori das alterações realizadas pelos participantes.

Ações a serem realizadas pelo programa	Produção esperada dos participantes
<p>Pedir que o usuário digite um número. Se o número for menor ou igual a 100, deve aparecer na tela "O numero nao e maior que 100". Se o número for maior que 100, deve aparecer na tela "O numero e maior que 100"</p>	<pre> variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numero nao e maior que 100" senao escrever "O numero e maior que 100" fim_se fim ou variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numero nao e maior que 100" fim_se se (a>100) escrever "O numero e maior que 100" fim_se fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numerodigitado nao e maior que 100" senao escrever "O numero digitado n e maior que 100" fim_se fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numero digitadonao e maior que 100" senao escrever "O numero digitado e maior que 100" fim_se fim </pre>
<p>Renata</p> <pre> variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numero digitado não e maior que 100" senao escrever "este numero e maior que 100" fim_se fim </pre>	<p>Helena</p> <pre> variaveis numerico a inicio escrever "Digite o numero" ler a se (a<=100) entao escrever "O numero digitado não e maior que 100" senao escrever "O numero digitado e maior que 100" fim_se fim </pre>

Fonte: elaborado pelo autor (2024).

Observando o Quadro 45, nota-se que por mais que apenas a participante Helena tenha atendido completamente às frases solicitadas como saída (“*O numero nao e maior que 100*” e “*O numero e maior que 100*”), todos conseguiram utilizar corretamente o comando “senao” na construção do código. Em relação aos problemas existentes nas saídas, foram observados: utilização inadequada do espaço entre palavras, como em “*O numero digitadonao e maior que 100*” (Adriana); construção incorreta da frase de saída, como em “*O numero digitado n e maior que 100*” (Marcos); utilização de letra minúscula no início da frase, como em “*este numero e maior que 100*” (Renata).

Os erros observados podem estar associados a limitações relacionadas à visão, o que dificulta ou impede a identificação imediata de equívocos que se referem à escrita. Um exemplo é o fato de que no caso do participante Marcos, que é cego, a simples audição do trecho “*O numero digitadonao e maior que 100*” pode não revelar os problemas envolvendo a ausência ou excesso de espaços na frase. Além disso, outro problema observado foi a dificuldade apresentada por todos os participantes na compreensão da equivalência das expressões “ser menor ou igual a” e “não ser maior que”. Com isso, foi necessária a mediação do professor em todos os grupos para que os participantes concluíssem o código adequadamente.

Em seguida, finalizando o encontro, passou-se ao momento de avaliação, que consistia na construção de um programa que deveria realizar ações diferentes de acordo com a entrada fornecida pelo usuário. Após apresentação da situação problema, pediu-se que os participantes construíssem um programa que atendesse ao que foi solicitado. O Quadro 46 exhibe os códigos construídos pelos participantes.

Quadro 46: Análise a posteriori da avaliação do encontro 5.

Avaliação	Produção esperada dos participantes
<p>Uma companhia telefônica realiza a seguinte promoção para seus clientes:</p> <ul style="list-style-type: none"> - Se o cliente utilizar até 40MB é cobrado um valor fixo de R\$50, além de R\$2 por megabyte utilizado. - Se o cliente utilizar mais que 40MB, segue sendo cobrado R\$2 por cada megabyte utilizado, mas o valor fixo cai pra R\$40. <p>Construa um programa que solicite ao usuário a quantidade de megabytes utilizada no mês e retorne o valor a ser pago pelo cliente.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de megabytes" ler a se (a<=40) entao b=50+2*a senao b=40+2*a fim_se escrever "O valor a ser pago e: R\$", b fim ou variaveis numerico a, b inicio escrever "Digite a quantidade de megabytes" ler a se (a<=40) entao b=50+2*a fim_se se (a>40) entao b=40+2*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a, b inicio escrever "Digite o numero de MB consumido" ler a se (a<=40) entao b=a*2+50 escrever "O nuomero de MB consumido custara ",b senao b=a*2+40 escrever "O numero de MB consumido custara",b fim_se fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, b inicio escrever "digite a quantidade de megabytes utilizada" ler a se (a<40) então b=50+a*2 escrever b se nao b=40+2*a escrever "a quantidade de megabytes e igual a", b fim_se fim </pre>
<p style="text-align: center;">Renata</p> <pre> variaveis numerico a, b inicio escrever "digite quantos mb" ler a se (a<40) entao b=50+a*2 escrever b fim_se se (a>=40) entao b=40+a*2 escrever b fim_se fim </pre>	<p style="text-align: center;">Helena</p> <pre> variaveis numerico a, b inicio escrever "digite a quantidade" ler a se (a<=40) entao b=50+2*a escrever b fim_se se (a>40) entao b=40+2*a escrever b fim_se fim </pre>

Fonte: elaborado pelo autor (2024).

A análise do Quadro 46 permite notar que, embora se observem alguns equívocos, de maneira geral os participantes construíram códigos que atendiam ao que foi solicitado. Em relação aos problemas verificados, nota-se que os participantes Marcos e Renata utilizaram operadores relacionais (< ou >=) de maneira equivocada. Como o valor promocional passaria a ser praticado se a quantidade de megabytes consumidos fosse maior ou igual a 40, esperava-se que os participantes utilizassem as relações $a \geq 40$ e $a < 40$ na construção das condições a serem verificadas. Além disso, Marcos comete um erro sintático ao utilizar “se nao” ao invés de “senao”. Por conta do limite de tempo no encontro, não foi possível testar o código, o que poderia fazer com que o estudante identificasse o equívoco. Assim como na atividade de construção de código que antecedeu o momento de avaliação, os erros cometidos apontam para uma dificuldade dos estudantes na compreensão e utilização adequada de operadores relacionais, especialmente no que se refere ao uso do maior ou igual (\geq) e do menor ou igual (\leq).

Além disso, mais uma vez foram observados erros ortográficos e de concordância na construção das frases de saída. A recorrência de erros dessa natureza pode se justificar pelo fato de não impedirem o funcionamento do programa, e por vezes serem de difícil identificação através da audição (como, por exemplo, uma letra minúscula no início de uma frase).

Encontro 6

No sexto encontro da sequência foi realizada uma atividade avaliativa na qual os participantes deveriam construir dois programas, de modo que cada um atendesse a uma situação problema diferente. O Quadro 47 apresenta os códigos construídos pelos participantes como resposta à primeira situação problema apresentada. Já no Quadro 48 são apresentados os códigos referentes à segunda situação problema.

Em relação à primeira situação problema, os dados apresentados mostram que apenas a participante Adriana construiu um programa que de fato atendia ao que foi solicitado. No que se refere às produções dos demais estudantes, foram observados diversos equívocos que serão discutidos em seguida. Embora todos os participantes tenham iniciado a construção do código corretamente, utilizando o comando “variaveis” e declarando aquelas que seriam utilizadas, bem como seu tipo, foram observados erros como emprego de uma quantidade excessiva de variáveis (Marcos e Renata), não utilização de vírgulas para separá-las (Renata e Helena) e uso de um tipo inadequado

de variável (Renata). Além disso, a participante Helena empregou o comando “início” antes de definir o tipo de variável a ser utilizada, o que também representa um erro.

Dando continuidade ao código, os participantes parecem ter compreendido a necessidade de fornecer uma saída através do comando “escrever”, com o objetivo de solicitar que o usuário digitasse a quantidade de quilômetros percorridos. Embora as frases escolhidas pelos estudantes apresentem problemas de natureza ortográfica, sintática e/ou semântica, infere-se que o objetivo dos participantes, nesse ponto, era solicitar que fosse informada a distância percorrida. Todos os participantes parecem ter compreendido, também, a necessidade do uso do comando de entrada “ler” logo em seguida, na medida em que todos dão continuidade à construção com a linha de código “ler a”. Cabe destacar, no entanto, que a participante Renata utiliza a instrução “ler a” sem ter declarado a variável “a”.

No que se segue, os participantes tomam rumos diferentes na elaboração do código. Ao passo que Helena finaliza sua construção com a instrução “ler a”, Marcos e Renata dão continuidade à construção do programa aparentando, no entanto, pouca clareza em relação a como solucionar o problema.

Em primeiro lugar, Marcos utiliza em dois momentos o comando “se” sem usar em seguida o comando “entao” para definir quais instruções deveriam ser respeitadas caso a condição por ele elaborada fosse satisfeita. De qualquer forma, a própria utilização do comando “se”, neste caso, é desnecessária, uma vez que a taxa de bandeirada e o valor por quilômetro rodado são constantes em todos os cenários possíveis do problema. Marcos demonstra, ainda, confusão em relação ao significado das variáveis “a” e “b” declaradas, na medida em que os trechos “se (a=6)” e “se (a=3)” não fazem sentido considerando-se o problema proposto.

Além disso, Marcos apresenta problemas na utilização dos comandos de entrada (ler) e saída (escrever), uma vez que alterna construções adequadas (“*escrever ‘digite a quantidade de km percorridos’*”; “*escrever ‘o valor da corrida e igual’*”; “ler a”) com inadequadas (“*escrever ‘digite o valor da corrida’*”; “*escrever (a+b=soma)*”; “ler b”; “*leer soma*”) à estrutura do programa. Em particular, o estudante parece utilizar o trecho “*escrever (a+b=soma)*” para atribuir à variável declarada “soma” o resultado de $a + b$. No entanto, comete tanto o equívoco da própria utilização do comando escrever, desnecessária nesse caso, quanto uma inversão na construção da instrução, na medida em que o formato correto seria “*soma=a+b*”.

Ainda no que se refere à variável “soma”, Marcos parece utilizá-la para representar o valor total da corrida. No entanto, o estudante constrói de maneira incorreta a lei de formação da função que representa o total gasto, na medida em que não explicita os coeficientes angular e linear da função utilizada. Desse modo, em lugar da relação $soma=a+b$, deveria ter sido utilizada a relação $soma=3*a+6$, ou $soma=3*b+6$, dependendo de qual seria a variável dependente no problema. Por outro lado, considerando a forma como o programa foi construído, é possível que o participante tenha utilizado as variáveis para representar, na realidade, os coeficientes angular e linear da função, mencionados anteriormente. Tal hipótese se justifica a partir da análise dos trechos ilustrados a seguir:

```
escrever "digite o valor da corrida"
ler a
se (a=6)
```

```
escrever "digite a quantidade de km percorridos"
ler b
se (b=3) por km rodado
```

No primeiro trecho, Marcos inicialmente parece associar a variável “a” a um possível “valor para a corrida”. No entanto, considerando que o estudante atribui o valor 6 à variável no trecho “se (a=6)”, é possível que o mesmo a tenha utilizado para representar uma constante, a bandeirada da corrida. Desse modo, o termo “valor da corrida” pode ter sido utilizado como referência ao valor da bandeirada, e não ao valor total da corrida em si. De modo análogo, considerando o que se verifica nas linhas de código “*escrever ‘digite a quantidade de km percorridos’*” e “*ler b*”, em um primeiro momento o estudante parece compreender a variável “b” como a variável independente da função, que representaria a quantidade de quilômetros percorridos. No entanto, no trecho “se (b=3) por km rodado”, Marcos parece associar “b” ao valor constante cobrado a cada quilômetro percorrido. Assim, além de apresentar problemas na utilização de comandos de entrada e saída, o participante aparenta não ter clareza em relação ao significado das variáveis na construção do código.

Por sua vez, Renata utiliza de maneira correta os comandos para solicitar que o usuário digite a quantidade de quilômetros percorridos e para fazer a leitura do valor digitado, à exceção do equívoco cometido ao não utilizar as aspas para fechar a frase

elaborada como saída no trecho “*escrever ‘digite o valor do kilometro’*”. No entanto, a participante comete uma série de erros mais evidentes ao dar continuidade à construção do código.

A análise das duas linhas de código que seguem sugere que a estudante compreendeu a estrutura da função que determina o valor total da corrida. Ao escrever “*a multiplica b somoa c = d*”, Renata parece compreender que é necessário realizar uma multiplicação seguida de uma soma para calcular o valor mencionado. Em seguida, a estudante constrói a linha de código “*a xx 3 + 6 = ‘a equivale ao resultado’*”, o que sugere ainda mais fortemente que houve a compreensão de que é necessário multiplicar a quantidade de quilômetros percorridos (*a*) por 3, e somar o resultado da multiplicação por 6. No entanto, a estudante comete erros sintáticos que impedem que o programa de fato realize a operação desejada.

Em primeiro lugar, Renata utiliza variáveis que não foram declaradas no início do código, como “*a*”, “*b*”, “*c*” e “*d*”. Além disso, a estudante usa o caracter “*x*” em lugar do asterisco “***” com o objetivo de representar uma multiplicação, o que representa um erro sintático. Ainda, Renata não atribui o resultado da operação realizada a uma segunda variável ($b=a*3+6$). Nesse ponto, a participante parece acreditar que ao escrever “*a xx 3 + 6 = ‘a equivale ao resultado’*”, o resultado da operação $a.3+6$ seria exibido imediatamente antes do trecho “*a equivale ao resultado*”, onde o termo “*a*” possivelmente representa um erro de digitação. Finalmente, Renata utiliza o comando “*fim_se*” em lugar do comando “*fim*” para concluir a escrita do código. Desse modo, a participante aparenta ter compreendido o problema, bem como seu mecanismo de resolução, ao propor uma relação adequada para a determinação do valor da corrida. No entanto, demonstra problemas relacionados à construção do algoritmo e utilização dos comandos do ila.

Quadro 47: Análise a posteriori da primeira situação problema do encontro 6.

Avaliação	Produção esperada dos participantes
<p>Nos últimos anos, aplicativos para transportes de passageiros em automóveis privados têm se multiplicado, oferecendo serviços voltados tanto para o público geral, quanto para grupos específicos. Ainda assim, em virtude de problemas como excesso de cancelamento de viagens e aumento de tarifas, os táxis por vezes continuam representando uma opção atrativa no transporte de passageiros. Na cidade de Taxilandia, por exemplo, a corrida de táxi é calculada da seguinte maneira:</p> <p>- É cobrada a taxa de R\$6 referente à bandeirada (valor fixo inicial de qualquer corrida), independente do horário em que seja realizada a corrida;</p> <p>- É cobrado o valor adicional de R\$3 por quilômetro rodado.</p> <p>Nessas condições, construa um programa que determine o valor a ser pago pelo usuário de acordo com a quantidade de quilômetros rodados.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de quilometros" ler a b=6+3*a escrever "O valor a ser pago e: R\$", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a, b inicio escrever "Quantos K foi a corrida" ler a b=a*3+6 escrever "O valor da corrida",b fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, b, soma inicio escrever "digite o valor da corrida" ler a se (a=6) escrever "digite a quantidade de km percorridos" ler b se (b=3) por km rodado escrever (a+b=soma) leer soma escrever "o valor da corrida e igual", soma fim_se fim </pre>
<p style="text-align: center;">Renata</p> <pre> variaveis caracter numerico inicio escrever "digite o valor do kilometro ler a a multiplica b somoa c = d a xx 3 + 6 = "a equivale ao resultado" fim_se </pre>	<p style="text-align: center;">Helena</p> <pre> variaveis INICIO numerico a b escrever "digite a quilometragem" ler a </pre>

Fonte: elaborado pelo autor (2024).

Já no que se refere à segunda situação problema apresentada, a análise do Quadro 48 revela que nenhum dos participantes apresentou um programa que atendesse ao solicitado. Assim como na primeira situação, os estudantes iniciam a construção do código como esperado, declarando as variáveis a serem utilizadas, bem como seus tipos. No entanto, por mais que seja possível resolver o problema construindo um programa com um número maior de variáveis, nesse caso seria suficiente a utilização de apenas duas, e Adriana é a única participante que declara essa quantidade. Por outro lado, Adriana não separa o comando “numerico” da variável “a”, o que representa um erro sintático. Dando continuidade à construção do código, todos os participantes cometem uma série de erros que comprometem em maior ou menor grau a estrutura do código. Tais erros são discutidos a seguir.

Em primeiro lugar, Adriana inicia o bloco de instruções utilizando o comando correto para solicitar que o usuário digite a quantidade de tapetes vendidos, embora cometa um equívoco sintático ao não utilizar as aspas para abrir a frase exibida como saída em “*escrever Quantoos tapetes foi vendido*”. Ademais, a frase elaborada apresenta erros ortográficos e sintáticos, além de se verificar um excesso de espaços na construção.

Em seguida, a participante apresenta o bloco referente ao primeiro cenário possível do problema, caso em que a quantidade de tapetes vendidos é maior que 100, e nesse ponto novos equívocos são cometidos. Na linha de código “*s(a>=100) enttao*”, a participante comete erros sintáticos ao digitar “s” e “enttao”, em lugar de “se” e “entao”, e ao não separar a condição “(a>=100)” do comando que inicia a estrutura condicional. Além disso, se observa um erro semântico, na medida em que o bônus é concedido se a quantidade de tapetes vendidos é maior que 100, e não a partir de 100. Assim, em lugar de “(a>=100)”, a construção correta seria “(a>100)”. Além disso, na elaboração da lei de formação da função que determina o salário dos vendedores, Adriana ignora a comissão que cada vendedor recebe por tapete vendido. Desse modo, em seu código, o valor recebido por quem alcança a bonificação acaba sendo uma constante, representada pela soma entre a parte fixa do salário e o bônus, independente da quantidade de tapetes vendidos ($b=1500 + 300$).

Para representar o que ocorre no segundo cenário possível do problema, Adriana utiliza a estrutura “*se_ naoentao=1500*”, possivelmente com intencionalidade de utilização do comando “senao” para definir que para uma quantidade de tapetes vendidos inferior a 100, o salário do vendedor corresponde a R\$1500. Nesse caso, do ponto de

vista sintático, além de utilizar o termo “se_ nao” ao invés de “senao”, Adriana usa equivocadamente o comando “entao”, desnecessário nesse contexto, sem, ainda, separar os termos com um espaço. Já do ponto de vista semântico, mais uma vez a participante ignora a comissão recebida por tapete vendido, apresentando como solução um salário constante de R\$1500 no cenário sem bonificação. Por fim, a estudante não utiliza o comando “fim” para finalizar o programa construído.

Marcos, por sua vez, declara corretamente as variáveis que pretende utilizar (embora em quantidade excessiva), e inicia o bloco de instruções utilizando o comando escrever para solicitar que o usuário digite o valor do salário. Tal construção não tem sentido levando-se em conta o problema proposto, uma vez que o salário do vendedor é exatamente o que o programa deve retornar ao usuário. Por outro lado, após utilizar o comando de entrada “ler a”, o participante faz novo uso do comando “escrever” para pedir que o usuário insira a quantidade de tapetes vendidos, cometendo um erro sintático ao não utilizar as aspas para fechar a frase que elaborou (*escrever "quantos tapetes voce vendeu"*). Considerando que o participante cometeu equívoco semelhante na avaliação do Encontro 6, é possível que o mesmo tenha utilizado a variável “a” para se referir a uma constante, a parte fixa do salário dos vendedores. Em todo caso, o estudante finalizou seu código nesta etapa, não dando seguimento ao desenvolvimento do programa.

Já Renata inicia seu código declarando um número de variáveis maior que o necessário para a resolução do problema, e não utilizando vírgula para separá-las. Em seguida, utiliza corretamente comandos de saída e entrada para solicitar que o usuário forneça a quantidade de tapetes vendida e associar tal quantitativo à variável “a”. No entanto, ao dar continuidade ao código, apresenta a relação “ $a \times 16 \times 1500 + 300 = a$ ”, possivelmente com intencionalidade de representar o salário dos vendedores como a soma entre 300 e o produto entre 1500, 16 e o número de tapetes vendidos (a). Nesse sentido, a participante comete os seguintes equívocos: utilização do caracter “x” em lugar do asterisco (*) para representação do produto; inversão na atribuição do valor da variável, utilizando a expressão “ $a \times 16 \times 1500 + 300 = a$ ”, em lugar de “ $a = a \times 16 \times 1500 + 300$ ”; e utilização da variável “a” em lugar de uma outra variável para o resultado da operação, que representa o valor do salário.

Ainda no que se refere à operação aritmética apresentada por Renata, pode-se observar que em lugar de somar o valor da parte fixa do salário (1500) à parte variável

proveniente das comissões (a.16), a participante efetua uma multiplicação, o que não tem sentido matematicamente considerando o problema proposto. Ademais, esse é apenas um dos cenários possíveis, já que se o vendedor não alcança a marca de 100 tapetes vendidos, o bônus de R\$300 não é adicionado ao salário, o que modificaria a função utilizada para o cálculo. Desse modo, seria necessária a utilização de uma estrutura condicional, o que não foi feito pela participante.

Por fim, Renata comete um possível erro de digitação ao utilizar o termo “escrevber” em lugar do comando “escrever” para apresentar uma saída ao usuário. Além disso, a solução apresentada pela participante é uma constante (11500) que não tem sentido nem considerando o problema proposto, já que o valor do salário é dependente da quantidade de tapetes vendidos, tampouco levando em conta a operação aritmética apresentada pela estudante, já que $16 \cdot 1500 + 300$ não equivale a 11500.

Já Helena inicia seu código declarando as variáveis de seu programa, equivocando-se ao não separá-las com vírgula. Em seguida, utiliza corretamente os comandos “escrever” e “ler” para solicitar que o usuário insira a quantidade de tapetes vendidos e registrar esse número na variável “a”. Então, começa a construir uma estrutura condicional referente ao cenário em que não há bonificação. Aqui, dois equívocos sintáticos são cometidos: a estrutura mencionada não é finalizada com o comando “fim_se”; e o resultado da operação “ $1500+16*a$ ” não é atribuído a nenhuma das outras variáveis declaradas (b ou x). Além disso, observa-se um erro de natureza semântica similar ao cometido por Adriana na medida em que, considerando o problema proposto, se um vendedor atingisse a marca de exatos 100 tapetes vendidos, a bonificação ainda não seria concedida. Assim, em lugar de “ $(a<100)$ ”, a estrutura correta seria “ $(a\leq 100)$ ”. Por fim, Helena não dá continuidade à construção do código, ignorando o cenário em que há bonificação ($a>100$), e tampouco utilizando o comando “fim” para explicitar o término do programa.

Quadro 48: Análise a posteriori da segunda situação problema do encontro 6.

Avaliação	Produção esperada dos participantes
<p>A comissão é um benefício muitas vezes utilizado no setor comercial para motivar os empregados e, conseqüentemente, aumentar os lucros da empresa. Tulio é proprietário de uma loja de tapetes, e para tentar impulsionar as vendas no período de Natal, decidiu conceder, além das comissões, um bônus aos funcionários que atingissem a meta estabelecida para a quantidade de tapetes vendidos. Assim, a renda mensal dos vendedores de sua loja passou a ser composta por uma parte fixa de R\$1500, e uma parte variável que dependia da quantidade de tapetes que o vendedor conseguisse vender. Para o cálculo da parte variável, ficou estabelecido que cada vendedor ganharia uma comissão de R\$ 16 por tapete vendido e, além disso, se o vendedor conseguisse vender mais de 100 tapetes no mês, ganharia adicionalmente um bônus de R\$300.</p> <p>Nessas condições, construa um programa que Tulio poderia utilizar para calcular o salário dos vendedores da loja de acordo com a quantidade de tapetes vendidos.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de tapetes" ler a se (a>100) entao b=1800+16*a senao b=1500+16*a fim_se escrever "O valor a ser pago e: R\$", b fim ou variaveis numerico a, b inicio escrever "Digite a quantidade de tapetes" ler a se (a>100) entao b=1800+16*a fim_se se (a<=100) entao b=1500+16*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a, b inicio escrever "Quantoos tapetes foi vendido" ler a s(a>=100) enttao b=1500+300 fim_se se_naoentao=1500 </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, b, c inicio escrever "digite o valor do salario" " ler a escrever "quantos tapetes voce vendeu </pre>
<p style="text-align: center;">Renata</p> <pre> variaveis numerico a b c inicio Escrever "digite o número de tapetes comprados" ler a a x 16 x 1500 + 300 = a escrevber "o numero dig itado é 11500" fim </pre>	<p style="text-align: center;">Helena</p> <pre> VARIAVEIS NUMERICO A B X INICIO ESCREVER "DIGITE O NUMERO DE TAPETES VENDIDOS" LER A SE (a<100) entao 1500+16*a </pre>

Fonte: elaborado pelo autor (2024).

Encontro 7

No sétimo e último encontro da sequência, realizou-se uma segunda atividade avaliativa. Assim como na primeira, os participantes deveriam construir dois programas com o objetivo de resolver duas situações problema diferentes. No entanto, neste encontro não seria possível utilizar nenhum material para consulta. No Quadro 49 são exibidos os códigos construídos pelos participantes como resposta à primeira situação problema apresentada no encontro. Já no Quadro 50 são apresentados os códigos referentes à segunda situação problema.

No que se refere à primeira situação problema proposta, apenas a participante Adriana apresentou um programa que atendeu ao solicitado. De maneira geral, no código construído pela estudante podem ser verificados apenas pequenos equívocos que não comprometem o funcionamento do programa. Entre os problemas identificados, o primeiro é o fato de a participante não utilizar letras maiúsculas no início das frases elaboradas como saída (“*digite o numero de meses*” e “*voce tem*”). Além disso, na linha de código “*escrever ‘voce tem’,b*”, seria necessário adicionar um espaço após a palavra *tem*, evitando assim que o valor acumulado ao longo dos meses aparecesse imediatamente junto à palavra “*tem*” na tela do computador (“*voce tem235*”, como exemplo). No entanto, como mencionado, as questões observadas não impedem o programa funcione adequadamente.

Marcos, por sua vez, inicia a construção do código corretamente, declarando as variáveis que pretende utilizar. No entanto, dando seguimento à sua produção, o participante parece não compreender completamente o problema proposto, na medida em que utiliza o comando “*escrever*” para solicitar que o usuário digite o valor depositado por Mariana a cada mês, valor este que já havia sido definido previamente como R\$35. Além disso, o estudante não utiliza letra maiúscula no início da solicitação elaborada como saída, nem no nome próprio empregado na frase.

A seguir, após instruir a leitura do número digitado através da linha de código “*ler a*”, Marcos informa que a variável “*b*” é dada pela soma entre a constante 200 e o valor atribuído à variável “*a*”, o que representa um erro semântico. Além disso, o participante não explicita como a quantidade de meses decorridos influi na quantia acumulada. Por fim, o participante usa o comando “*fim_se*” antes de finalizar o código com o comando “*fim*”, um erro sintático na medida em que não foi utilizada qualquer estrutura condicional no programa.

Já Renata apresenta um código no qual se observam diversos erros, de modo que o programa construído não corresponde minimamente ao solicitado. Em primeiro lugar, a participante realiza a declaração de variáveis em dois momentos de sua construção, quando na realidade as variáveis devem ser declaradas unicamente no início do código. Além disso, para tal declaração, Renata utiliza de maneira equivocada o termo “*Variavel*” em lugar do comando “variaveis”, o que representa erro sintático. Ainda neste sentido, os termos que deveriam representar variáveis no código apresentado, na realidade parecem conectar-se constituindo sentenças, o que demonstra que a participante não tem clareza do que representam as variáveis no programa (“*numerico a soma b igual c*” e “*numerico a multiplicado por b igual c*”).

Em segundo lugar, a participante parece não compreender o significado dos comandos de entrada e saída, cometendo, nesse sentido, diferentes equívocos de natureza conceitual. A estudante, por exemplo, aparenta utilizar o comando “escrever” para orientar alguma ação, como em “*Escrever ‘Some a contia ja ganha com o valor adquirido do mes seguinte*””, e em “*Escreva ‘Multiplique o dinheiro guardado durante um ano*””. Por outro lado, o comando “ler” parece ser utilizado como um recurso de saída, e não de entrada, como em “*Ler ‘o valor foi 235*”” e em “*Ler ‘O valor é 93 mas somado a quantia anterior e 328*””.

Além disso, Renata também aparenta não ter compreendido o problema em si, na medida em que insere as linhas de código “*200+35=235*” e “*35*12=93*” no código, e apresenta um valor equivalente à soma dos dois resultados como solução para o problema ao final do programa. Nesse caso, nem as linhas mencionadas fazem sentido sintaticamente, como tampouco o problema poderia ter como solução uma constante, na medida em que o valor acumulado é uma variável dependente da quantidade de meses decorridos. Por fim, embora não se verifique estrutura condicional em nenhum momento do código, o comando “fim_se” é utilizado em sua construção. Além disso, o programa é concluído com a linha de código “*fim se _*”, possivelmente com intencionalidade de utilização do comando “fim_se”, o que também representaria um erro.

De maneira geral, a quantidade e a natureza dos erros cometidos por Renata destoam do que foi apresentado pela estudante nos encontros anteriores. Tal fato pode ser consequência do longo espaço de tempo decorrido entre os encontros 6 e 7, no caso da participante. Como mostra o Quadro 31, Renata realizou a primeira avaliação (Encontro 6) no dia 19 de dezembro de 2022, última semana de aulas naquele ano no

Colégio Pedro II. Já a segunda avaliação (Encontro 7) só pôde ser realizada após o período de férias da estudante, tendo ocorrido quase dois meses depois, no dia 13 de fevereiro de 2023. Desse modo, tal descontinuidade no andamento da sequência pode ter impactado negativamente seu rendimento no segundo momento avaliativo.

Por fim, Helena parece ter compreendido o problema e constrói um código que apresenta poucos erros. Em primeiro lugar, a participante declara as variáveis a serem utilizadas no programa, cometendo dois equívocos sintáticos nessa etapa: o uso do termo “variavel” em lugar do comando “variaveis”, e a inexistência de vírgula para separar as duas variáveis declaradas. Em seguida, a estudante utiliza corretamente os comandos de entrada e saída para solicitar que o usuário insira o número de meses decorridos e fazer a leitura do número digitado. Nesse ponto, Helena comete apenas equívocos no uso da língua portuguesa ao não utilizar letra maiúscula no início da frase, nem flexionar corretamente o verbo digitar no trecho “digiten o numero de meses”. No entanto, dando seguimento à construção do código, Helena utiliza novamente o comando “ler” na linha de código “*ler 200+a*35*”, quando na realidade deveria atribuir o resultado da operação $200+a.35$ à variável “b”, utilizando a estrutura “*b=200+a*35*”.

No que segue, a participante deixa, sem necessidade, uma linha em branco antes de utilizar um comando de saída para apresentar ao usuário o que seria o valor acumulado ao longo dos meses, representado pela variável “b”. Destaca-se novamente, entretanto, que a participante não utilizou anteriormente a estrutura adequada para associar tal valor à variável “b” (“*b=200+a*35*”). Finalmente, a estudante não utiliza o comando “fim” para explicitar o término do programa.

Quadro 49: Análise a posteriori da primeira situação problema do encontro 7.

Avaliação	Produção esperada dos participantes
<p>Mariana ganhou de sua mãe como presente de Natal um cofre que já continha uma nota de R\$200. Animada com o presente, decidiu, a partir do mês seguinte, depositar R\$35 no cofre a cada mês. Construa um programa que determine quantos reais haverá no cofre de acordo com a quantidade de meses passados.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de meses" ler a b=200+35*a escrever "O valor total e: R\$", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a, b inicio escrever "digite o numero de meses" ler a b=a*35+200 escrever "voce tem",b fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, b inicio escrever "digite o valor depositado por mariana a cada mes" ler a b=200+a escrever b fim_se fim </pre>
<p style="text-align: center;">Renata</p> <pre> Variavel numerico a soma b igual c Escrever "Some a contia ja ganha com o valor adquirido do mes seguinte" 200+35=235 Ler "o valor foi 235" fim_se variavel numerico a multiplicado por b igual c Escreva "Multiplique o dinheiro guardado durante um ano" 35*12=93 Ler "O valor é 93 mas somado a quantia anterior e 328" fim se _ </pre>	<p style="text-align: center;">Helena</p> <pre> variavel numerico a b inicio escrever "digiten o numero de meses" ler a ler 200+a*35 escrever b </pre>

Fonte: elaborado pelo autor (2024).

Quanto à segunda situação problema apresentada, a análise do Quadro 50 revela que embora alguns participantes tenham se aproximado da produção esperada, nenhum deles construiu um programa que de fato atendesse ao solicitado. As construções individuais, bem como os problemas observados, são detalhados a seguir.

Adriana inicia seu código declarando corretamente as variáveis que serão utilizadas no programa, e usando comandos de saída e entrada para solicitar e registrar a quantidade de horas frequentadas na academia. Em seguida, a participante constrói três cenários possíveis para valores atribuídos à variável “a” ($a > 10$, $a = 10$ e $a < 10$), criando uma estrutura condicional para abordar cada caso.

Do ponto de vista semântico, Adriana comete um erro na determinação do valor total a ser pago no cenário em que o usuário frequenta a academia exatamente 10 horas durante o mês. Nesse caso, na instrução “*se(a=10) entao b=30*”, ao estabelecer que se $a = 10$ o valor a ser pago equivale a R\$30, a estudante ignora o fato de que, além do valor fixo de R\$30, nesse cenário também devem ser cobrados R\$2 por hora frequentada. Já do ponto de vista sintático, Adriana utiliza equivocadamente o termo “se_fim” ao invés do comando “fim_se” nos cenários onde $a > 0$ e $a < 0$, e não utiliza nenhum comando desse tipo para finalizar o cenário onde $a = 10$. Cabe colocar, ainda, que embora isso não comprometa o funcionamento do programa, a participante comete erros de digitação ou no uso da língua portuguesa em todas as frases elaboradas como saída (“*horas frequentadas*”, “*voce dee paenao*”, “*voce devepagar*”, “*voce deve pagar*”).

Marcos também inicia o código de maneira correta, declarando as variáveis que serão utilizadas no programa e usando os comandos “escrever” e “ler” para solicitar e registrar a quantidade de horas frequentadas na academia, cometendo pequenos erros no uso da língua portuguesa em “*digite o numero de horas frequentadas na cademia*”. Em seguida, diferentemente de Adriana, o participante elabora dois cenários possíveis: um para uma quantidade de horas frequentadas menor que 10 ($a < 10$), e um para uma quantidade maior ou igual a 10 ($a \geq 10$).

No primeiro cenário, Marcos acerta ao definir que o valor a ser pago é de R\$80, mas comete alguns erros sintáticos, a saber: utilização de vírgula após a condição “*se (a < 10)*”; uso da variável “a” ao invés de “b” na representação do valor a ser pago em “*a=80*”, e omissão do comando “fim_se” na finalização do bloco de instruções. Já no segundo cenário, o participante novamente utiliza a vírgula após ($a \geq 10$) e, ainda, além de utilizar a variável “a” em lugar de “b” para representar o valor gasto no mês, cria uma função constante para tal valor ($a = 30 + 2$), que independe da quantidade de horas frequentadas na academia.

Por fim, Marcos utiliza a instrução “*escrever b*” para retornar ao usuário o valor a ser pago pelo cliente; no entanto, não é atribuído um valor à variável “b” em nenhum

trecho do código. Além disso, como tal instrução é utilizada antes do comando “fim_se”, a mesma se insere no bloco de instruções referente ao cenário onde $a \geq 10$. Desse modo, seria necessário utilizá-la também no outro bloco, onde $a > 10$. Alternativamente, a instrução poderia ser usada após o comando “fim_se”, não estando, desse modo, associada a nenhum dos dois blocos.

Já Renata, assim como observado na primeira situação problema do encontro 7, constrói um código que não corresponde minimamente ao solicitado, possivelmente pelos mesmos motivos mencionados anteriormente. Em primeiro lugar, novamente a participante parece utilizar sentenças no campo onde deveriam ser declaradas as variáveis do programa (“*numerico a = menor que 10 b maior que 10*”). Em seguida, embora pareça fazer referência aos valores fixos a serem pagos em dois cenários distintos, Renata constrói duas linhas de código sem utilizar qualquer comando identificável no ILA (“*80 menor que 10*” e “*30 maior que 10*”). Tal fato se repete mais adiante em seu código, quando são criadas as linhas “*maior que 10*” e “*menor que 10*”.

No que se refere ao uso de comandos de entrada e saída, embora a participante utilize o comando “escrever” corretamente para solicitar a quantidade de horas frequentadas na academia, são verificados erros no uso da língua portuguesa na frase elaborada (“*digite o horário frequentados menssalmente*”). Além disso, tal instrução se encontra isolada no código, não sendo utilizado nenhum comando de entrada para o registro da informação fornecida pelo usuário. Nesse sentido, o comando “ler” é usado posteriormente em “*ler 'horário é igual*”, mas a construção não tem sentido do ponto de vista sintático. Por fim, a participante utiliza aspas após o comando “fim”, o que também representa um erro.

Finalmente, Helena inicia seu código corretamente declarando as variáveis do programa e utilizando o comando “escrever” para solicitar a inserção da quantidade de horas frequentadas, cometendo apenas dois equívocos até este ponto: não utilização de vírgula para separação das variáveis em “*numerico a b*”, e não utilização de letra maiúscula no início da frase “*digite a quantidade de horas*”. Em seguida, a participante utiliza o comando “ler” de maneira incorreta, aparentando ter a intenção de iniciar a elaboração de uma estrutura condicional (“*ler "se a maior que 10"*”). Cabe destacar que, assim como Renata, Helena utiliza a palavra “maior” ao invés do operador “>”, este último de fato identificável pelo ILA.

Dando seguimento à construção do código, a participante parece continuar um bloco de instruções referente ao cenário em que o número de horas frequentadas é maior que 10, na medida em que apresenta a linha “*emtao 30+a*2*”. Aqui, embora Helena acerte na relação que determina o valor a ser pago pelo cliente no cenário mencionado, dois equívocos de natureza sintática são cometidos: utilização do termo “*emtao*” em lugar do comando “*entao*”, e não atribuição do resultado da operação $30+a*2$ à variável “*b*”, através da instrução “*b=30+a*2*”. Com isso, na sequência a estudante comete um erro semelhante ao cometido por Marcos, já que é utilizada a instrução “*escrever b*” sem que tenha sido atribuído um valor à variável “*b*” em nenhum momento no código. Como Helena finaliza seu programa nesse ponto, os cenários em que as quantidades de horas frequentadas na academia não são maiores que 10 são ignorados pela estudante.

Quadro 50: Análise a posteriori da segunda situação problema do encontro 7.

Avaliação	Produção esperada dos participantes
<p>Academias de ginástica e musculação têm se espalhado por todo o país, passando a fazer parte do cotidiano de muitos brasileiros. No entanto, é comum encontrar pessoas que inicialmente se engajam na realização de atividades físicas mas, com o passar o tempo, passam a se sentir desmotivadas e abandonam a prática. Pensando em atrair novos clientes e estimular a realização de exercícios físicos entre eles, uma academia lançou a seguinte promoção:</p> <p>- Se o cliente frequentar a academia menos que 10 horas no mês, deve pagar apenas um valor fixo de R\$80.</p> <p>- Se o cliente frequentar a academia 10 ou mais horas no mês, o valor fixo cai para R\$30, e é cobrado um adicional de R\$2 por cada hora utilizada na academia.</p> <p>Construa um programa que determine o valor a ser pago pelo cliente de acordo com a quantidade de horas frequentadas na academia.</p>	<pre> variaveis numerico a, b inicio escrever "Digite a quantidade de horas" ler a se (a<10) entao b=80 senao b=30+2*a fim_se escrever "O valor a ser pago e: R\$", b fim ou variaveis numerico a, b inicio escrever "Digite a quantidade de tapetes" ler a se (a<10) entao b=80 fim_se se (a>=10) entao b=30+2*a fim_se escrever "O valor a ser pago e: R\$", b fim </pre>
Produção dos participantes	
<p style="text-align: center;">Adriana</p> <pre> variaveis numerico a, b inicio escrever "horas frequentadas " ler a se (a<10) entao b=80 escrever "voce dee paenao",b se_fim se(a=10) entao b=30 escrever "voce devepagar",b senao (a>10) entao b=a*2+30 escrever "voce deve pagar",b se_fim fim </pre>	<p style="text-align: center;">Marcos</p> <pre> variaveis numerico a, b inicio escrever "digite o numero de horas frequentadas na cademia" ler a se (a<10), entao a=80 se (a>=10), entao a=30+2 escrever b fim_se fim </pre>
<p style="text-align: center;">Renata</p> <pre> variaveis numerico a = menor que 10 b maior que 10 80 menor que 10 30 maior que 10 escrever "digite o horário frequentados menssalmente" maior que 10 menor que 10 ler "horário é igual fim" </pre>	<p style="text-align: center;">Helena</p> <pre> variavel numerico a b inicio escrever "digite a quantidade de horas" ler "se a maior que 10" emtao 30+a*2 escrever b </pre>

Fonte: elaborado pelo autor (2024).

Por fim, levando em conta o objetivo geral da pesquisa, no Quadro 51 é exibido um resumo com os principais problemas e dificuldades apresentados pelos participantes ao longo dos encontros. Já no Quadro 52, são mostradas as mais evidentes potencialidades quanto ao uso do ILA como recurso para o desenvolvimento do pensamento computacional e a resolução de problemas de matemática através da programação de computadores. Tanto as limitações quanto as potencialidades foram identificadas a partir da etapa de experimentação e da análise a posteriori da Engenharia Didática.

Quadro 51: Quadro resumo com principais dificuldades e problemas apresentados pelos participantes nos encontros.

Dificuldades identificadas
Utilização de letra minúscula em início de frases
Problemas relacionados à digitação
Problemas com ortografia e gramática
Utilização incorreta de comandos de entrada
Utilização incorreta dos comandos condicionais se, entao, senao e fim_se
Dificuldades na memorização de comandos
Problemas com os operadores relacionais \geq e \leq
Problemas na realização de operações aritméticas
Problemas na declaração de variáveis
Utilização de função incorreta na atribuição do valor de uma variável
Ocorrência de inversão dos termos da sentença na atribuição do resultado de uma operação a uma variável
Problemas com o operador lógico "não"
Problemas na utilização de operadores aritméticos (uso de palavras ao invés dos símbolos)
Não utilização de vírgulas para separar variáveis na etapa de declaração
Utilização de variáveis para caracterizar valores constantes

Não utilização do comando fim para finalizar um programa
Tempo insuficiente para realização das atividades de avaliação
Dificuldade na compreensão da devolutiva do programa no momento de teste do código

Fonte: elaborado pelo autor (2024).

Quadro 52: Quadro resumo com principais potencialidades quanto ao uso do ILA identificadas ao longo dos encontros.

Potencialidades identificadas
Existência de comandos em língua portuguesa
Desenvolvimento de habilidades de modelagem matemática
Curiosidade em relação às funcionalidades do ILA
Desenvolvimento de habilidades metacognitivas através da reflexão sobre o processo de construção do programa
Desenvolvimento de habilidades metacognitivas através da realização de testes para avaliar o funcionamento do programa
Desenvolvimento de habilidades de programação de computadores
Desenvolvimento de habilidades do pensamento computacional
Retomada de conceitos desenvolvidos na disciplina de matemática
Desenvolvimento de habilidades de conversão entre diferentes sistemas de representação semiótica
Desenvolvimento da colaboração entre os participantes
Desenvolvimento de habilidades quanto ao uso do teclado do computador
Desenvolvimento de habilidades quanto ao uso do leitor de tela
Sentimento de satisfação ao solucionar um problema proposto

Fonte: elaborado pelo autor (2024).

6.2. A ANÁLISE DE CONTEÚDO

Na seção anterior foram analisadas falas, ações e produções dos participantes, elaboradas como resposta a um conjunto de solicitações e questionamentos previamente estabelecidos. No entanto, como é característico de situações de ensino e aprendizagem, os encontros conduzidos proporcionaram momentos ricos de interação entre os participantes e o pesquisador, constituindo um percurso formativo complexo que não pode ser mensurado satisfatoriamente a partir de uma simples avaliação quantitativa. É natural supor que, ao longo dos encontros, foram produzidos conhecimentos, habilidades e afetos que, em um primeiro momento, talvez não sejam identificáveis sem um olhar atento. De modo análogo, podem ter sido geradas dúvidas e inquietações cujos significados, para serem extraídos, demandem uma análise mais cuidadosa.

Levando em conta a necessidade de análise das entrelinhas do que foi vivenciado ao longo dos encontros da sequência didática, optou-se pela realização da análise de conteúdo do material textual gerado a partir da transcrição dos áudios dos encontros. Nesse contexto, conforme mencionado no referencial metodológico deste trabalho, a análise de conteúdo se organizou em torno de três etapas. A primeira fase consistiu na pré-análise, que se caracteriza por um período de intuições e sistematização inicial de ideias. Como coloca Bardin (1977, p.121), nesta etapa são feitas “a escolha dos documentos a serem submetidos à análise, a formulação das hipóteses e dos objetivos e a elaboração de indicadores que fundamentem a interpretação final”. Ainda segundo a autora, embora esta fase seja composta por atividades não estruturadas e abertas em lugar de uma exploração sistemática dos documentos, o objetivo é a organização da análise.

No contexto desta pesquisa, os documentos submetidos à análise foram as transcrições dos áudios dos encontros da sequência didática. Bardin (1977) sugere que, definidos os documentos, a primeira atividade em relação aos mesmos consiste em uma leitura flutuante do material, estabelecendo um primeiro contato com os dados e possibilitando a formulação de impressões e orientações iniciais. Neste caso, a leitura flutuante do material foi realizada em conjunto com o processo de revisão das transcrições, inicialmente realizadas por meio do *software Transkriptor*. A necessidade de revisão do material se deu por conta da existência de inconsistências ao longo dos

textos gerados, especialmente em virtude da natureza muitas vezes informal dos diálogos ao longo dos encontros. Já as hipóteses e objetivos adotados para análise de conteúdo se relacionam com as hipóteses de trabalho e o objetivo geral da pesquisa explicitados em seções anteriores deste trabalho.

Dando seguimento à pré-análise, Bardin (1977) coloca que, demarcado o gênero de documentos sobre os quais se pode efetuar a análise, pode ser necessária a constituição de um *corpus textual*. Este, por sua vez, se trata do conjunto de documentos a serem submetidos aos procedimentos analíticos. No entanto, tal constituição implica que sejam satisfeitas três regras principais: a regra da exaustividade, a regra da representatividade e a regra de pertinência:

- Regra da exaustividade: uma vez definido o campo do *corpus* (entrevistas de um inquérito, respostas a um questionário, editoriais de um diário de Paris entre tal e tal data, emissões de televisão sobre determinado assunto, etc.), é preciso ter-se em conta todos os elementos desse corpus (...). Esta regra é complementada pela de *não-selectividade*. (...)
- Regra da representatividade: a análise pode efectuar-se numa *amostra* desde que o material a isso se preste. A amostragem diz-se rigorosa se a amostra for uma parte representativa do universo inicial. Neste caso, os resultados obtidos para a amostra serão generalizados ao todo. (...)
- Regra da homogeneidade: os documentos retidos devem ser homogêneos, quer dizer, devem obedecer a critérios precisos de escolha e não representar demasiada singularidade fora destes critérios de escolha. (...)
- Regra de pertinência: os documentos retidos devem ser adequados, enquanto fonte de informação, de modo a corresponderem ao objetivo que suscita a análise (Bardin, p.122-124).

Como mencionado anteriormente, o corpus textual submetido à análise foi o conjunto de transcrições dos encontros. Considerando que todos os encontros (cinco para desenvolvimento de conceitos, dois para avaliação e um para condução de entrevista) realizados com todos os grupos foram levados em conta para a análise, considera-se que a regra da exaustividade foi satisfeita. Da mesma forma, não foi necessário recorrer a nenhum tipo de amostragem, na medida em que os dados de todos os participantes que optaram por participar da pesquisa foram utilizados em sua completude. Assim, a regra da representatividade também foi contemplada.

No que se refere à homogeneidade dos dados, ressalta-se que todos os grupos passaram por encontros que seguiram o mesmo planejamento. Dessa forma, todos os participantes foram expostos a estímulos similares e realizaram as mesmas produções quando solicitados. Assim, considera-se que a regra da homogeneidade foi satisfeita.

Por fim, entendemos que as transcrições dos encontros constituem documentos adequados enquanto fonte de informação, considerando o objetivo da análise. Isto porque tais documentos evidenciam elementos importantes acerca das potencialidades e limitações do ILA como recurso para desenvolvimento do pensamento computacional e resolução de problemas matemáticos através da programação de computadores. Assim, considera-se que a regra de pertinência também foi contemplada.

Por fim, no que se refere à última etapa da pré-análise, representada pela elaboração de indicadores que fundamentem a interpretação final, optou-se pela utilização de um recurso computacional. Como coloca Bardin (1977), a depender do material analisado, o procedimento de análise dos dados pode ser fastidioso e, em alguns casos, inviável se realizado manualmente. No contexto da pesquisa, as transcrições dos áudios dos encontros geraram um corpus textual denso e volumoso, totalizando mais de 400 páginas de falas a serem analisadas, o que inviabilizaria uma avaliação estritamente manual. Desse modo, o uso de um *software* específico para o gerenciamento de dados qualitativos facilitaria o processo de análise e, além disso, promoveria maior rigor científico à identificação e análise de padrões, na medida em que possibilitaria a quantificação e o emprego de cálculos estatísticos sobre as variáveis qualitativas subjetivas analisadas (Salvador et al., 2018).

Nesse cenário, optou-se pelo uso de um *software* para a análise lexicográfica dos dados antes da realização da análise de conteúdo em si. Mais especificamente, foi utilizado o IRaMuTeQ (Interface de R pour les Analyses Multidimensionnelles de Textes et de Questionnaires), *software* gratuito de código fonte aberto, desenvolvido sob a lógica *open source* e licenciado por GNU GPL (GNU General Public License). Inicialmente construído em língua francesa por Pierre Ratinaud, em 2009, o recurso possui atualmente dicionários completos em outras línguas e, como outros *softwares* de fonte aberta, possibilita alteração e expansão por meio da linguagem Python. Em termos operacionais, o *software* utiliza o ambiente estatístico do *software* R para realização de cálculos, e se vale do mesmo algoritmo do Alceste para condução de análises estatísticas de textos, incorporando, além da CHD (Classificação Hierárquica Descendente), outras análises lexicais que auxiliam na análise e interpretação de textos (Bueno, 2018; Salvador et al., 2018; Salviati, 2017).

6.2.1. ANÁLISE LEXICOGRÁFICA NO IRAMUTEQ

Na medida em que o IRaMuTeQ é baseado no *software* ALCESTE, seu funcionamento, de forma análoga, parte da análise da distribuição do vocabulário dentro de um material textual. Desse modo, para além da determinação do significado de um texto, o recurso é útil para analisar a organização dos elementos que compõem cada texto, estabelecendo relações entre a composição dos diferentes textos que formam o conjunto analisado (Dalud-Vincent, 2011).

O material de análise, referido como *corpus textual*, é composto por um conjunto de textos reunidos em um único arquivo de acordo com o objetivo da análise. Como coloca Dalud-Vincent (2011), para a obtenção de um resultado confiável utilizando esse tipo de algoritmo (ALCESTE, IRaMuTeQ), o corpus precisa satisfazer a duas condições: compor um todo coerente, seja em termos temáticos, seja no que se refere às condições de produção; deve ser suficientemente volumoso, de modo que as análises estatísticas sejam válidas. Entende-se que, no contexto deste trabalho, tais condições são satisfeitas, na medida em que o material de análise é composto por transcrições de áudios de situações de aprendizagem com mesma temática, conduzidas sob condições semelhantes.

Para atender a aspectos operacionais do *software*, é necessária uma preparação prévia do corpus de análise. Assim, deve ser realizada uma série de ajustes no conjunto de textos a ser analisado, de modo que os dados sejam disponibilizados em um formato adequado para tratamento. Em primeiro lugar, o corpus deve consistir em um único arquivo, no qual cada texto deve ser separado e identificado por uma linha de comando que, por sua vez, deve ser iniciada por quatro asteriscos (****). Além disso, tais linhas de comando podem apresentar variáveis específicas de acordo com o objetivo da análise, de modo que cada variável utilizada é introduzida por um asterisco (*) dentro da linha. Posteriormente, podem ser conduzidas análises específicas com o objetivo de identificar padrões relacionados às variáveis definidas nas linhas de comando.

No contexto desta pesquisa, como seria interessante analisar a existência de padrões intragrupo ou, ainda, a ocorrência de especificidades de acordo com os subtemas desenvolvidos em cada encontro, foram definidas duas variáveis de análise: grupo e encontro. Desse modo, cada texto consistia na transcrição de um encontro de um determinado grupo. Assim, para organização do corpus, em primeiro lugar as

transcrições dos encontros da sequência foram codificadas de acordo com o grupo e a ordem do encontro no desenvolvimento da sequência. Os códigos atribuídos para representar cada texto (ou seja, cada transcrição) são exibidos no quadro.

Quadro 53: Codificação dos textos no IRaMuTeQ.

	Grupo 1	Grupo 2	Grupo 3	Grupo 4
Encontro 1	*Grupo_1 *Encontro_1	*Grupo_2 *Encontro_1	*Grupo_3 *Encontro_1	*Grupo_4 *Encontro_1
Encontro 2	*Grupo_1 *Encontro_2	*Grupo_2 *Encontro_2	*Grupo_3 *Encontro_2	*Grupo_4 *Encontro_2
Encontro 3	*Grupo_1 *Encontro_3	*Grupo_2 *Encontro_3	*Grupo_3 *Encontro_3	*Grupo_4 *Encontro_3
Encontro 4	*Grupo_1 *Encontro_4	*Grupo_2 *Encontro_4	*Grupo_3 *Encontro_4	*Grupo_4 *Encontro_4
Encontro 5	*Grupo_1 *Encontro_5	*Grupo_2 *Encontro_5	*Grupo_3 *Encontro_5	*Grupo_4 *Encontro_5
Avaliação 1	*Grupo_1 *Encontro_6	*Grupo_2 *Encontro_6	*Grupo_3 *Encontro_6	*Grupo_4 *Encontro_6
Avaliação 2	*Grupo_1 *Encontro_7	*Grupo_2 *Encontro_7	*Grupo_3 *Encontro_7	*Grupo_4 *Encontro_7
Entrevista	*Grupo_1 *Encontro_8	*Grupo_2 *Encontro_8	*Grupo_3 *Encontro_8	*Grupo_4 *Encontro_8

Fonte: elaborado pelo autor (2024).

Em seguida, foi necessária uma adequação do arquivo utilizado, bem como de alguns termos presentes no texto, de acordo com orientações relacionadas às funcionalidades do *software*. Nesse sentido, foram realizadas as seguintes adaptações:

- Utilização de um arquivo no formato txt (sem recuo de parágrafo, margens ou tabulações de texto) usando o padrão de codificação UTF-8 (*UCS Transformation Format 8*), mais adequado para evitar problemas de compatibilidade;
- Eliminação de todas as falas do pesquisador, na medida em que se deseja analisar padrões nas falas e comportamentos dos participantes;
- Eliminação de apóstrofes, asteriscos, aspas e reticências;
- Eliminação de expressões desnecessárias à análise, como “aham”, “uhum” e “né”;
- Utilização do símbolo “_” para representação de palavras compostas ou expressões, como “guarda_chuva”, “com_certeza” ou “meu_Deus”.

Por fim, o corpus foi importado para o *software* para que se pudesse proceder à análise léxica do material.

Nota-se que a palavra ativa mais presente no corpus textual é o termo “número”. Tal resultado é condizente com o teor da pesquisa, uma vez que na maior parte dos encontros o objetivo final era a construção de um ou mais programas para a resolução de problemas de matemática. Termos como “escrever”, “digitar”, “ler”, “variável”, “ila”, “programa” e “numérico” também aparecem em destaque, o que está de acordo com o esperado, na medida em que representam comandos ou palavras bastante utilizadas na construção dos programas ao longo dos encontros. Cabe ressaltar, ainda, o destaque que se evidencia para os termos “pergunta_se_está_correta”, criado para abranger expressões como “é isso?”, “tá certo” e “não é?”, entre outros, e “participante_sorri”, utilizado em momentos em que o participante sorria por algum motivo. Tal fato pode indicar a ocorrência frequente de, por um lado, momentos de dúvidas e inseguranças dos participantes e, por outro, momentos de alegria e descontração ao longo dos encontros. Finalmente, verbos como “dar”, “colocar”, “botar”, “falar”, “deixar”, “achar”, entre outros termos, ganham evidência por serem utilizados com frequência na oralidade, embora não apresentem destaque considerando os objetivos da pesquisa.

Em seguida, com o intuito de identificar possíveis particularidades na frequência de palavras em algum dos grupos de estudantes ou, ainda, nos encontros onde alguma temática específica era abordada, foram construídas nuvens de palavras utilizando-se as variáveis definidas no IRaMuTeQ (grupo e encontro). Desse modo, as nuvens apresentavam representações específicas de um determinado grupo ao longo dos encontros, ou de um determinado encontro específico da sequência didática em todos os grupos. Os resultados podem ser encontrados no Apêndice E.

No que se refere aos grupos, não foram encontradas diferenças significativas, o que indica um comportamento similar dos estudantes nos diferentes grupos em termos de uso da linguagem. Já quanto aos encontros, é possível observar algumas variações. No primeiro encontro, momento em que foram desenvolvidas as noções de entrada e saída de dados, a palavra “número” não ganha tanto destaque quanto nos demais encontros. Nesse caso, é possível verificar maior frequência de comandos como “escrever”, “ler”, “variável”, além de verbos e outras palavras bastante utilizadas no contexto do encontro, como “digitar”, “colocar”, “idade” e “apagar”.

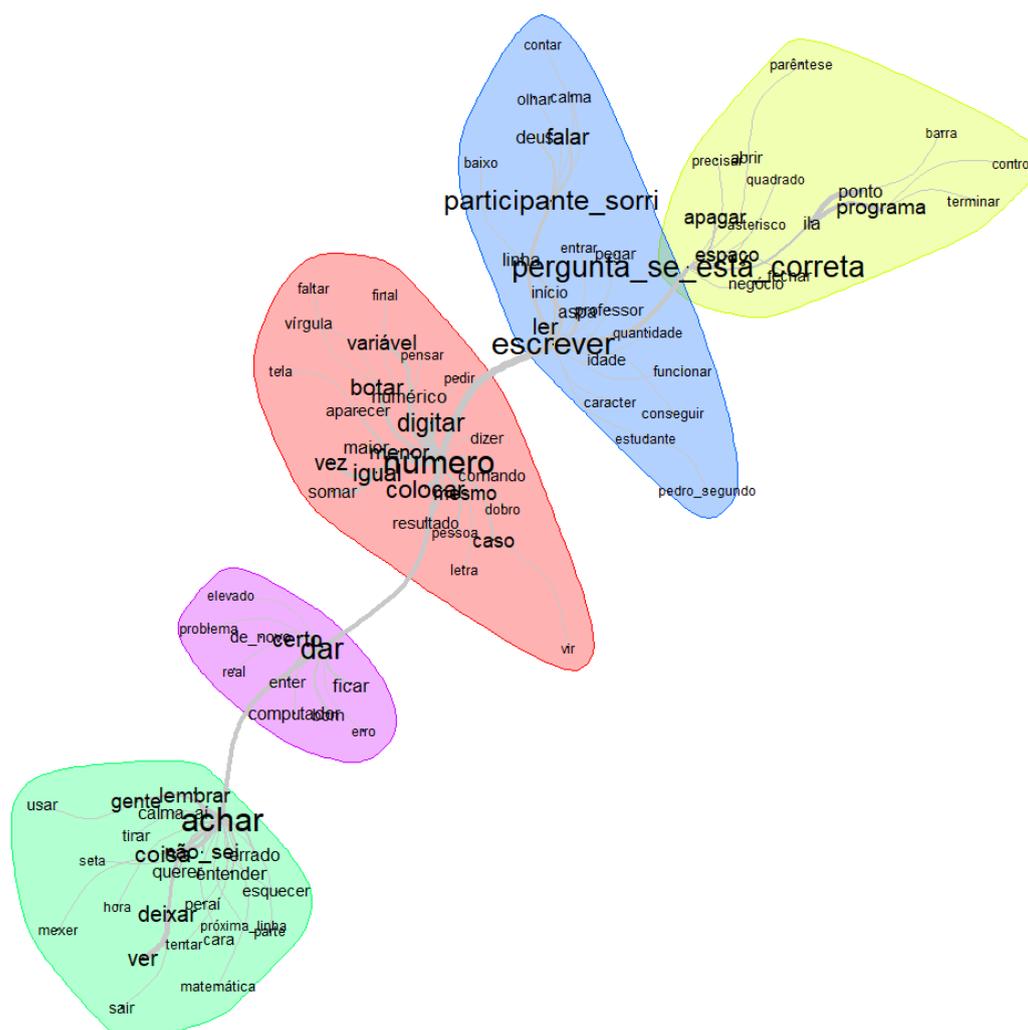
Por sua vez, a palavra “número” ganha maior destaque nos encontros 2, 3, 4 e 5. No segundo encontro o termo “somar” também aparece com frequência elevada, o que pode ser explicado pelo fato de ter sido construído um programa que realizava uma

operação de soma entre dois números. Já nos encontros 5 e 6, é possível identificar grande frequência no uso de operadores relacionais (maior, menor, igual). Tal fato se justifica por se tratarem de momentos em que foram desenvolvidos conceitos relacionados à programação condicional, envolvendo o uso de tais operadores no estabelecimento das condições a serem verificadas no programa. Por fim, nos encontros 7 e 8 o vocábulo “achar” aparece com frequência destacada, o que pode ser consequência de dúvidas e inseguranças expressas na avaliação sem fonte de consulta, ou de ideias e opiniões pessoais manifestadas na entrevista final.

Análise de similitude

A análise de similitude baseia-se na teoria dos grafos, e possibilita identificar coocorrências entre as palavras em um material textual (Camargo; Justo, 2018). Desse modo, através de uma representação gráfica no formato de árvore, pode-se identificar conexões entre palavras frequentemente utilizadas de forma conjunta no texto, auxiliando na identificação da estrutura do conteúdo do corpus textual. Aqui, a análise foi realizada, inicialmente, considerando-se todas as palavras existentes no corpus, o que gerou uma árvore excessivamente densa, impossibilitando a leitura dos termos e a identificação de relações. Desse modo, optou-se, em um segundo momento, pela construção da árvore utilizando-se apenas as 100 palavras com maior frequência no corpus textual, excetuando-se vocábulos sem significado expressivo, como “ok”, “né”, “ô”, “só” e “tô”. O resultado é apresentado na Figura 48.

Figura 48: Resultado da análise de similitude.



Fonte: elaborado pelo autor (2024).

Algumas inferências podem ser realizadas a partir da análise da Figura 48. Em primeiro lugar, observa-se que foram identificadas cinco comunidades de palavras que se ramificam em torno dos vocábulos “número”, “dar”, “achar”, “escrever”, e do termo “pergunta_se_está_correta”. Observa-se, ainda, que a disposição das comunidades tem como grupo central aquele que se ramifica em torno da palavra “número”. Tal resultado está de acordo com o que se observa na nuvem de palavras apresentada anteriormente.

Em primeiro lugar, no grupo central, em torno da palavra “número”, observam-se termos como “maior”, “menor”, “igual”, “somar” e “resultado”, normalmente utilizados em sentenças, relações e operações aritméticas. O vocábulo “vez” poderia, nesse caso, representar uma lematização da palavra “vezes”, utilizada comumente na representação

de multiplicações. No mesmo grupo se encontram palavras como “variável”, “numérico”, “digitar” e “comando”, bastante utilizadas na declaração de variáveis e, de maneira mais geral, na construção dos programas ao longo dos encontros. Mais especificamente, uma ramificação que merece destaque neste grupo é aquela onde se observa a palavra “vírgula”, que aparece diretamente relacionada aos termos “faltar” e “botar”. Tal fato pode indicar alguma dificuldade dos participantes no que se refere à inserção de vírgulas nos códigos, quando necessário.

A comunidade que tem o vocábulo central “escrever” apresenta, como esperado, algumas das palavras que foram utilizadas em frases de saída na construção dos programas ao longo dos encontros. Termos como “quantidade”, “idade”, “estudante” e “pedro_segundo” se inserem nesse contexto. Nota-se que o vocábulo “ler” também se relaciona com estas palavras, fato esperado na medida em que as mesmas também foram muitas vezes utilizadas em contextos de entrada de dados.

Um ponto que merece atenção é o fato do termo “pergunta_se_está_correta” ser central em uma das comunidades. Nota-se, ainda, que vocábulos que representam teclas do teclado do computador também se encontram nesse bloco de palavras, como “asterisco”, “ponto”, “barra”, “espaço”, “control” e “parêntese”. Tal fato pode indicar a ocorrência frequente de dúvidas relacionadas à digitação ou à localização de teclas. Observando, ainda, a proximidade de termos como “ila”, “programa”, “espaço”, “barra” e “abrir”, a maioria na mesma ramificação dentro da comunidade, pode-se inferir que também possa ter havido problemas relacionados ao procedimento de abertura e execução dos programas no ILA.

Já na comunidade que se dispõe em torno do vocábulo “dar”, é possível identificar vocábulos como “erro”, “certo”, “problema”, e “de_novo”, o que pode indicar momentos de avaliação e teste dos programas concluídos. A existência do termo “enter” na comunidade é natural, na medida em que é comum o uso da expressão “dar enter” para indicar a ação de clicar nesta tecla.

Por fim, a comunidade que se ramifica em torno do vocábulo “achar” também pode estar relacionada com momentos onde os participantes exibiram dúvidas ou incertezas, na medida em que no mesmo grupo se encontram vocábulos como “lembrar”, “esquecer”, “entender”, “não_sei”, “calma_á” e “peraí”. Como tal comunidade se relaciona diretamente com aquela que se organiza em torno do vocábulo “dar”,

mencionada anteriormente, uma possível inferência é a existência de dúvidas e incertezas relacionadas ao processo de avaliação e teste dos programas.

Classificação hierárquica descendente

De acordo com Camargo e Justo (2013, p.516), o método da CHD, proposto por Reinert (1990), “classifica os segmentos de texto em função dos seus respectivos vocabulários, e o conjunto deles é repartido com base na frequência das formas reduzidas”. Por sua vez, tais formas reduzidas são obtidas através de um processo de lematização, por meio do qual é encontrada uma forma representativa de todas as formas que uma palavra pode tomar (variações de gênero e número, flexões verbais, entre outros). Conforme coloca Sousa (2021, p. 1548), isso tem como objetivo “reduzir a variabilidade do vocabulário, o que permite conferir maior homogeneidade ao material que será submetido às diferentes técnicas de tratamento lexicométrico”.

No caso do IRaMuTeQ, a CHD é utilizada para obter classes de segmentos de texto que, ao mesmo tempo, apresentam vocabulário semelhante entre si e diferente daqueles das outras classes. A partir de matrizes relacionando segmentos de textos e palavras, são realizados repetidos testes do tipo χ^2 (Qui-Quadrado), usados para avaliar se há associação entre diferentes variáveis, obtendo-se, enfim, uma classificação definitiva. O *software* organiza, então, um dendograma, tipo de representação gráfica útil na análise de agrupamentos, que ilustra as relações identificadas entre as classes. Por fim, com base nos resultados retornados pelo programa, é possível analisar cada uma das classes identificadas, principalmente a partir de seu vocabulário característico (Camargo e Justo, 2018).

Nesta pesquisa, a Classificação Hierárquica Descendente foi realizada utilizando-se todo o corpus textual, com o objetivo de identificar possíveis classes de segmentos de texto com vocabulário característico. Conforme colocam Camargo e Justo (2018), para que análises do tipo CHD sejam úteis na classificação de um material textual, é necessária uma retenção mínima de 75% dos segmentos de texto. No contexto deste estudo, o percentual de retenção foi de 84,08%, o que representa um nível satisfatório de homogeneidade do material. Os dados quantitativos da análise, bem como o percentual de retenção, são exibidos na Figura 49.

Figura 49: : Dados quantitativos da Classificação Hierárquica Descendente no IRaMuTeQ.

```

+-+--+--+--+--+
|i|R|a|M|u|T|e|Q| - Thu Apr 18 19:07:28 2024
+-+--+--+--+--+

Number of texts: 32
Number of text segments: 1539
Number of forms: 2656
Number of occurrences: 52261
Número de lemas: 1750
Number of active forms: 1513
Número de formas suplementares: 224
Número de formas ativas com a frequência >= 3: 685
Média das formas por segmento: 33.957765
Number of clusters: 3
1294 segments classified on 1539 (84.08%)

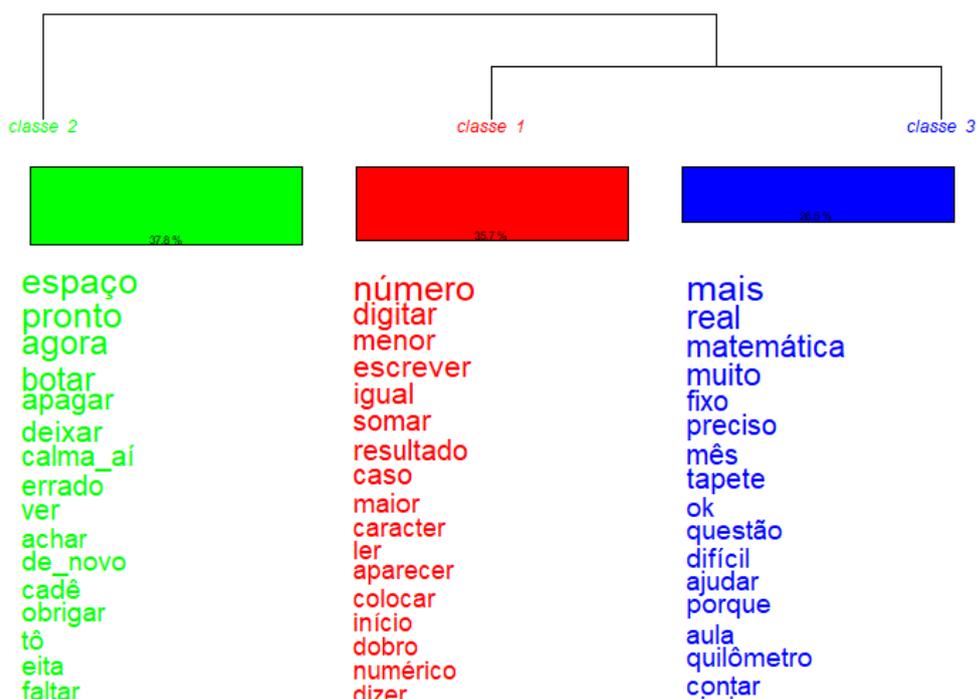
#####
tempo : 0h 0m 24s
#####

```

Fonte: elaborado pelo autor (2024).

O método da CHD forneceu um total de três classes de palavras, que são ilustradas no dendograma da Figura 50. Nota-se que, nos 32 textos do corpus textual, foram identificados 1539 segmentos de texto com base no tamanho de seus enunciados e na pontuação. Além disso, o material textual é composto por um total de 2656 formas léxicas, configurando 1750 lemas após o processo de lematização. Observa-se, ainda, um total de 1513 formas ativas (verbos, substantivos e adjetivos), e 224 formas suplementares (pronomes, adjetivos e advérbios, entre outros). Essa diferenciação é importante na medida em que as palavras mais relevantes nas análises são as ativas pelo fato de indicarem características semânticas dos enunciados (Sousa, 2021).

Figura 50: Dendograma da Classificação Hierárquica Descendente.



Fonte: elaborado pelo autor (2024).

A classe 1, na qual a palavra “número” é a que aparece com maior frequência, apresenta comandos normalmente utilizados na construção dos programas, bem como termos comuns no estabelecimento de relações e sentenças matemáticas, também amplamente utilizadas nos códigos ao longo dos encontros. Nesse contexto se inserem os comandos de entrada e saída “ler” e “escrever”, tipos de variáveis como “caracter” e “numérico”, operadores relacionais como “igual”, “maior” e “menor”, e outros termos comumente utilizados em sentenças matemáticas e operações aritméticas, como “somar”, “resultado” e “dobro”.

Na Figura 51 são apresentadas as palavras com maior aderência à classe. A coluna “eff. s.t.” representa o número de segmentos de texto na classe que contêm a palavra. Já a coluna “”eff. total” representa o número de segmentos de texto no corpus que contêm, ao menos uma vez, a referida palavra. A coluna “pourcentage” denota a porcentagem de ocorrência da palavra nos segmentos de texto nesta classe, em relação às ocorrências no corpus inteiro. A coluna “chi2” representa o qui-quadrado de associação da palavra com a classe (quanto maior o qui-quadrado, mais provável é a hipótese de associação entre a palavra e a classe). Por fim, a coluna “Type” representa a classe gramatical à qual a palavra foi identificada.

Figura 51: Palavras identificadas na classe 1 da CHD.

n...	eff. s.t.	eff. total	pourcentage	chi2	Type	forme
0	216	310	69.68	204.98	nom	número
1	135	185	72.97	130.61	ver	digitar
2	99	130	76.15	103.01	adj	menor
3	173	288	60.07	95.81	ver	escrever
4	127	189	67.2	95.62	adj	igual
5	69	86	80.23	79.57	ver	somar
6	54	63	85.71	72.15	nom	resultado
7	80	111	72.07	69.96	nom	caso
8	76	111	68.47	56.78	adj	maior
9	25	26	96.15	42.24	nr	caracter
10	94	166	56.63	36.32	ver	ler
11	42	58	72.41	35.65	ver	aparecer
12	96	174	55.17	33.19	ver	colocar
13	40	57	70.18	30.87	nom	início
14	19	22	86.36	25.02	nom	dobro
15	53	89	59.55	23.68	adj	numérico
16	30	43	69.77	22.48	ver	dizer
17	15	17	88.24	20.71	nom	saída
18	50	87	57.47	19.25	adv	depois
19	21	28	75.0	19.25	nom	tela
20	66	124	53.23	18.34	adj	variável

Fonte: elaborado pelo autor (2024).

Figura 52: Segmentos de texto onde são destacadas associações entre vocábulos alocados na classe 1.

Concordância - Classe 1

**** *Grupo_3 *Encontro_1

aí eu **digito** em forma de nome mesmo em palavra ou em **número** ah porque eu **digitei** no **início numérico** hum enter deu entendi sim ele a gente no **início colocou** a **variável** pra **ler** a idade

**** *Grupo_3 *Encontro_1

e **depois colocou numérico** mudando o nome e **depois** ele **leu colocou digite** sua idade a gente **digitou** a idade aí deu tudo certinho a idade do usuário é esta hum **escrever número**

**** *Grupo_3 *Encontro_1

a idade do usuário é esta um **número** aleatório que a pessoa **digitou** então **depois** de **escrever** não necessariamente precisa **colocar número** e tem sempre que **escrever variáveis** pergunta_se_está_correta enter pergunta_se_está_correta **digite** sua idade

**** *Grupo_4 *Encontro_3

vai dar certo esse apagar aqui nova linha **ler** a **escrever digite** um **número início número igual** a d **variáveis variáveis** o d e f participante_sorri ves veis s **variáveis** não tá **lendo** não

**** *Grupo_4 *Encontro_4

então aí é pra linha de baixo enter então aí tá é a **igual** a 10 eu tenho que **escrever** o **número digitado igual** a 10 ah **escrever escrever escrever número digitado** é isso

**** *Grupo_4 *Encontro_4

não **escrever** se a é **igual** a 10 não participante_sorri que o **número** que o **número é igual** a 10 que 10 é **igual** a 10 é **igual** a 10 **número igual** a 10

**** *Grupo_3 *Encontro_2

é barra aonde é que o meu tem uma barrinha do_lado da seta por isso que eu me confundi aqui aqui aqui ah ah sim hum é raiz o **número** elevado a ele mesmo

**** *Grupo_1 *Encontro_6

início enter aí eu tenho que começar com o comando é ele quer **ler** não **escrever escrever** tá ná não voltar ao mesmo **número** aqui **coloca** só o k aqui por favor quilômetros eu erreí foi quantos quilômetros

**** *Grupo_3 *Encontro_2

Fonte: elaborado pelo autor (2024).

A Figura 52, por sua vez, representa alguns segmentos de texto onde são destacadas as associações entre palavras da classe 1. Percebe-se que, em geral, o vocábulo número aparece associado com comandos de entrada e saída, ou verbos como “digitar” e “colocar”. A existência de palavras como “variável”, “numérico” e “caracter” remete, ainda, a situações de declaração de variáveis. Uma inferência possível, então, é que a classe represente momentos iniciais da construção do código, onde são definidas as variáveis e seus tipos, bem como são solicitadas e armazenadas informações.

Já a classe 2 apresenta termos normalmente utilizados para exprimir dúvidas ou sentimentos negativos, como “achar”, “apagar”, “errado”, “faltar”, “obrigar”, “eita” e “cadê”. Embora se trate de uma classe com termos mais heterogêneos que a anterior, pode-se inferir que os contextos onde os mesmos foram utilizados estejam relacionados a momentos de dúvida ou incerteza. Na Figura 53 são exibidos os vocábulos com maior aderência à classe.

Figura 53: Palavras identificadas na classe 2 da CHD.

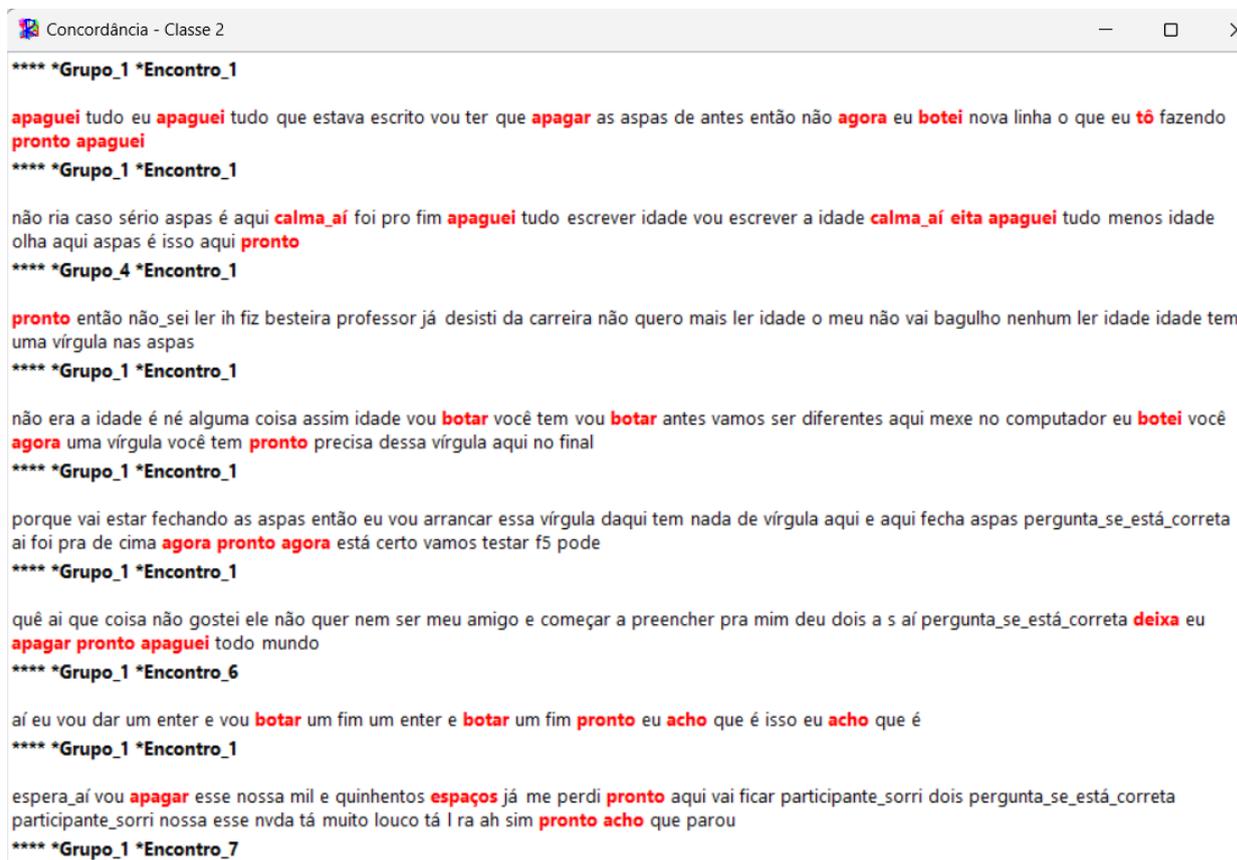
n...	↑	eff. s.t.	eff. total	pourcentage	chi2	Type	forme
0		73	80	91.25	103.66	nom	espaço
1		76	86	88.37	100.26	adv	pronto
2		114	158	72.15	90.39	adv	agora
3		109	153	71.24	82.59	ver	botar
4		86	122	70.49	61.27	ver	apagar
5		96	143	67.13	58.88	ver	deixar
6		62	87	71.26	44.46	nr	calma_ái
7		52	69	75.36	43.77	adj	errado
8		83	131	63.36	40.53	ver	ver
9		158	305	51.8	33.33	ver	achar
10		41	55	74.55	33.01	nr	de_novo
11		28	33	84.85	31.9	adv	cadê
12		18	18	100.0	30.05	ver	obrigar
13		102	182	56.04	30.02	nr	tô
14		21	23	91.3	28.52	nr	eita
15		24	28	85.71	27.96	ver	faltar
16		19	21	90.48	25.2	ver	salvar
17		15	15	100.0	24.98	nom	fulano
18		17	18	94.44	24.92	adv	cá
19		17	18	94.44	24.92	ver	consertar
20		76	134	56.72	22.78	adj	certo

Fonte: elaborado pelo autor (2024).

Na Figura 54 são destacadas algumas associações entre vocábulos da classe 2 em segmentos de texto. Como mencionado anteriormente, podem ser observados momentos de dúvida (“*alguma coisa assim*”, “*o que eu tô fazendo*”, “*eu acho que é isso*”),

cometimento ou correção de erros (“*vou ter que apagar as aspas*”, “*apaguei tudo que estava escrito*”, “*pronto apaguei todo mundo*”), e sentimentos negativos (“*ih fiz besteira professor já desisti da carreira*”, “*não gostei ele não quer nem ser meu amigo*”).

Figura 54: Segmentos de texto onde são destacadas associações entre vocábulos alocados na classe 2.



Fonte: elaborado pelo autor (2024).

A classe 3, por sua vez, apresenta termos normalmente utilizados no contexto dos problemas de matemática resolvidos pelos estudantes. Nesta classe, na qual “mais” é o termo com maior frequência, se verificam, também, palavras como “matemática”, “fixo”, “mês”, “tapete”, “questão”, “quilômetro” e “aula”, vocábulos que estiverem presentes nos problemas resolvidos ao longo dos encontros. Na Figura 55 são exibidos os termos com maior aderência à classe. Já na Figura 56 são representadas algumas associações entre vocábulos da classe em diferentes segmentos de texto. É possível observar com frequência a utilização de associações das palavras alocadas na classe em situações que remetem à resolução de problemas.

Figura 55: Palavras identificadas na classe 3 da CHD.

n...	↑	eff. s.t.	eff. total	pourcentage	chi2	Type	forme
0		125	244	51.23	94.34	adv	mais
1		31	35	88.57	71.13	adj	real
2		25	27	92.59	61.82	nom	matemática
3		41	59	69.49	58.63	adv	muito
4		17	18	94.44	43.25	adj	fixo
5		16	17	94.12	40.42	adj	preciso
6		16	17	94.12	40.42	nom	mês
7		14	14	100.0	39.24	nom	tapete
8		53	101	52.48	37.92	nr	ok
9		15	16	93.75	37.6	nom	questão
10		15	16	93.75	37.6	adj	difícil
11		16	18	88.89	36.46	ver	ajudar
12		65	137	47.45	34.48	adv	porque
13		11	11	100.0	30.76	nom	aula
14		16	20	80.0	29.84	nom	quilômetro
15		19	26	73.08	29.54	ver	contar
16		13	15	86.67	28.19	nom	ajuda
17		10	10	100.0	27.94	nom	conta
18		10	10	100.0	27.94	ver	aprender
19		85	209	40.67	25.67	nr	participante_sorri
20		18	26	69.23	24.86	nom	quantidade

Fonte: elaborado pelo autor (2024).

Figura 56: Segmentos de texto onde são destacadas associações entre vocábulos alocados na classe 3.

Concordância - Classe 3
<p>**** *Grupo_1 *Encontro_6</p> <p>af eu fiz 2 vezes o que vai ficar 2 vezes o 3 que vai dar 6 mais 6 12 deu certinho eu atribui a variável a à quantidade de quilômetros</p> <p>**** *Grupo_1 *Encontro_6</p> <p>somado a isso a bandeirada que eu entendi lá era de 6 reais então tem que ser o valor a vezes a quantidade de quilômetros rodados vezes é mais o 6</p> <p>**** *Grupo_1 *Encontro_7</p> <p>situação 4 da situação ah tá sim eu estou lendo aqui meu deus se o senhor quisesse ler pra mim seria muito mais rápido não não eu que tô lendo por isso que tá demorando</p> <p>**** *Grupo_4 *Encontro_5</p> <p>ah que resolver o problema o quê eu vou resolver um problema ou eu vou criar um um ah não eu não sou boa em matemática olha só eu sou ruim eu sou ruim com 2 mais 2 você não tá entendendo</p> <p>**** *Grupo_4 *Encontro_5</p> <p>não não ok não eu não entendi participante_sorri não mas é porque eu não entendi isso 50 reais ah se ele usar mais de 40</p> <p>**** *Grupo_4 *Encontro_5</p> <p>af ele vai ai vai ser difícil isso não pera não era se ele usasse mais de 40 ele ia pagar os 50 mais 2 ah tá então se ele gasta se ele usar 1 mega af ele vai pagar 42 reais</p> <p>**** *Grupo_4 *Encontro_5</p> <p>sim ah se ele gastar mais af que o valor fixo muda vamos ver tem que criar do zero pergunta_se_está_correta ah variável variável não sim variável embaixo é numérico pergunta_se_está_correta você pode me ajudar</p> <p>**** *Grupo_4 *Encontro_3</p> <p>participante_sorri é 4 8 que dá 1 ah 4 3 mais 6 7 esse negócio af confunde com mais 1 8 6 7 isso confunde tá 8 9 mais ou menos</p> <p>**** *Grupo_1 *Encontro_7</p> <p>acho que era 1 não lembro 2 tá e mais a taxa fixa de 30 reais tá aí não vai dar nem tempo de testar né mas pronto escrever calma_af calma_af</p>

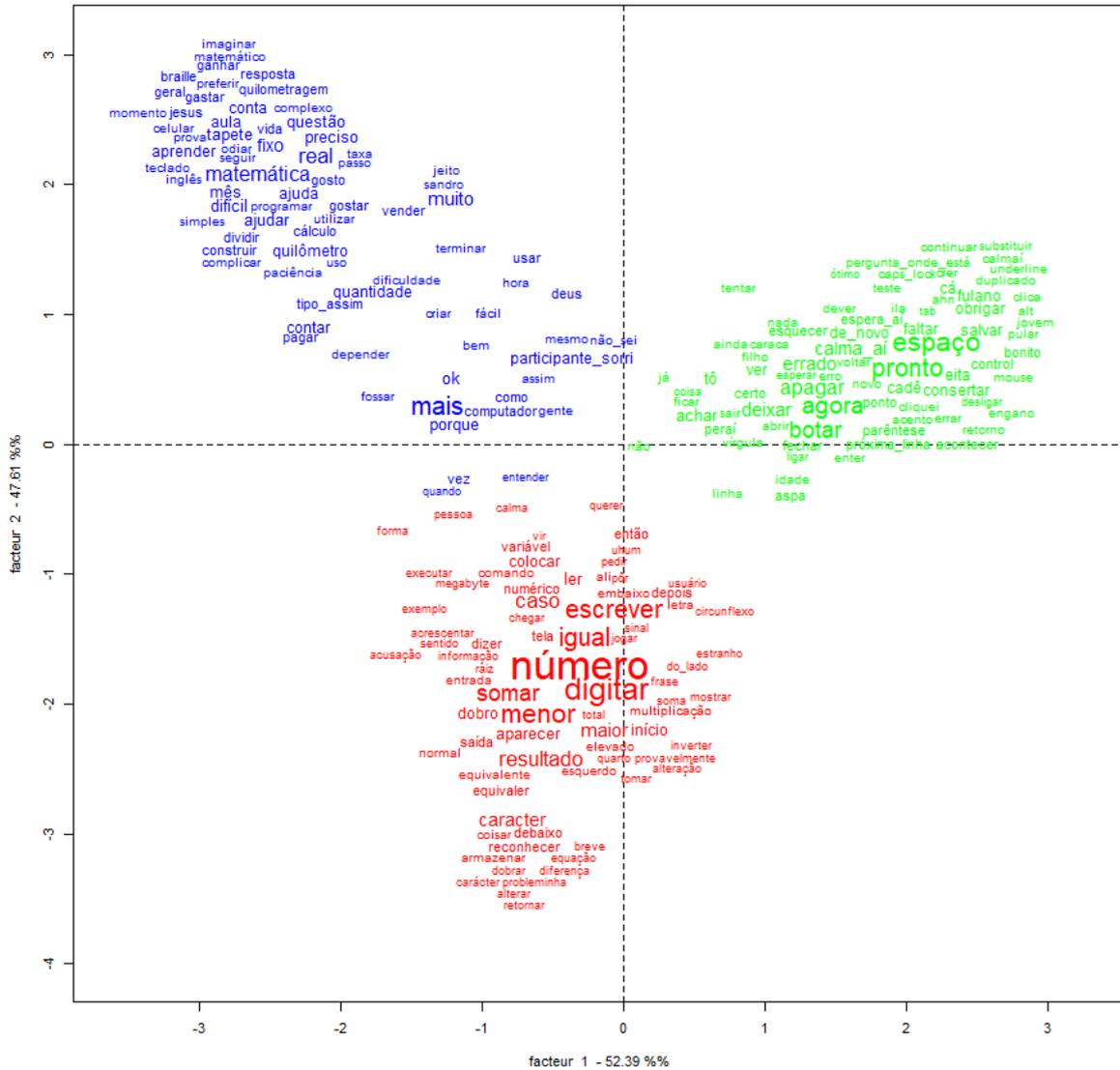
Fonte: elaborado pelo autor (2024).

No que se refere às relações identificadas entre as classes, nota-se que existe uma proximidade maior entre os vocábulos das classes 1 e 3. Uma inferência possível deste fato é que os comandos e operadores identificados entre os vocábulos da classe 1 foram utilizados nos códigos construídos para resolução dos problemas que apresentavam os vocábulos exibidos na classe 3. Nesse processo de construção, podem ter existido dúvidas e inseguranças que foram expressas pelos participantes a partir de um vocabulário característico da classe 2.

Por fim, na análise fatorial de correspondência encontrada na Figura 57 podem ser observadas mais palavras que se inserem nas três diferentes classes de palavras identificadas. Além disso, considerando o fator 1 da análise (eixo horizontal da figura), observa-se que os vocábulos da classe 1 e da classe 3 encontram-se no eixo negativo, o que sugere uma maior correspondência entre os vocábulos dessas classes, em contraposição aos da classe 2. Assim como mencionado anteriormente, uma possível justificativa para a aproximação das classes 1 e 3 reside no fato de que as mesmas se referem a momentos da construção do código em si: as fases iniciais (classe 1) e a mobilização de elementos para a resolução do problema (classe 3).

Já considerando o fator 2 (eixo vertical da figura), observa-se que a maior parte dos vocábulos da classe 2 e da classe 3 encontram-se no eixo positivo. Ou seja, nota-se uma maior correspondência entre os vocábulos dessas classes, em detrimento da classe 1. Uma inferência possível é que os participantes podem ter apresentado mais dúvidas, inseguranças e sentimentos negativos (classe 2) nos momentos de resolução dos problemas em si (classe 3), que nas fases iniciais de declaração de variáveis e solicitação e armazenamento de informações (classe 1).

Figura 57: Análise fatorial de correspondência da CHD.



Fonte: elaborado pelo autor (2024).

6.2.2. ANÁLISE CATEGORIAL

Como coloca Bardin (1977, p. 40), a análise de conteúdo representaria “um conjunto de técnicas de análise das comunicações que utiliza procedimentos sistemáticos e objetivos de descrição do conteúdo das mensagens”. No entanto, a própria autora ressalta que, para além de uma simples descrição dos conteúdos, o interesse da análise de conteúdo estaria em identificar o que estes poderiam revelar e ensinar após serem tratados relativamente a outras coisas. Assim, o objetivo final da

análise seria a “inferência de conhecimentos relativos às condições de produção (ou eventualmente, de recepção), inferência esta que recorre a indicadores (quantitativos ou não)” (Bardin, 1977, p.40).

A construção dos indicadores tomados como referência para a realização da análise pode se dar de acordo com diferentes técnicas e critérios de classificação. Segundo Bardin (1977), entre o conjunto das técnicas da análise de conteúdo, a análise categorial é a mais antiga e, em termos práticos, a mais utilizada. Ao utilizar esta técnica, o analista desmembra o texto em unidades, reagrupando-as de acordo com diferentes possibilidades de categorização.

Este tipo de análise, o mais generalizado e transmitido, foi cronologicamente o primeiro, podendo ser denominado análise categorial. Esta pretende tomar em consideração a totalidade de um «texto», passando-o pelo crivo da classificação e do recenseamento, segundo a frequência de presença (ou de ausência) de itens de sentido. Isso pode constituir um primeiro passo, obedecendo ao princípio de objectividade e racionalizando através de números e percentagem, uma interpretação que, sem ela, teria de ser sujeita a aval. É o método das categorias, espécie de gavetas ou rubricas significativas que permitem a classificação dos elementos de significação constitutivas, da mensagem. É portanto um método taxionómico bem concebido para satisfazer os colecionadores preocupados em introduzir uma ordem, segundo certos critérios, na desordem aparente (Bardin, 1977, p. 38).

De maneira geral, a análise categorial envolve a delimitação do texto em unidades de codificação, ou unidades de registro, que podem ser palavras, frases, minutos, ou outras unidades dotadas de sentido de acordo com o material analisado. De acordo com Bardin (1977), em alguns casos há, ainda, a necessidade de definição de unidades de contexto, que embora não sejam levadas em consideração no recenseamento de frequências, são superiores às unidades de codificação, permitindo compreender a significação dos itens obtidos ao repô-los no seu contexto dentro do material analisado.

Como colocam Sousa e Santos (2020), a repetição de palavras e/ou termos pode representar um ponto de partida no processo de codificação e criação de unidades de registro para o posterior desmembramento em categorias de análise. Nessa perspectiva, utilizamos a análise lexicográfica conduzida a partir do *software* IRaMuTeQ, bem como os resultados da análise a posteriori da Engenharia Didática, sintetizados no Quadro 51 e no Quadro 52, como elementos orientadores no processo de categorização do material analisado. Partindo desse referencial passou-se, então, à segunda fase da análise de conteúdo, a exploração do material. Esta etapa, conforme mencionam Capelle, Melo e

Gonçalves (2003), culmina com a codificação dos dados brutos do material, envolvendo procedimentos de recorte, contagem, definição e classificação.

Realizada a exploração do material, foram elencadas as principais dificuldades exibidas pelos estudantes ao longo dos encontros. Tais dificuldades foram reagrupadas em categorias, de modo que pudéssemos identificar, de maneira geral, os principais desafios para o desenvolvimento de conceitos e habilidades do pensamento computacional através da utilização de uma linguagem algorítmica de programação. As categorias definidas a partir de tal exploração são exibidas no Quadro 54.

Quadro 54: Categorias identificadas através da análise de conteúdo.

Categoria final	Categoria intermediária	Categoria inicial
C1: Problemas relacionados ao uso do teclado	C1A: Dificuldades quanto à localização de teclas no teclado	C1A1. Dificuldade na localização de operadores aritméticos (+, -, *, /)
		C1A2: Dificuldade na localização de operadores relacionais (>, <, =)
		C1A3. Dificuldade na localização de outros caracteres especiais que desempenham funções específicas (vírgula, aspas, parênteses...)
		C1A4: Dificuldade na localização de demais teclas
		C1A5: Dificuldade por conta da diferença na disposição de teclas em teclados diferentes
	C1B: Problemas relacionados à digitação	C1B1: Participante apaga algo involuntariamente
		C1B2: Participante insere ou deixa de inserir espaço entre caracteres equivocadamente
		C1B3: Participante digita alguma outra tecla equivocadamente (exceto espaço e apagar)
C1B4: Participante tem algum problema por conta de sensibilidade do sensor tátil do notebook (<i>touch pad</i>)		
C2: Problemas na identificação de elementos na tela do computador	C2A: Dificuldade na identificação da posição do cursor	C2A1: Dificuldade na localização da posição do cursor em uma linha
		C2A2: Dificuldade na identificação da linha em que o cursor está localizado
		C2A3: Dificuldade na identificação da posição do cursor na tela
C2B: Participante não identifica algum ocorrido na tela	C2B1: Participante não identifica alguma ocorrência na tela	
	C3A: Inconsistências na leitura de símbolos ou palavras	C3A1: Leitor de telas não faz a leitura de algum símbolo ou palavra

Categoria final	Categoria intermediária	Categoria inicial
C3: Problemas relacionados ao leitor de tela		C3A2: Leitor de telas faz a leitura de algum símbolo ou palavra de maneira não usual
	C3B: Problemas relacionados à voz ou à cadência na leitura do leitor de telas	C3B1: Participante não compreende a leitura do leitor de telas
		C3B3: Participante manifesta insatisfação com a voz ou a forma como o leitor de telas realiza a leitura
	C3C: Problemas relacionados ao funcionamento do leitor de telas	C3C1: O leitor de telas apresenta mau funcionamento aparente
C4: Problemas relacionados à sintaxe do ILA	C4A: Dificuldades relacionadas a variáveis e declaração de variáveis	C4A1: Participante comete equívoco ao utilizar o comando para declaração de variáveis
		C4A2: Participante realiza declaração de variáveis em local equivocado no código
		C4A3: Participante comete equívoco ao declarar o tipo de uma variável
		C4A4: Participante utiliza um identificador indevido para uma variável
		C4A5: Participante confunde o identificador com o tipo de uma variável
		C4A6: Participante declara um número excessivo de variáveis
		C4A7: Participante confunde variável dependente com variável independente
		C4A8: Participante utiliza uma variável que não foi previamente declarada
		C4A9: Participante utiliza ou deseja utilizar variável para representar uma constante
		C4A10: Participante comete equívoco na atribuição de valor a uma variável
		C4A11: Participante parece confundir as ideias de variável e função
	C4B: Dificuldades com a inicialização ou a finalização do bloco de instruções	C4B1: Dificuldades com a inicialização do bloco de instruções
	C4C: Dificuldades relacionadas a entrada e saída de dados	C4C1: Dificuldade na memorização ou utilização de comando de saída
		C4C2: Dificuldade conceitual relacionada à saída de dados
		C4C3: Dificuldade relacionada à exibição de um texto (saída de dados)
		C4C4: Dificuldade na memorização ou utilização de comando de entrada
C4D: Dificuldades relacionadas à memorização ou utilização de	C4D1: Dificuldades na memorização ou utilização de operadores aritméticos (+, -, *, /)	

Categoria final	Categoria intermediária	Categoria inicial
	caracteres especiais que desempenham funções específicas	C4D2: Dificuldades na memorização ou utilização de operadores relacionais (>, <, =)
		C4D3: Dificuldades na memorização ou utilização de outros caracteres especiais que desempenham funções específicas (vírgula, aspas, parênteses...)
	C4E: Dificuldades relacionadas a estruturas condicionais	C4E1: Dificuldade na memorização ou utilização de comandos que desempenham funções específicas em estruturas condicionais
		C4E2: Dificuldade na formulação de uma condição em uma estrutura condicional
		C4E3: Participante não considera todos os cenários possíveis em uma estrutura condicional
C5: Problemas relacionados às ferramentas para edição e execução dos códigos	C5A: Problemas relacionados ao uso do bloco de notas para criação dos códigos	C5A1: Participante apresenta dificuldades para abrir o bloco de notas
		C5A2: Participante apresenta dificuldades para salvar um arquivo adequadamente
	C5B: Problemas relacionados ao uso do prompt de comando	C5B1: Participante apresenta dificuldades na utilização do prompt de comando para abrir ou executar programas
C6: Problemas relacionados a aspectos organizacionais	C6A: Problemas relacionados ao ambiente onde foram realizados os encontros	C6A1: Participante expressa incômodo com o barulho no ambiente
	C6B: Problemas relacionados à organização dos encontros	C6B1: Participante apresenta dificuldades por conta do tempo planejado para as avaliações
		C6B2: Participante apresenta dificuldades por conta de um grande intervalo de tempo entre dois encontros consecutivos
C7: Problemas relacionados a aspectos emocionais	C7A: Participante expressa alguma emoção negativa	C7A1: Participante expressa falta de concentração, ansiedade, impaciência, raiva ou frustração
		C7A2: Participante expressa baixa autoestima
C8: Problemas relacionados a conceitos matemáticos	C8A: Dificuldades com operações básicas	C8A1: Dificuldades com operações básicas
	C8B: Dificuldades relacionadas à noção de função	C8B1: Dificuldades com o conceito de função
		C8B2: Dificuldades na determinação da função adequada
C8C: Dificuldades relacionadas à resolução de problemas	C8C1: Dificuldades na compreensão do problema a ser resolvida	

Fonte: elaborado pelo autor (2024).

Definidas as categorias iniciais, intermediárias e finais, passou-se, então, à terceira etapa da análise de conteúdo, que consistiu no tratamento dos resultados obtidos. Nesse

ponto, segundo Capelle, Melo e Gonçalves (2003), os dados brutos são submetidos a operações estatísticas, bem como são realizadas inferências e interpretações de acordo com o quadro teórico e os objetivos propostos no trabalho. No contexto desta pesquisa, todas as transcrições dos áudios foram lidas na íntegra, e os segmentos de texto que se encaixavam em alguma categoria foram devidamente codificados. Desse modo, como exemplo, se em um trecho foi identificado que um participante apresentava dificuldade na localização de um operador aritmético, tal segmento foi imediatamente categorizado com o código C1A1 (Quadro 54).

Finalizada a etapa de categorização, os segmentos de texto alocados em cada categoria foram analisados, com o objetivo de identificar padrões e realizar inferências a partir das falas dos participantes. Os resultados obtidos nesta etapa são abordados a seguir.

Desafios para o desenvolvimento de conceitos e habilidades do pensamento computacional através de uma linguagem algorítmica

C1: Problemas relacionados ao uso do teclado

Assim como anteriormente identificado através da análise lexicográfica, os resultados da análise de conteúdo sugerem que o uso do teclado é um aspecto que pode trazer dificuldades quando estudantes com deficiência visual empregam uma linguagem algorítmica na construção de códigos. Em particular, foi possível identificar problemas em duas subcategorias principais: na localização de teclas, e no que se refere à digitação equivocada de teclas.

C1A: Dificuldades quanto à localização de teclas no teclado

No que tange à localização, as dificuldades giraram especialmente em torno da identificação de operadores aritméticos, relacionais e outros caracteres especiais que desempenham funções específicas (vírgulas, aspas, e parênteses, por exemplo). Uma possível justificativa para o problema é o fato de alguns destes símbolos serem, em geral, utilizados com menor frequência na construção de textos. Além disso, em virtude das limitações sensoriais, os participantes precisavam memorizar a posição destes

elementos no teclado, o que representou um desafio adicional no processo de construção dos códigos. Os trechos a seguir ilustram dificuldades neste sentido.

HELENA: Barra! Essa daqui?

Pesquisador: Não, isso daqui é uma crase.

HELENA: Tá, uma barra tipo essa. Só que normal...

Pesquisador: É, só que essa tá invertida.

HELENA: A normal tá... a normal tá aonde? Aqui! [risos]

Pesquisador: Beleza. O símbolo de maior, você sabe onde é que tá no teclado?

RENATA: É... eu sabia, agora eu não tô lembrando onde fica. Seria o igual mais o shift?

Pesquisador: Não, esse é o mais, né? É o ponto com o shift. Isso. O que ele leu?

Outra dificuldade apontada, com menor frequência, se refere à mudança na localização de teclas devido ao formato diferente do teclado. Alguns participantes, já acostumados com determinado modelo de dispositivo, apresentaram problemas para encontrar teclas que esperavam localizar em outro ponto do teclado, como se percebe na passagem seguinte.

RENATA: Ah! Nossa, não! É que eu tava mexendo num computador mais cedo, que a seta era pra cá...

Pesquisador: Lembrou?

RENATA: Sim!

C1B: Problemas relacionados à digitação

Já em relação à digitação equivocada de teclas, foi possível notar três grupos de situações diferentes: o erro era notado e corrigido de imediato; o erro era notado em um momento posterior; ou o erro não era notado. Embora tal fato não tenha representado um impeditivo para a realização das atividades propostas na sequência didática, em alguns momentos o problema dificultou o processo de construção e execução dos códigos. Na análise lexicográfica, por exemplo, foram expostos casos em que tais erros de digitação não impediam que o programa fosse executado adequadamente e, com isso, muitas vezes os problemas não eram sequer notados. No entanto, tais equívocos tinham como consequência a exibição de textos incorretos para o usuário, podendo comprometer a semântica do programa construído. O trecho a seguir ilustra o caso de um programa criado pela participante Adriana, no qual se pede que o usuário digite um

número qualquer, sendo retornada uma mensagem que podia variar de acordo com o número digitado.

```

variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<=100) entao
escrever "O numerodigitado nao e maior que 100"
senao
escrever "O numero digitado n e maior que 100"
fim_se
fim

```

Embora não se verifiquem erros sintáticos no programa construído, caso o usuário digitasse um número maior que 100, seria exibida na tela do computador a mensagem “*O numero digitado n e maior que 100*”. A letra “n” em “*n e maior*”, que possivelmente foi mantida no código por engano pela participante, pode levar a uma interpretação equivocada do usuário, já que o “n” pode ser entendido como uma abreviação da palavra “não”.

Por outro lado, não é incomum que um erro de digitação impeça que um programa funcione, na medida em que os comandos devem ser escritos adequadamente para que sejam compreendidos pelo interpretador da linguagem. Em alguns casos, por exemplo, erros de digitação impediam que os participantes sequer conseguissem executar o programa. O trecho a seguir ilustra um exemplo no qual o participante Marcos não consegue executar um programa pelo fato de ter utilizado um termo incorreto para se referir ao arquivo no prompt de comando (“*ila programma04.ila*” ao invés de “*ila programa04.ila*”).

MARCOS: Ha ha ha. Erro na leitura do arquivo. Hum. Interessante. Deu erro no programa... Vou ver o que eu fiz de errado ali... Eita!

Pesquisador: Vê o nome do arquivo, se está programa 04 mesmo.

MARCOS: Vou ver se eu acho. O problema é que aqui não dá para ver.

(...)

MARCOS: Vai sumir... Calma aí... Acho que dá pra fazer isso aqui... Não, não vai dar...

Pesquisador: Tá, deixa eu ver aqui como é que está o nome do arquivo... Ah tá, é que você colocou programa com dois “m”s...

MARCOS: Oxi! Ah, computadorzinho... Tá bom, bora lá... Eita! Assim, ó... Calma aí, jovem... Programa... Foi, acho que foi...

Também foram observadas situações em que erros de digitação impediriam que o programa funcionasse, na medida em que comandos com funções específicas não foram

escritos adequadamente. Os trechos a seguir ilustram casos em que os termos “escrever” e “numérico”, respectivamente um comando de saída e um tipo de variável, foram escritos de maneira incorreta e, com isso, o programa não funcionaria se executado pelo participante.

Pesquisador: Tá, dá uma lida nesse “escrever” que você colocou aí e vê se tá tudo certinho...

MARCOS: Ah. Entrou isso aqui... Oxi... peraí... entrou duas letras né, meu pai amado...

Pesquisador: O que você tá dizendo aí: que você criou uma variável, que é numérica, e o nome dessa variável é “r”, tá? Só dá uma lida no seu “numérico” pra ver se tá tudo certinho com ele.

MARCOS: Ah... Já até entendi onde você quer chegar. Até entendi... Maldade “docê”, né cara... ô rapaz, sai daí... aqui a maldade... sai daí rapaz...

Outro fato que foi observado com elevada frequência foi a inserção excessiva, ou, pelo contrário, a ausência indevida de espaço entre caracteres. É possível inferir que tal fato seja consequência da leitura contínua realizada pelo leitor de telas. Ao posicionar o cursor sobre uma linha de código, o leitor faz uma leitura fluida do conteúdo da linha e, com isso, caracteres como espaço, aspas ou vírgulas não são apontados. Para percebê-los, é necessário realizar a leitura caracter por caracter, utilizando as setas direcionais do teclado ao longo da linha. O código abaixo, construído pelo participante Marcos, constitui um exemplo em que é difícil notar problemas relacionados à ausência ou ao excesso de espaços entre caracteres. Na linha “*escrever ‘O numero digitadonao e maior que 100’*”, a leitura realizada pelo leitor de telas dificulta que sejam notados os problemas no trecho “*digitadonao e maior*”.

```
variaveis
numérico a
inicio
escrever "Digite o numero"
ler a
se (a<=100) entao
escrever "O numero digitadonao e maior que 100"
senao
escrever "O numero digitado e maior que 100"
fim_se
fim
```

Por fim, outro problema observado com frequência foi o fato de os participantes apagarem involuntariamente algum caracter ou, em alguns casos, trechos significativos

do código construído. No caso de estudantes com deficiência visual, tal problema comumente ganhava proporções maiores porque muitas vezes o ocorrido não era notado de imediato, o que dificultava a correção do problema.

Pesquisador: Você apagou boa parte do que você escreveu!

ADRIANA: Ah, não! Apaguei tudo?

Pesquisador: Quase tudo!

ADRIANA: Eu me perdi!

(...)

ADRIANA: [risos] Ai, me perdi bonito!

C2: Problemas na identificação de elementos na tela do computador

A segunda categoria de desafios observados compreende problemas na identificação de elementos na tela do computador. Um fato comum, nesse sentido, era a perda da posição do cursor na tela, seja ao longo de uma linha de código, seja ao longo do programa como um todo. Já outro ponto identificado nesse aspecto era a não identificação de alguma ocorrência na tela do computador, como a inserção equivocada de letras maiúsculas ou minúsculas, ou um toque involuntário no *touch pad*, por exemplo.

C2A: Dificuldade na identificação da posição do cursor

Em geral, pessoas com deficiência visual utilizam o teclado para movimentar o cursor na tela do computador, ou mesmo localizar a sua posição em caso de dúvida. Para tal, é preciso utilizar as setas direcionais e ouvir a leitura realizada pelo leitor de tela a cada movimento. No entanto, esse processo pode ser confuso, uma vez que o leitor faz a leitura do caractere imediatamente posterior ao ponto onde está localizado o cursor. Desse modo, muitas vezes ocorriam dificuldades para identificar se o cursor estava antes ou depois de algum termo, o que acabava estendendo o processo de realização de correções no código. Os trechos a seguir ilustram duas situações onde o referido problema ocorre.

Pesquisador: Vê como é que ficou aí.

ADRIANA: Menor que 100, não... onde que tá esse menor, gente?

Pesquisador: Você tá nele.

ADRIANA: Tô nele?

Pesquisador: Uhum.

ADRIANA: **Aqui é o começo dele?** Calma aí... aqui?

Pesquisador: **É o final dele.**

Pesquisador: (...) Vem cá pro final da linha, usando a setinha, e aí no final, depois das aspas, você vai dar o enter. Mais um... **aí ele tá antes.**

HELENA: Mas por que?

Pesquisador: **Aí ele tá antes das aspas.**

HELENA: Ah, é, verdade!

Em outros casos, o problema residia em encontrar uma linha específica do programa para fazer alguma alteração ou, ainda, identificar em qual linha o cursor havia parado após a ocorrência de algum incidente, como um clique involuntário em uma seta direcional ou um esbarrão no sensor tátil do *notebook*. Por vezes, o participante acabava cometendo equívocos que demandavam grande esforço para correção, uma vez que o cursor havia se deslocado para uma posição indesejada, e o fato passava despercebido.

Pesquisador: Adriana... vê onde é que você tá...

ADRIANA: Eu tô quase no último aqui...

Pesquisador: Mas dá...

ADRIANA: Falta botar... eu tô no senão...

Pesquisador: Tá, mas dá uma olhadinha aonde você tá, em qual linha você tá...

ADRIANA: **Eu não faço a mínima ideia, eu me perdi no meio do caminho...** [risos]

Pesquisador: Acho que você clicou em alguma coisa aí, **você não tá no final do programa...**

ADRIANA: Não? **Eu tô aonde?**

Pesquisador: Tenta se localizar aí, onde é que você tá...

ADRIANA: Calma aí, deixa eu voltar aqui no começo... Mariana...

ADRIANA: Tá, aqui tá certo... ué, **onde que eu tô? Nem eu tô me achando!** Será que eu tô no programa da Mariana aqui de cima? Me desconcentrei... me desconcentrei bonito aqui... **cadê eu, que eu não tô vendo?** Achei o “b” igual não sei o quê... não tô achando... eu não escrevi em lugar nenhum?

Pesquisador: Você escreveu no meio do programa.

ADRIANA: Do primeiro ou do segundo? Pode dar esse *spoiler*? [risos]

Em alguns casos, a dificuldade na identificação da posição do cursor era tamanha que acabava tornando a tarefa de construção do código exaustiva, trazendo um sentimento de frustração para o estudante, como se observa no trecho seguinte.

ADRIANA: É maior que 100, calma aí... **eu vou me achar...** calma aí, eu me perdi do “a”, do “m”. **Eu tô na frente ou depois do “m”?**

Pesquisador: Você tá antes do “m”.

ADRIANA: Tô no número! Não, não tô não... não! Não é menor ou maior, **eu já tô toda perdida.**

Pesquisador: O número é maior que 100.

ADRIANA: Maior. Eu botei um “i”, né... **nem sei o que que eu fiz mais**, calma aí...

Pesquisador: Tá, aí você foi pra linha de baixo. Apaga esse “i” e esse “a”.

ADRIANA: **Eu fui pra linha de... pra linha de baixo?**

Pesquisador: Sim. Volta pra linha de cima... isso!

ADRIANA: Tá faltando o...

Pesquisador: Você tá no “não”, você tá no “não”.

ADRIANA: “Não”? [risos]... **eu vou dar um grito!** Agora eu tô no “m”.

C2B: Participante não identifica algum ocorrido na tela

Como mencionado, cabe destacar, ainda, a ocorrência de momentos em que o participante não notava alguma ação que tinha sido praticada involuntariamente no processo de construção do código. De modo semelhante, às vezes o estudante apresentava dificuldades para identificar se alguma ação tinha de fato sido implementada ou não. Os trechos a seguir ilustram os fatos mencionados.

Pesquisador: Perfeito. Aí, para terminar, você vai escrever o que? O b foi para baixo, hein!

MARCOS: Ahn?

Pesquisador: O b foi pra linha debaixo.

MARCOS: Ah! Foi “comer uma farinha sozinho”. Sai daí!

HELENA: Eu apaguei o outro numérico, né? Não!

Pesquisador: Você apagou só o escrever.

C3: Problemas relacionados ao leitor de tela

Uma terceira categoria de dificuldades observadas remete a problemas com o leitor de tela. Nesse sentido, as principais questões identificadas se referem a inconsistências na leitura de símbolos ou palavras, bem como desconfortos relacionados à voz ou à cadência na leitura realizada. Uma terceira subcategoria, identificada com menor frequência, se refere a possíveis ocorrências de mau funcionamento no leitor de tela.

C3A: Inconsistências na leitura de símbolos ou palavras

Em primeiro lugar, observou-se que em alguns casos não era realizada a leitura de algum símbolo específico em uma linha. Por vezes, tal fato era esperado por conta do tipo de leitura realizada pelo leitor de telas, que ao ler uma linha por inteiro, por exemplo,

não demarca determinados símbolos e caracteres, como sinais de pontuação ou espaços. No entanto, em alguns casos a omissão de algum símbolo na leitura levava a uma mudança na interpretação do conteúdo da linha. O trecho a seguir ilustra um exemplo em que a omissão do símbolo “circunflexo” na leitura realizada pelo leitor de telas levou a participante a uma compressão matematicamente equivocada do que de fato estava escrito na linha.

Pesquisador: Você colocou quadrado igual a o que nessa linha?

HELENA: Quadrado igual a 2.

Pesquisador: A 2?

HELENA: Igual...

Pesquisador: Vai indo pra direita com a setinha pra você ver todos os termos que você escreveu.

HELENA: Hum... tá com... tá com... é... **os simbolozinhos que você explicou pra fazer os cálculos.**

Pesquisador: E esse simbolozinho que tá aí tá representando que cálculo?

HELENA: É... **quadrado de um número.**

Pesquisador: Perfeito.

HELENA: O número vezes ele mesmo.

Pesquisador: Exatamente. O **circunflexo é potência, então o que você colocou aí é a circunflexo 2, que significa a elevado ao quadrado.** Ou seja, a vezes ele mesmo, 2 vezes só. Próxima linha...

O trecho remete a uma etapa da construção do código de um programa para elevar um número ao quadrado, realizado pela participante Helena. No caso, a estudante era questionada pelo pesquisador em relação ao conteúdo de uma linha do código, na qual se verificava o texto “*quadrado=a^2*”. Influenciada pela leitura realizada pelo leitor, a participante entendeu que na linha estava escrito “quadrado igual a 2”, o que não correspondia à realidade pois o símbolo “^” é utilizado para representar uma operação de potenciação. Desse modo, a interpretação correta do conteúdo da linha seria “quadrado igual a ‘a’ elevado a 2”.

O problema na interpretação pode ser justificado por dois fatores: o fato de o símbolo “circunflexo” ter sido omitido na leitura; e o fato de o leitor de tela ler o símbolo “=” como “igual”, e não “igual a”. Dessa forma, além de não perceber a operação de potenciação, a participante pode ter confundido a variável “a” com uma possível preposição “a” de “igual a”. Assim, em termos matemáticos, em lugar de compreender o texto como “quadrado = a²”, a participante o compreendeu como “quadrado = 2”.

Já em outros casos, o leitor de tela realizava a leitura de um termo de forma não usual ou não esperada. Um exemplo é o caso do símbolo “-”, utilizado para representar

a operação de subtração, mas que também pode ser entendido como hífen. Nesse caso, o contexto irá definir a interpretação adequada. No entanto, além do leitor de tela não fazer a leitura do símbolo ao ler uma linha inteira, na leitura por caractere o símbolo é sempre lido como hífen, como ilustra o trecho a seguir.

Pesquisador: Foi? Então, beleza! Esse vai ser o sinal que a gente vai utilizar no ILA pra poder fazer uma operação de adição, tá? E o sinal de menos?

RENATA: Ah, esse é fácil.

Pesquisador: Beleza, o que que ele leu pra você?

RENATA: Hífen

Pesquisador: Hífen. Ele não vai ler “menos” quando você colocar esse sinal, mas esse é o sinal que você vai usar.

Outro símbolo cuja leitura com frequência levava a interpretações equivocadas era o “=”, em especial quando o mesmo era seguido de uma variável “a”, como mencionado anteriormente. No trecho seguinte, por exemplo, ao fazer a leitura da linha “soma=a+b”, Helena parece ter uma compreensão diferente da esperada.

Pesquisador: Perfeito. Na linha 8 do programa, vai lá na linha 8.

HELENA: Soma é igual a mais b.

Pesquisador: O que vocês acham que tá acontecendo nessa linha?

HELENA: Hum... ali é... ele tá, os números que a gente colocou...

Pesquisador: Uhum.

HELENA: Vai dar o resultado mais b. É isso?

C3B: Problemas relacionados à voz ou à cadência na leitura do leitor de telas

Outro ponto observado em relação ao leitor de tela foi que, em alguns momentos, os participantes apresentavam dificuldades para compreender a leitura realizada. Em casos específicos, os estudantes chegavam a manifestar insatisfação ou desconforto com a voz utilizada pelo recurso ou com a forma como algum trecho era lido. Os trechos seguintes ilustram as situações apontadas.

Pesquisador: Perfeito.

HELENA: Eu não tava entendendo o que ela tava falando.

Pesquisador: Agora você entendeu?

HELENA: É! Variáveis...

HELENA: Falou meu nome e que eu sou aluna do Pedro II.

FABIO: Nossa senhora! Falou meu nome de um jeito...

HELENA: [risos] O meu nome também saiu muito ruim.

FABIO: Nossa senhora!

C3C: Problemas relacionados ao funcionamento do leitor de telas

Por fim, também houve casos em que os participantes relataram algum mau funcionamento no leitor de telas, o que os impedia de continuar o processo de construção dos códigos adequadamente. Alguns exemplos são situações em que o estudante passava a não ouvir mais a leitura, ou esta passava a ser realizada de modo impróprio, como segue.

Pesquisador: Pronto. Então é o seguinte: se o número for menor ou igual a 100, tem que aparecer na tela: “o número não é maior que 100”.

RENATA: Não tá falando... ah!

Pesquisador: Não tá falando?

ADRIANA: Tá duplicado!

Pesquisador: O que que ele tá lendo?

ADRIANA: 1, 1...

Pesquisador: Deixa eu ver aqui...

ADRIANA: Mas tá aparecendo dois “1” aí?

Pesquisador: Não, tá aparecendo um só.

ADRIANA: [risos] Nossa, esse NVDA tá muito louco...

Pesquisador: Ele está lendo coisas aleatórias aí?

ADRIANA: Tá L-RA...

C4: Problemas relacionados à sintaxe do ILA

A categoria na qual foi identificado o maior número de ocorrências se refere a problemas relacionados à sintaxe do ILA. Nesse aspecto, foram identificadas cinco subcategorias intermediárias de dificuldades, sendo elas relacionadas a: variáveis e declaração de variáveis; inicialização e finalização de blocos de instruções; entrada e saída de dados; memorização e utilização de caracteres especiais que desempenham funções específicas; e estruturas condicionais.

C4A: Dificuldades relacionadas a variáveis e declaração de variáveis

De maneira geral, pôde-se observar significativa dificuldade dos estudantes com a noção de variável, bem como sua aplicação na construção dos códigos. Um dos principais problemas, nesse aspecto, consistia na falta de compreensão de variável enquanto um elemento capaz de armazenar valores que poderiam ser modificados durante a execução de um programa. Assim, embora seja possível atribuir um valor constante a uma variável declarada, no contexto da pesquisa as variáveis eram utilizadas para armazenar dados fornecidos pelo usuário ou resultados de operações de processamento de tais dados. No entanto, muitas vezes os participantes pensavam em utilizar variáveis para valores que, nos problemas apresentados, eram constantes.

O trecho a seguir ilustra um caso em que a participante Helena pensa em usar variáveis para os coeficientes angular e linear de uma função afim. No caso, esperava-se que a estudante montasse uma estrutura condicional utilizando as funções $y=2x+40$ e $y=2x+50$. Porém, Helena manifesta a intenção de usar variáveis para os valores constantes nas funções.

Pesquisador: Aí pensa na quantidade de variáveis que você vai precisar nesse problema.

HELENA: Ai, nossa, um tanto!

Pesquisador: Será?

HELENA: Seria uma variável ou um número fixo? **Eu preciso de uma variável pro 2 reais...** pro 2 reais é... número fixo! Número fixo? Ai, não sei! **Eu preciso de uma variável pro 50, pro 2 e pro 40.**

Pesquisador: Pensa, variável é uma coisa que tá variando! O 50 é um número fixo, o 40 também, e o 2 também. Pensa no que vai estar variando de acordo com as coisas do programa.

Foi possível observar, ainda, dois equívocos comuns entre os participantes: a confusão entre identificador e tipo de uma variável, e a confusão entre os identificadores de variáveis dependentes e independentes. No primeiro caso, o participante usava o tipo de uma variável para se referir ao seu identificador. Já no segundo caso, o participante aparentava não ter clareza em relação a que variável representava qual grandeza em um problema. Os trechos a seguir ilustram, respectivamente, os casos mencionados.

Pesquisador [se dirigindo à Helena]: E qual é o nome da sua variável agora?

HELENA: Numérico.

Pesquisador: Numérico é o tipo de variável, você tem que dar um nome pra ela. Você tá perguntando o que da pessoa?

HELENA: A idade dela.

Pesquisador: B igual a “a” vezes 8. Só que agora vai ser qual conta que você vai colocar ele pra fazer?

RENATA: “A” igual a “b” vezes 6?

Pesquisador: B igual...

RENATA: “B” igual a “a” vezes... nossa!

Outro problema verificado com frequência consistia na atribuição de valor a uma variável. Muitas vezes, os participantes compreendiam um problema, chegando a uma relação matemática correta para modelá-lo. No entanto, enfrentavam dificuldades para representar tal relação corretamente no código do programa. O trecho abaixo ilustra um momento em que Marcos tenta construir uma operação de processamento para dobrar o valor de um número fornecido pelo usuário. O participante, porém, inverte a ordem da atribuição ($a.2=b$, em lugar de $b=a.2$), e ao tentar corrigir o erro, acaba cometendo outro equívoco ($b.2=a$, em lugar de $b=a.2$).

MARCOS: Ah, tá. A vezes... Nossa, está confuso. A é igual a... vezes 2... Ah, **no caso, seria a vezes 2 é igual a b?**

Pesquisador: Isso, só que você tem que inverter porque a variável resultado...

MARCOS: Vem antes...

Pesquisador: Vem antes

MARCOS: Do número.. do resultado...

Pesquisador: Do igual. Exato, do igual.

MARCOS: Ah, do igual?

Pesquisador: Aham

MARCOS: Aí **no caso o a que vem depois do igual?**

Pesquisador: Isso.

MARCOS: No caso, **fica b vezes 2 é igual a a?**

Pesquisador: Hum, quase isso. O b tem que vir antes, sim, mas qual é o número que você está multiplicando por 2?

MARCOS: Hum, então seria a variável a ali também... É, tá estranho... Tá... Seria no caso b...

Pesquisador: Uhum...

MARCOS: Aí teria que ter a variável a, que no caso vai ser o número que está somando...

Pesquisador: Uhum...

MARCOS: Vezes 2...

Pesquisador: E onde é que entra o igual aí?

MARCOS: Entre os dois ali, né?

Pesquisador: Então...

MARCOS: Ah tá, b é igual a a vezes 2...

Pesquisador: Isso...

MARCOS: Ah, entendi...

C4B: Dificuldades com a inicialização ou a finalização do bloco de instruções

Em um pequeno número de situações, foram identificados problemas com a inicialização ou a finalização de um bloco de instruções. Nesses casos, de maneira geral o participante não utilizava corretamente o comando “início” ou o comando “fim”. Nos trechos a seguir, podem ser observados momentos em que Renata e Ana não utilizam adequadamente o comando “início”.

RENATA: Pronto.

Pesquisador: O seu, Renata, o início, ele é... ele tá em linha trocada com a linha do numérico.

RENATA: ãhn?

Pesquisador: Ele tá na segunda linha, ele tem que estar na terceira linha.

RENATA: Ah!

Pesquisador [se dirigindo à Ana]: Faltou uma coisa no seu programa.

ANA: O quê?

Pesquisador: Quando a gente vai começar a programação de fato, que palavra que a gente tinha que usar? Que começa com i...

ANA: Ian! Igual [risos]...

Pesquisador: Início! Lembra? Sempre que a gente...

ANA: Inicie este número [risos]...

Já o programa a seguir, construído por Adriana e apresentado anteriormente na análise a posteriori da sequência didática, representa um caso em que, além de outros erros cometidos, a participante não utiliza o comando “fim” para finalizar o bloco de instruções.

```
variaveis
numericoa, b
inicio
escrever      Quantoos tapetes foi vendido"
ler a
s(a>=100) enttao
b=1500+300
fim_se
se_naoentao=1500
```

C4C: Dificuldades relacionadas a entrada e saída de dados

Outra subcategoria de problemas relacionados à sintaxe do ILA consiste em dificuldades referentes à entrada e à saída de dados. Em primeiro lugar, no que se refere à saída de dados, a maior parte dos problemas girava em torno da correta elaboração de um texto que seria exibido para o usuário. Nesse sentido, os participantes ora apresentavam dificuldades em definir o que deveria ser retornado para o usuário, ora tinham problemas em como proceder para exibir cadeias de texto, variáveis, ou combinações entre ambos. O trecho seguinte representa uma situação em que o participante Marcos apresenta as duas dificuldades mencionadas.

MARCOS: Entra aqui, no caso, é... deixa eu só limpar isso aqui... esse... aí na linha de baixo, vai ter que entrar aqui, no caso, a parte do retorno, né? A saída dele...

Pesquisador: Exatamente. E qual vai ser a saída?

MARCOS: Escrever.

Pesquisador: Escrever...

MARCOS: No caso...

Pesquisador: Uhum...

MARCOS: É... a quantidade de megabytes.

Pesquisador: Uhum...

MARCOS: É maior... é menor que “b”... ô, maior que “a”... e... maior que “a”? É... nesse caso, seria maior que “a”, menor que “b”?

Pesquisador: Nesse caso a gente não precisa dizer se é menor ou maior, porque o programa só tá querendo quanto ele vai gastar. Quanto ele vai gastar?

(...)

MARCOS: Ah, tá, então no caso seria só b.

Pesquisador: O “b”. O “b” é o valor total. O “b” vai ser igual a 50 mais 2 vezes “a”. Então o que você precisa escrever é esse valor, e esse valor você tá usando a variável.

MARCOS: B.

Pesquisador: B, então escrever “b”.

MARCOS: Ah, tá. Esse “b”... esse “b” aqui, no caso, ele vai ser entre aspas, né?

Pesquisador: Não, porque...

MARCOS: Ah, vai ser solto aqui?

Pesquisador: Vai ser solto porque é a variável. Se você colocasse entre aspas, ele ia escrever na tela a letra “b”.

MARCOS: Uhum. Muito bem. Ok.

Ainda considerando problemas relacionados à entrada e à saída de dados, também foram identificados, em menor número, problemas com a memorização dos comandos a serem utilizados para armazenar informações ou exibir textos na tela do computador, como ilustram os trechos a seguir.

ANA: Escreva...

Pesquisador: Isso aí! Só que a gente usava escreva ou usava outra palavra parecida?

ANA: Digite... não...

Pesquisador: Qual era o comando que a gente usava pra saída?

ANA: Ai, não lembro... ai...

Pesquisador: Quase escreva, mas...

ANA: Digite... é... não sei, esqueci!

Pesquisador: Escrever!

ANA: Escre... tava como antes?

Pesquisador: Escreva.

ANA: Escreva, é escrever, né?

ADRIANA: Pronto... cismeí com esse “x”, pelo amor de Deus!

ADRIANA: [mexe no computador]

ADRIANA: Não, é ler! Não é leia não, é ler! Botei dois “e”s, que legal... ler... aí, na próxima linha...

Já em outras situações, os problemas tinham natureza conceitual, especialmente no que se refere à entrada de dados. Muitas vezes, por exemplo, os participantes

utilizavam o comando “ler” para armazenar um valor que já havia sido atribuído a uma variável, em lugar de uma nova entrada fornecida pelo usuário.

Pesquisador: Agora me explica uma coisa, o que vai acontecer com esse “ler” que você colocou aí?

ADRIANA: Ele vai ler a... o resultado aqui de “a” vezes 2.

Pesquisador: Mas tem alguma entrada sendo colocada pra ele ler?

ADRIANA: Sim, eu declarei a variável “b” que vai ser um resultado da variável... que vai ser a multiplicação de “a” vezes 2, e agora ele vai ler esse... esse “b”... esse resultado aqui de “b”. Aí vai retornar o valor.

Pesquisador: É, na verdade...

ADRIANA: Não precisa?

Pesquisador: Não precisa, na linha de cima você já colocou que o “b” vai ser 2 vezes “a”. Então ele já tá armazenado no “b”, esse resultado.

C4D: Dificuldades relacionadas à memorização ou utilização de caracteres especiais que desempenham funções específicas

Outra subcategoria de questões relacionadas à sintaxe, que se relaciona com a mencionada anteriormente, se refere à memorização e utilização de caracteres especiais que desempenham funções específicas na linguagem, como vírgulas, aspas ou parênteses. Isto porque, muitas vezes, o uso desses caracteres era associado ao uso de comandos de entrada e saída.

Uma confusão comum, neste sentido, se referia à necessidade ou não de utilizar aspas após tais comandos. Tal dúvida ocorria porque, em geral, quando o comando “ler” era utilizado para armazenar algum valor inserido pelo usuário, não se fazia o uso de aspas. Já no caso do comando “escrever”, as aspas poderiam ser utilizadas ou não, de acordo com o que se desejasse exibir como saída. Tal fato, muitas vezes, ocasionou dúvidas e culminou em equívocos realizados pelos participantes.

Pesquisador: Isso, mas quando a gente quer escrever uma frase, essa frase tem que estar entre...?

HELENA: Ai...

Pesquisador: Mas tá certo, é escrever... tá certo, tá certo, por que você tá apagando?

HELENA: Eu não fiz...

Pesquisador: Escrever...

HELENA: Entre aspas, né?

Pesquisador: Aspas vem depois do escrever.

HELENA: Ah, é!

RENATA: E quando a gente põe “ler”, depois tem que botar entre aspas também, né?

Pesquisador: Não. O “ler” é sem aspas.

No entanto, as dificuldades nesse sentido não se limitavam a situações relacionadas à entrada e saída de dados. Em muitos casos, por exemplo, os participantes não se recordavam da necessidade de separar identificadores de variáveis por vírgulas no momento da declaração. Também houve situações em que o estudante não lembrava qual caractere (aspas ou parênteses, na maior parte das vezes) deveria ser utilizado em um contexto específico. No trecho seguinte, Helena apresenta, em diferentes momentos, problemas para definir o caractere adequado a ser utilizado.

Pesquisador: É espaço, ou é outra coisa?

HELENA: Vírgula!

Pesquisador: Isso.

HELENA: É vírgula ou é ponto? Que eu lembro que tinha dado problema da outra vez.

Pesquisador: Não, isso foi no número decimal, que era 6 e 50. Nesse caso não tem a ver com isso, não.

HELENA: Então vírgula.

Pesquisador: Isso.

HELENA: B.

Pesquisador: Dá um spacinho.

HELENA: Tem que dar espaço...

Pesquisador: Isso, show. E agora?

HELENA: Próxima linha, início. V... início.

Pesquisador: Uhum.

HELENA: Início, aí agora vem os comandos... tá, se... ai... complicado... se a é menor ou igual a 40... eu tenho que colocar entre aspas? Não, né?

Pesquisador: Entre parênteses.

HELENA: Entre parênteses.

C4E: Dificuldades relacionadas a estruturas condicionais

A última subcategoria de dificuldades relacionadas a aspectos sintáticos remete às estruturas condicionais. Nesse ponto, a primeira dificuldade identificada se refere à memorização e organização dos comandos nas estruturas “se entao” e “se entao senao”. Além de equívocos na escrita dos comandos, em alguns casos os estudantes apresentavam problemas para organizar corretamente a estrutura condicional simples (se (<condição>) entao <bloco de instruções> fim_se), e a composta (se (<condição>) entao <bloco de instruções> senao <bloco de instruções> fim_se). Os trechos seguintes ilustram exemplos de problemas na escrita de comandos, e também na organização da estrutura.

Pesquisador: E agora olha esse fim se, e vê se ele tá certo.

ADRIANA: Fim se, deixa eu ver... será que era com underline?

Pesquisador: Era com underline.

ADRIANA: Era com underline?

Pesquisador: Uhum.

ADRIANA: Calma aí... era com isso aqui.

Pesquisador: Perfeito.

RENATA: Tá certo, né?

Pesquisador: Uhum... **sem o til.**

RENATA: Oi, falou alguma coisa?

Pesquisador: É, **sem o til, a gente não coloca nenhum acento aí nos comandos.**

RENATA: Ah, eu esqueci.

Pesquisador: Beleza!

ADRIANA: Fim se... fim... é... fulaninho... fulaninho, fulaninho, fulaninho, aqui... fim se... fim se... tá, a próxima linha... não... senão, senão eu acho que era junto... sublinha também, né? **Senão... é... senão, então, senão então... sinal? Ué? Senão... senão, né, é... então... senão então... aí próxima linha... escrever... a condição vai estar na próxima linha... não espera aí, não é escrever não... senão então... calma aí... é, eu vou botar na próxima linha... "b" igual... não... o que que eu tô fazendo?**

Além disso, houve casos em que os participantes tiveram dificuldades para construir adequadamente as condições a serem verificadas, ou ainda para considerar todos os cenários possíveis de um problema. Nos casos seguintes, por exemplo, Marcos e Helena se confundem ao tentarem estabelecer qual condição deve ser inserida em uma estrutura condicional.

MARCOS: Nesse caso aqui... é... se "a" for... **se "a" for menor... na verdade lá é maior, né? Não, se "a" for... se "a" for menor do que 10 horas, então "a" é igual a 80.** Ah, não, escrever... ah, não, aqui nesse caso é... **se "a" menor que 80... se "a" menor... ah, tá, se "a" menor que 10...** sim, sim, peguei, peguei a referência! Peguei! Inclusive tinha um modo desse ali na outra, mas de boa!

HELENA: É, tipo, **se a pessoa vender 100 tapetes, ela ganha 300 "conto", quem dera! É...** ela consegue ganhar 300 reais, além dos 16 e dos 1500. **Então... se a for menor que 300... não, menor que 100...** se a menor que 100... se a menor... o quê que eu fiz de errado?

O programa a seguir, por sua vez, já mencionado na análise a posteriori do encontro 4 da sequência, foi construído pela participante Renata e tinha por objetivo retornar o valor total a ser gasto com a compra de uma determinada quantidade de cadernos. Pode-se notar que a estudante não considera o caso em que são comprados exatamente quatro cadernos, e com isso o programa é incompleto.

```
variaveis
numerico a, b
inicio
escrever "Digite o numero de cadenos"
```

```

ler a
se (a<4) entao
b=a*8
escrever "O resultado e igual: ", b
fim_se
se (a>4) entao
b=a*6.50
escrever "o resultado e:", b
fim_se
fim

```

C5: Problemas relacionados às ferramentas para edição e execução dos códigos

C5A: Problemas relacionados ao uso do bloco de notas para criação dos códigos

Outra categoria de dificuldades identificada na análise de conteúdo se refere a problemas relacionados a ferramentas para a edição e execução dos códigos. Como o ILA não possui um editor de código próprio, é necessária a utilização de algum editor de textos para a criação dos códigos. No contexto da pesquisa, foi utilizado o bloco de notas. No entanto, em alguns casos os estudantes apresentaram dificuldades para memorizar os procedimentos para acessar o editor, uma vez que o acesso deveria ser feito pelo teclado por conta das limitações de natureza visual, como ilustra o trecho a seguir.

Pesquisador: Quase isso, só que você tem que deixar apertada a tecla do Windows e usar uma letra que era a letra...

SARA: O “r”?

Pesquisador: Isso. Só dá um “esczinho”. Isso, show. E aí a gente usava uma palavra pra abrir...

SARA: Era... **piad, alguma coisa assim...**

Pesquisador: Quase isso: note...

SARA: Era notepid...

Pesquisador: Notepad...

SARA: Notepad...

Pesquisador: Com “a”... **com “a”, sem o “e”.**

Em outros casos, os participantes se esqueciam de salvar um arquivo modificado, ou não se recordavam do atalho para a realização do salvamento (control+s). Os trechos a seguir ilustram as situações mencionadas.

Pesquisador: Beleza. Você salvou? Salvou, né?

ADRIANA: **Ih, não! Boa! Esqueci disso! É a minha pressa.** Pronto.

Pesquisador: Então vamos lá. **Lembra como é que salva?**

HELENA: Não.

Pesquisador: Control s.

HELENA: Control s, né? Esse aqui é o control.

Pesquisador: Isso.

HELENA: Control s.

C5B: Problemas relacionados ao uso do prompt de comando

No entanto, o maior problema observado nessa categoria se refere ao uso do *prompt* de comando para a execução dos programas criados, ou mesmo para abertura de arquivos no bloco de notas. Em diversos momentos, os participantes apresentaram dificuldades para se recordar das etapas ou para realizar corretamente os procedimentos necessários. Em muitos dos casos, o processo se tornava demorado e demandava mediação intensa do pesquisador. Destaca-se que todos os participantes em algum momento da sequência didática apresentaram essa dificuldade, que muitas vezes persistia até os últimos encontros. Os trechos a seguir são alguns exemplos do problema evidenciado.

Pesquisador: Foi. Abre pra mim o programa 4, por favor.

ADRIANA: Tá, é... deixa eu lembrar... ila ponto ila... é... programa 4, né? Ila ponto ila?

Pesquisador: Primeiro você tem que ir pra pasta do ila, que é o cd...

ADRIANA: Pasta do... ah, eu sempre esqueço disso... cd...

Pesquisador: Espaço...

Pesquisador: Show, e aí você vai abrir o programa4. Antes disso, você tem que ir para a pasta do ILA. Você lembra como é que você fazia para ir para a pasta do ILA?

MARCOS: Ila ponto...

Pesquisador: Antes disso você colocava cd...

MARCOS: Ah é, cd ponto...

Pesquisador: Cd espaço c dois pontos...

MARCOS: Ah sim, lembrei. Acho que eu lembrei esse código aí. Deixa eu ver só... Aqui não... C dois pontos... Cara, tinha algum outro comandozinho aqui.

Pesquisador: Tinha o contrabarra. Barra invertida.

MARCOS: Ah, tá por isso. Esse, foi?

SARA: Hum, eu tenho que apertar de novo...

Pesquisador: De novo, isso...

SARA: Aí depois... hum... "r".

Pesquisador: Os dois juntos. Pode ir lá.

SARA: Os dois juntos...

Pesquisador: Só que ao invés de notepad...

SARA: Aí eu clicava no ila...

Pesquisador: "C"...

SARA: "C"... era "c", ponto... "c" ponto ila...

Pesquisador: Quase isso. Isso, você vai usar alguma coisa parecida com isso, quando você já tiver lá no *prompt*. Pra abrir o *prompt* você usava três letras: "c", "m", "d".

SARA: Cod, né?

Pesquisador: Cod?

SARA: É, cod, não?

Pesquisador: Cmd.

SARA: Cmd?

Pesquisador: Uhum. Digita aí... show.

RENATA: Hum, eu acho que era a tecla Windows.

Pesquisador: Aham.

RENATA: Mas eu não lembro o que era pra escrever.

Pesquisador: Pode ser ou prompt ou cmd... isso!

RENATA: Enter também?

Pesquisador: Aham.

RENATA: É um pouco diferente do meu... foi!

Pesquisador: Foi! Aí agora, pra abrir o arquivo usando o prompt a gente tinha que primeiro ir pra pasta onde o arquivo tá. Você lembra como fazia isso?

RENATA: Hum... é... c... é.... c dois pontos raiz ila, algo assim?

Pesquisador: Quase isso. Primeiro cd, espaço.

RENATA: Calma... c...

Pesquisador [se dirigindo à Helena]: Dá um enter. Isso. Agora vamos abrir esse programa. Lembra como é que faz para abrir?... Uhum...

HELENA: Control c [risos]... eu não lembro! Control c?

Pesquisador: Teclinha do Windows.

HELENA: É, isso aqui, é!

Pesquisador: E qual letra?

HELENA: C?

ANA: Pode falar?

HELENA: B?

ANA: R.

HELENA: R.

Pesquisador: E aí a gente tinha que digitar três letras, quais eram as...

ANA: É... cmd!

Pesquisador: Quase isso.

ANA: Cn...

Pesquisador: Cmd.

ANA: Cmd.

Pesquisador: Isso.

ANA: Eu só errei uma. Cmd.

HELENA: É isso?

Pesquisador: Isso. Dá o enter.

ANA: Eba! Agora eu vou fazer... licença agora, que eu vou fazer meu programa matemático! Com licença!

Pesquisador: Primeira coisa, a gente tem que ir pra pasta onde tá o programa. Lembra que...

HELENA: ILA...

ANA: Não.

HELENA: Cd...

Pesquisador: Cd...

ANA: Cd...

Pesquisador: Hum...

HELENA: Cd ponto...

Pesquisador: Espaço...

HELENA: Espaço...

Pesquisador: Hum...

HELENA: Ila...

ANA: Ila... não!

HELENA: Barra...

ANA: Não! [risos]

HELENA: Ponto...

Pesquisador: C dois pontos...

HELENA: Nome do programa...

ANA: C...

Pesquisador: Isso é depois.

(...)

HELENA: C dois pontos...

Pesquisador: [se dirigindo à Ana]: Ó, primeiro você tinha que ter colocado o c, que você não colocou!

ANA: Eu coloquei sim.

Pesquisador: Hum hum...

ANA: Coloquei sim!

Pesquisador: Barra invertida. Vai! Cd, apagou tudo...

ANA: Cd...

HELENA: Ila agora!

Pesquisador [se dirigindo à Ana]: Espaço...

Pesquisador [se dirigindo à Helena]: Isso.

ANA: Espaço...

Pesquisador [se dirigindo à Ana]: C! C...

ANA: C...

Pesquisador: Dois pontos...

HELENA: Ila espaço nome do programa...

ANA: Dois pontos...

Pesquisador [se dirigindo à Helena]: Isso. Enter...

HELENA: Enter...

Pesquisador [se dirigindo à Ana]: Barra invertida...

HELENA: C ila maior...

Pesquisador [se dirigindo à Helena]: Beleza.

HELENA: Helena ponto ila...

ANA: Barra invertida...

Pesquisador [se dirigindo à Helena]: Aí agora é ila espaço helena ponto ila...

ANA: Ila...

Pesquisador [se dirigindo à Ana]: Ila...

HELENA: Ila espaço...

ANA: I... ila...

Pesquisador [se dirigindo à Ana]: Enter... o que ele leu?

ANA: C ila maior.

Pesquisador: Perfeito, entrou na pasta. Aí agora você vai abrir o seu programa. Como é que a gente abria o programa, botava ele pra rodar, lembra?

ANA: Hum...

Pesquisador: Ila...

ANA: Ila ponto...

Pesquisador: Ila espaço...

ANA: Ila espaço...

Pesquisador: O nome do arquivo. Qual o nome que você salvou?

ANA: É... pro... pro... calma aí, como foi? Acho que foi programa matemático? Uma coisa assim...

Pesquisador: Programa matemático.

ANA: Programa matemático, aí bota isso, né?

HELENA [se dirigindo à Ana]: É o nome do programa? Meu Deus!

Pesquisador: Sem espaço... ponto?

ANA: Ponto ila!

C6: Problemas relacionados a aspectos organizacionais

Entre as dificuldades apresentadas pelos estudantes, foi possível notar que algumas foram consequência de falhas na organização ou imprevistos durante o andamento dos encontros. Entendemos que tais possibilidades devem ser consideradas no momento de planejamento da sequência, de modo que não haja prejuízos na aprendizagem e no rendimento dos estudantes.

C6A: Problemas relacionados ao ambiente onde foram realizados os encontros

Uma das subcategorias identificadas remete a problemas no ambiente em que os encontros eram realizados. Em particular, os estudantes apontaram, em alguns momentos, que havia muito ruído na sala em que a sequência didática era conduzida, o que possivelmente teve impacto em sua capacidade de concentração. O trecho seguinte é exemplo de uma situação nesse sentido.

RENATA: Pode ser! Meu notebook sempre quando vou conectar o fone fica aparecendo um ícone perguntando qual foi a entrada que eu usei.

Pesquisador: Ok.

RENATA: Nossa, não tem nem comparação.

Pesquisador: Tá dando pra ouvir?

RENATA: Uhum.

Pesquisador: Show!

RENATA: É que **também tem um monte de gente falando, né?**

Pesquisador: Aham.

RENATA: Mas isso não dá pra evitar.

Renata menciona a impossibilidade de evitar o ruído (falas de outras pessoas) pelo fato de os encontros serem realizados na sala do Napne, local em que diversas sessões de atendimento educacional especializado ocorriam simultaneamente. O mesmo problema ocorria nos encontros com o grupo 4 (Helena, Ana e Fabio). Por outro lado, por mais que os encontros com os grupos 1 (Adriana) e 2 (Marcos) ocorressem em uma sala onde não haviam outros atendimentos, muitas vezes os participantes eram afetados pelos ruídos provenientes de fora da sala, como se percebe no trecho a seguir.

ADRIANA: Hum... ai! Eu só não tô botando... é... interrogação, por meramente não lembrar onde que ela fica, mas é só por causa disso, eu entendo que deveria botar uma interrogação ali. Mas finge que tem! Vamos fingir que ela tá entre nós, a interrogação!

Pesquisador: [risos]

ADRIANA: Hum... ler... tá!

[nesse momento a participante faz uma reclamação em relação ao barulho fora da sala]

De acordo com Dreossi e Momensohn-Santos (2005), “a necessidade de manter-se fixado em um estímulo apesar do ruído tende a desenvolver um enorme cansaço e desgaste no jovem”. Com isso, ele passa a ter dificuldades para manter sua atenção, podendo ocorrer falhas na aprendizagem e, inclusive, sintomas físicos. Desse modo, é possível inferir que os ruídos existentes tenham influenciado negativamente no aprendizado dos participantes.

C6B: Problemas relacionados à organização dos encontros

Já no que se refere à organização dos encontros, o primeiro ponto identificado reside no tempo destinado à realização das atividades avaliativas (Encontro 6 e Encontro 7). Inicialmente, havia sido previsto o intervalo de uma hora para que os participantes pudessem construir dois programas que resolvessem duas situações problema previamente estabelecidas. No entanto, na maior parte dos casos observou-se que o tempo não foi suficiente para que os estudantes elaborassem os códigos.

O segundo ponto identificado em termos de organização tem origem no fato de que, em algumas ocasiões, foi necessário alongar o intervalo de tempo entre dois encontros consecutivos. Tal problema ocorreu em diferentes momentos, por motivos diversos, como: feriados, pontos facultativos (as datas de alguns encontros coincidiram com dias de jogos da seleção brasileira na Copa do Mundo de 2022), impossibilidade de participação dos estudantes, ou mesmo férias escolares (em virtude do prolongamento do tempo previsto para a sequência). Com isso, por vezes era necessário que um encontro ocorresse semanas, ou até mesmo meses após o anterior, o que comprometia a continuidade da aprendizagem. Tal fato foi claramente sentido e por vezes explicitamente colocado pelos participantes. O trecho a seguir ilustra uma passagem em que Helena aborda os dois problemas mencionados.

HELENA: Ah! Isso aí! Tem que ter mais uma variável, né?

Pesquisador: Olha pro seu problema e olha pras variáveis!

HELENA: Ai, que agonia! Tô agoniada! Ok... vamos fazer mais um, pelo amor de Deus [risos]. Vamos fazer mais um! Vamos enforçar essa aula só pra isso! **Por favor, que não vai dar certo fazer dois programas em meia hora** [risos]... variável... ler a... a... o a, que é a quilometragem vezes 3 b, vezes b... ai, meu Deus [risos]! Ai, Jesus! **É porque eu tô há muito tempo sem fazer, entendeu?**

Pesquisador: Uhum.

HELENA: Que eu lembro das coisas que eu fiz, eu lembro! Igualzinho o da... o da conta lá de internet, igualzinho da conta de internet.

C7: Problemas relacionados a aspectos emocionais

C7A: Participante expressa alguma emoção negativa

Em alguns casos, foram observadas manifestações de emoções negativas pelos participantes, como ansiedade, angústia, raiva ou frustração. Foi possível observar que, em ocasiões, tal fato era resultado de alguma dificuldade enfrentada no processo de construção do código, o que acaba levando à expressão de uma emoção negativa. Já em outros momentos os participantes demonstravam ansiedade e falta de concentração, por vezes antes mesmo do início do processo de programação, o que acabava impactando na capacidade de raciocínio e criação do participante.

Os trechos a seguir ilustram dois momentos distintos em que a participante Ana, após apresentar algumas dificuldades na construção de seus códigos, demonstra frustração e impaciência para dar continuidade ao processo.

ANA: Desisto, tchau, vou embora pra minha casa, beijos, foi bom estar com você, tá bom?
(...)

ANA: Tchau, vou embora, tá?

Pesquisador [se dirigindo à Ana]: Deixa eu ver aqui. Só vamos ver o que tá acontecendo aqui, Ana.
(...)

ANA: Ai, sair daqui, vou sair com a roupa da escola, ainda por cima. Ai, triste...
(...)

Pesquisador [se dirigindo à Ana]: É porque o seu não tava com a extensão, Helena...

ANA: Helena?

Pesquisador: Helena não... Ana.

ANA: Pô! Ah, não!

(...)

Pesquisador [se dirigindo à Ana]: Então agora vê se vai dar certo aí mesmo!

(...)

Pesquisador [se dirigindo à Ana]: Tenta aí de novo, Ana. Ila...

ANA: Ah, não... desisto!

Pesquisador: Só pra ver se vai funcionar, o seu...

(...)

ANA: Se não funcionar eu vou embora de verdade.

Pesquisador: Enter. Terminou o programa, não terminou?

ANA: Sim.

Pesquisador: O que que a gente...

ANA: Fim. Fim! Tchou! Fim! Não te aguento mais, computador!

Pesquisador: O seu já tá tudo certinho?

ANA: Sim.

Pesquisador: Então o que você tem que fazer agora?

ANA: Não sei [risos]! Não sei, realmente! Não, sério, aqui e aqui, né?

(...)

ANA: Faz pra mim que eu tô com dor nas costas!

Pesquisador: Só voltar lá, ó! Bota a setinha pra cima. Até lá no começo. Aí, aí, aí! Aí apaga essas três primeiras linhas porque elas não fazem parte do programa, elas só são o...

ANA: Usuário, coloque um valor, por favor, tô te pedindo [risos].

Pesquisador: Isso aí é só o algoritmo que você tinha montado.

ANA: Ai, professor, faz para mim, senão vou acabar perdendo isso aqui tudo.

Já os trechos a seguir se referem a momentos em que a participante Helena demonstra nervosismo e ansiedade no processo de construção do código. Tal fato pode ser reflexo de uma cultura escolar que muitas vezes resume o desempenho acadêmico à realização de provas e à obtenção de bons resultados em termos numéricos. Por mais que a participante tivesse sido tranquilizada no sentido de que não seria atribuída uma nota para o seu desempenho, e que não havia problemas em apresentar dificuldades na realização de alguma tarefa, é possível notar sua apreensão. A estudante chega a pedir desculpas pelo insucesso na construção do código.

HELENA: Eu vou adicionar a variável x. Calma... foi? X... foi? Eu não tô enxergando... “oxi”! Ué, cadê o x? Cadê o x! Você viu que eu coloquei o x [risos]... apaga... espaço, x. Início. Tá. Não... eu botei errado! **Ó minha perna, ó! Tremendo** [risos]...

Pesquisador: Isso que você tá falando, você pode escrever aí e depois você tentar organizar o que que tem que vir antes, o que que tem que vir depois. Você pode ir escrevendo e depois você vai organizando. Mesma coisa do outro, você falou uma coisa interessante ali pro outro, que de fato vai estar em algum lugar. A questão é você organizar depois a linguagem, né?

HELENA: [risos] **Ai, Deus. Tô nervosa, olha, se eu tivesse estudado... eu podia ter estudado.** Ué, por que que é assim? Estranho. Tá... não cliquei em nada! Ai... ah...

HELENA: Ok! Começa com início? Não. Variável! Eu acho que é isso! **Deus, me ajuda! Deus!** Posso fechar?... É variável? Ah, não é variável [risos]! É variável? **Ai, Deus! Você gosta do meu desespero!** Ok... variável... é variável? Eu acho que é variável! Variável, espaço... aí eu tenho que colocar se é... a gente sempre usa numérico, mas eu esqueci o nome da outra! Qual é o nome da outra? É variável! Variável numérico, é isso! Eu não lembro! **Eu não lembro, Sandro, desculpa!** Não... não, não, não, não, não... hum... hum... eu acho que não é isso... é isso? É, tipo... depois que o sistema vai ter o... o numérico a, b... ah, tá, tá, tá, tá... eu acho que eu lembrei... calma! Apaga aqui, né? Não é aqui que apaga? Tá, eu acho que... tá, ok! Hum, pra baixo... ah, aqui! Tá, aí depois... eu coloco numérico... aí eu tenho que ter, tipo a, b, c... é... vamos colocar a, b por enquanto... a, b... tem que fazer alguma coisa? **Ai, meu Deus!** Será que eu tenho que colocar algum comando? Enfim, segue a vida... é... tá! Numérico a, b... eu tenho que criar um sistema... finge que você não é professor e me ajuda!

HELENA: **Que caos!** Não tô gostando disso... nossa, Sandro, você tem muita paciência!

Pesquisador: Por que?

HELENA: Porque você tá aí há uma hora sentado vendo eu tentar fazer um negócio que você me ensinou várias vezes, e eu esqueci!

Pesquisador: Não tem problema.

HELENA: **É claro que tem problema! Se tivesse valendo nota, eu tava reprovada.** Talvez! Vai que você é aquele professor que se colocar as fórmulas... você é esse professor?

Por fim, também foram observadas falas de participantes que denotam baixa autoestima, sentimentos de impotência em relação à Matemática, ou até mesmo uma possível sensação de incapacidade em virtude da deficiência visual. Os trechos seguintes ilustram situações neste sentido.

ANA: **Eu me perdi, tudo eu me perco.**

HELENA: Eu vou resolver um problema ou eu vou criar um... um... ah, não! **Eu não sou boa em matemática!**

Pesquisador: [risos]... mas é um problema simples. Volta lá pro bloco de notas...

HELENA: **Olha só, eu sou ruim... eu sou ruim com 2 mais 2, você não tá entendendo.**

Pesquisador: Se quiser dar uma olhada no de cima também.

HELENA: **Você pode me ajudar, você tá aqui! Você tá enxergando!** [risos]

Pesquisador: Você tava fazendo um negócio aí, que você tava fazendo antes, o que era?

HELENA: O que eu tava fazendo antes? Eu não tava fazendo nada antes. **Eu tava fazendo uma coisa horrível antes, um trabalho péssimo.**

É possível que em virtude de dificuldades, fracassos e experiências negativas anteriores, os estudantes já tenham desenvolvido barreiras em relação à Matemática ou, de maneira mais ampla, em relação ao próprio ato de aprender. O insucesso vivenciado em experiências anteriores e seus efeitos sobre a autoestima dos estudantes podem, por sua vez, dificultar o desenvolvimento de novas aprendizagens. Como apontam Horta e Ferreira (2021), o desempenho dos estudantes é afetado por uma série de fatores cognitivos, relacionais e emocionais. Além disso, a autoestima desempenha papel crucial na aprendizagem e socialização dos alunos. Por mais que o sucesso acadêmico seja tratado como um indicador de qualidade da vida escolar, o bem estar psicológico, que tem relação direta com a autoestima, é um elemento chave para o desenvolvimento dos estudantes.

Por fim, cabe destacar que em alguns casos foi possível observar a manifestação de impaciência ou insatisfação direcionada especificamente a alguma característica do ILA. A necessidade de realizar o procedimento para executar o programa através do *prompt* de comando repetidas vezes, por exemplo, podia tornar o processo de depuração

enfadonho, trazendo um sentimento de frustração e impaciência. Tal fato pode ser observado nos trechos a seguir.

Pesquisador: Tem certeza?

ADRIANA: Tenho.

Pesquisador: Bota pra rodar aí de novo.

ADRIANA: Ai, repetir o processo de novo? Brincadeira não, hein! Tô com saudade do “Visual Code”! Reclamei tanto dele, mas ele pelo menos completava...

HELENA: Poxa, mas muito ruim, né, ter que ficar fazendo várias...

C8: Problemas relacionados a conceitos matemáticos

C8A: Dificuldades com operações básicas

No que se refere aos problemas relacionados a conceitos matemáticos, foi possível observar, em algumas situações, dificuldades na realização de operações matemáticas básicas, como multiplicação e potenciação. Tal fato, por vezes, representou um dificultador na condução da sequência, uma vez que era esperado que, nesta etapa de escolarização, os participantes dominassem tais operações.

Pesquisador: O que o computador vai ter que dar como resposta?

ANA: O dobro do 18.

Pesquisador: Qual é o dobro do 18?

ANA: Nem eu sei! [risos]

Pesquisador: Então tenta fazer um teste com um número que você sabe, pra ver se vai dar certo mesmo.

ANA: Então apaga esse?

Pesquisador: Apaga.

ANA: O dobro de... o dobro... é duas vezes a mais, né?

Pesquisador: Isso! Duas vezes o número.

ANA: 5, então.

Pesquisador: Vai ter que ser quanto?

ANA: 5 vezes 5, que vai dar 25.

Pesquisador: Mas é duas vezes. Você tá calculando o quadrado, você tá calculando ele vezes ele. Tem que ser duas vezes o número.

ANA: Como? Eu não sei!

Pesquisador: Ué, se for 5... o dobro de 5 é...

ANA: 5. Ai, sei lá, não sei!

C8B: Dificuldades relacionadas à noção de função

A maior parte dos problemas relacionados a conceitos matemáticos envolvia a noção de função e, mais especificamente, a determinação de uma função que modelasse o problema proposto. Em alguns casos, o participante parecia ter compreendido o problema, mas apresentava dificuldade para expressar as operações a serem realizadas através de uma função. O trecho a seguir, por exemplo, se refere a um problema em que deveria ser construído um programa que retornasse o valor total de uma corrida de táxi em que se cobrava R\$6,00 pela bandeirada e R\$3,00 por quilômetro percorrido. No trecho, Helena parece entender que a quantidade de quilômetros deveria ser multiplicada por 3 e, em seguida, deveriam ser somadas 6 unidades ao resultado da multiplicação. No entanto, a participante tem dificuldades para utilizar as variáveis e as constantes do problema para construir uma função.

HELENA: A aqui é a quilometragem. A é a quilometragem. Ai, meu Deus! A é a quilometragem, e aí eu preciso fazer a quilometragem vezes 3 mais 6. Ai... a quilometragem vezes 3 mais 6... a quilometragem vezes...

Pesquisador: Como é que você coloca isso no programa agora?

HELENA: A quilometragem soma... a quilometragem vezes 3. Eu posso colocar número? Eu posso fazer assim?

O trecho a seguir é outro exemplo em que um participante apresenta dificuldades para construir adequadamente a função que modela o problema. Inicialmente, Renata parece ter compreendido que deverá multiplicar o valor fornecido pelo usuário por 2. No entanto, enfrenta dificuldades para representar o fato através de uma relação matemática, usando as variáveis declaradas de maneira apropriada ($b=2.a$).

RENATA: E aí eu vou botar embaixo escrever “a vezes 2”?

Pesquisador: Agora não. Agora você vai dizer quem é essa variável. Qual é a outra variável que você tá usando aí?

RENATA: B.

Pesquisador: B. Então b vai ser igual a o quê?

RENATA: 2?

Pesquisador: 2 não. Qual é a multiplicação que você tá fazendo?

RENATA: 2 vezes alguma coisa.

Pesquisador: E que coisa é essa? Qual é a variável que você usou?

RENATA: Vezes.

Pesquisador: A variável, vezes é operação.

RENATA: Eu tô usando a letra “a” e a letra “b”.

Pesquisador: Aham. E aí o “b” vai ser igual a...

RENATA: O número que a pessoa colocar.

Pesquisador: E qual é a conta que você vai fazer para achar esse número? Quem vezes quem?

RENATA: 2 vezes...

Pesquisador: 2 vezes...

RENATA: O número!

Pesquisador: Que você chamou de?

RENATA: B.

Pesquisador: B é o resultado. Qual foi o outro número que você tinha usado? A outra letra?

RENATA: A.

Pesquisador: A. Então...

RENATA: 2 vezes a.

Pesquisador: Isso. Então b igual...

RENATA: Vai então... hum... eu vou botar embaixo “b” igual a 2 vezes “a”?

Pesquisador: Isso. Isso aí!

O principal problema mencionado nesta categoria se relaciona com um fato já discutido nas análises preliminares: a dificuldade que os estudantes apresentam em relação à ideia de generalização. Na medida em que os participantes da pesquisa muitas vezes não conseguiam chegar, com facilidade e de maneira autônoma, a uma função que modelasse o problema proposto, a mobilização da noção de generalização não ocorria de modo adequado. Desse modo, nesse aspecto os resultados da pesquisa dialogam com os apresentados por Lorencini, Nogueira e Rezende (2020), Rezende, Nogueira e Calado (2020), Silva e Nogueira (2022) e Calado e Rezende (2022).

C8C: Dificuldades relacionadas à resolução de problemas

Em menor número, também foram identificadas dificuldades em relação ao próprio ato de resolver problemas de Matemática. Nesse sentido, em geral os estudantes apresentavam falhas na compreensão de algum aspecto do problema. O trecho a seguir, por exemplo, remete ao processo de resolução de um problema em que deveria ser construído um programa que retornasse a renda mensal dos vendedores de uma loja. Tal renda era composta por uma parte fixa de R\$1500, e uma parte variável que dependia da quantidade de tapetes que o vendedor conseguisse vender. Para o cálculo da parte variável, ficou estabelecido que cada vendedor ganharia uma comissão de R\$16 por tapete vendido e, além disso, se o vendedor conseguisse vender mais de 100 tapetes no mês, ganharia adicionalmente um bônus de R\$300.

HELENA: Eu apaguei o que eu fiz? [risos]... Ai, tudo bem! Aonde que eu tô escrevendo isso, meu Deus? Hum... tá! Mais 1.... mais 16... ou vai ser de vezes? Mais... não! Não, não é 16, porque se o número for maior que 100, ele ganha 300. Se for menor que 100, ele não ganha nada. Brincadeira, ganha sim [risos]. Tipo, eu não entendi. Ele vai ganhar 16 por cada tapete, mais 300 se ele vender 100?

Pesquisador: Volta lá!

HELENA: Por que você não pode falar? Você que criou o negócio!

HELENA: [mexe no computador]

HELENA: **Eu ainda não entendi se ele vai cortar os 16 e dar os 300 ou se ele vai dar os 300 mais os... não, dar os 16 menos os 300. Eu não entendi! Ô, Jesus!**

A fala de Helena sugere que a estudante não compreendeu que, além da comissão de R\$16 por tapete vendido, seria dado o bônus de R\$300. De modo análogo, em outros casos os participantes tiveram dúvidas ou cometeram equívocos provenientes de uma interpretação equivocada dos elementos descritos na situação problema.

Síntese dos desafios identificados a partir da condução da sequência didática

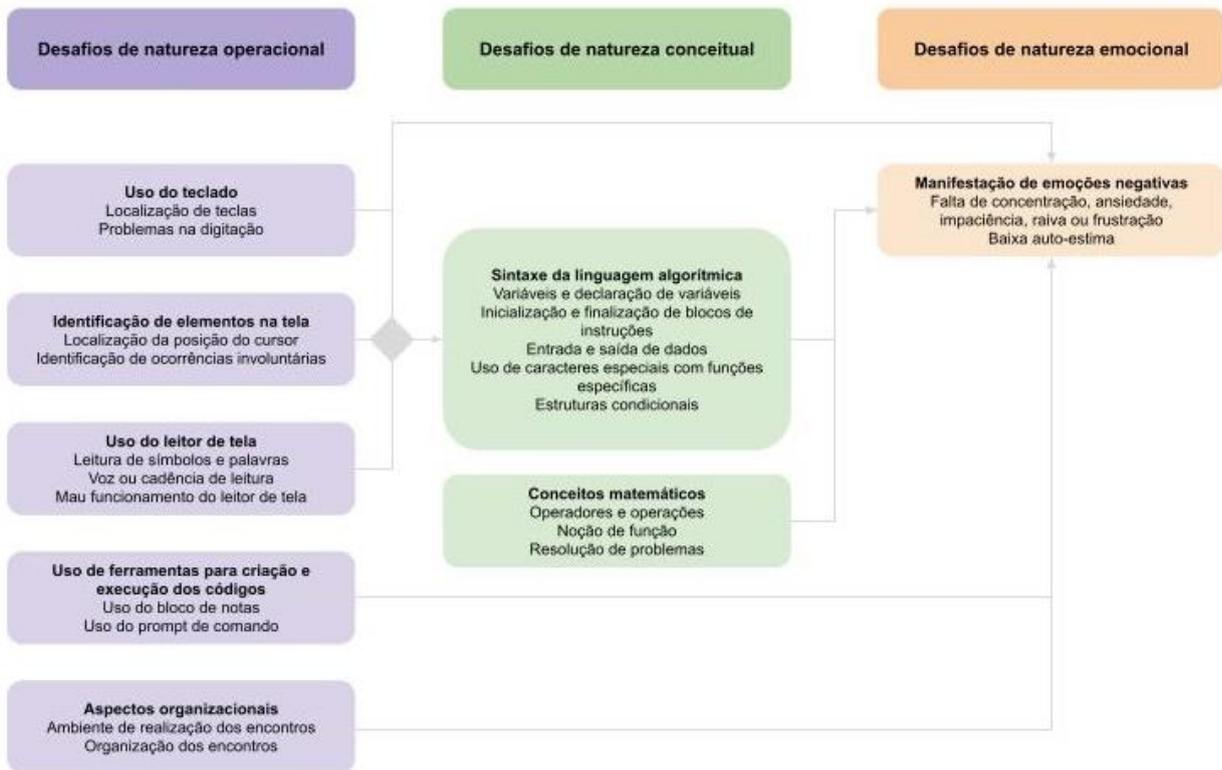
Por fim, a Figura 58 representa uma síntese dos desafios identificados ao longo dos encontros realizados. Também são exibidas as relações entre as categorias de dificuldades apresentadas pelos participantes. Entre os desafios de natureza operacional, identificamos o uso do teclado, a identificação de elementos na tela do computador, o uso do leitor de tela, o uso de ferramentas para criação e execução de códigos, e os aspectos organizacionais dos encontros. Já no que se refere aos desafios de natureza conceitual, foram identificadas a sintaxe da linguagem algorítmica e a mobilização de conceitos matemáticos. Por fim, entre os desafios de natureza emocional, identificamos a manifestação de emoções negativas.

De maneira geral, as dificuldades que os participantes apresentaram em relação ao uso do teclado, à identificação de elementos na tela do computador e ao uso do leitor de tela acabavam culminando na ocorrência de equívocos relacionados à sintaxe da linguagem algorítmica. Isto se justifica pelo fato de que muitos dos erros sintáticos cometidos eram fruto de uma escrita equivocada de algum comando, da não identificação de algum problema visualmente claro no código, ou mesmo por conta da omissão de elementos na leitura realizada pelo leitor de tela.

Além disso, muitas vezes as dificuldades de natureza operacional ou conceitual tinham como consequência um problema de natureza emocional. Em diversos momentos, por exemplo, os participantes manifestaram impaciência, frustração, ou até mesmo raiva pelo fato de não conseguirem localizar algum elemento na tela do computador, cometerem sucessivos erros de digitação ou apresentarem dificuldades

com o prompt de comando. De modo análogo, em muitos casos os participantes aparentavam ansiedade ou baixa auto-estima pelo fato de apresentarem dificuldades com conceitos matemáticos ou aspectos sintáticos da linguagem de programação.

Figura 58: Desafios para o desenvolvimento do pensamento computacional em estudantes com deficiência visual através do uso de uma linguagem de programação algorítmica.



Fonte: elaborado pelo autor (2024).

Potencialidades no desenvolvimento de conceitos e habilidades do pensamento computacional

Para além dos desafios apresentados anteriormente, a condução da sequência didática possibilitou identificar uma série de situações onde conceitos e habilidades do pensamento computacional foram mobilizados. São apresentados, a seguir, elementos relacionados às cinco habilidades fundamentais referidas por Czismadia et al. (2015).

Abstração

Em diversos momentos, por exemplo, os estudantes precisaram identificar as informações relevantes de um problema, colocando em segundo plano elementos que

não eram de fato necessários para a modelagem do fenômeno observado e construção de uma solução. Em particular, as situações-problema elaboradas para os encontros avaliativos (Encontro 6 e Encontro 7) foram criadas com o objetivo de demandar dos estudantes a realização de abstrações. No Quadro 55 são exibidos os problemas referidos e, além disso, são destacadas as informações de fato essenciais para a construção do código solicitado.

Quadro 55: Situações-problema dos encontros avaliativos, destacando-se as informações relevantes para a solução.

Situação-problema 1	Situação-problema 2
<p>Nos últimos anos, aplicativos para transportes de passageiros em automóveis privados têm se multiplicado, oferecendo serviços voltados tanto para o público geral, quanto para grupos específicos. Ainda assim, em virtude de problemas como excesso de cancelamento de viagens e aumento de tarifas, os táxis por vezes continuam representando uma opção atrativa no transporte de passageiros. Na cidade de Taxilandia, por exemplo, a corrida de táxi é calculada da seguinte maneira:</p> <ul style="list-style-type: none"> - É cobrada a taxa de R\$6 referente à bandeirada (valor fixo inicial de qualquer corrida), independente do horário em que seja realizada a corrida; - É cobrado o valor adicional de R\$3 por quilômetro rodado. <p>Nessas condições, construa um programa que determine o valor a ser pago pelo usuário de acordo com a quantidade de quilômetros rodados.</p>	<p>A comissão é um benefício muitas vezes utilizado no setor comercial para motivar os empregados e, conseqüentemente, aumentar os lucros da empresa. Tulio é proprietário de uma loja de tapetes, e para tentar impulsionar as vendas no período de Natal, decidiu conceder, além das comissões, um bônus aos funcionários que atingissem a meta estabelecida para a quantidade de tapetes vendidos. Assim, a renda mensal dos vendedores de sua loja passou a ser composta por uma parte fixa de R\$1500, e uma parte variável que dependia da quantidade de tapetes que o vendedor conseguisse vender. Para o cálculo da parte variável, ficou estabelecido que cada vendedor ganharia uma comissão de R\$ 16 por tapete vendido e, além disso, se o vendedor conseguisse vender mais de 100 tapetes no mês, ganharia adicionalmente um bônus de R\$300.</p> <p>Nessas condições, construa um programa que Tulio poderia utilizar para calcular o salário dos vendedores da loja de acordo com a quantidade de tapetes vendidos.</p>
Situação-problema 3	Situação-problema 4
<p>Mariana ganhou de sua mãe como presente de Natal um cofre que já continha uma nota de R\$200. Animada com o presente, decidiu, a partir do mês seguinte, depositar R\$35 no cofre a cada mês. Construa um programa que determine quantos reais haverá no cofre de acordo com a quantidade de meses passados.</p>	<p>Academias de ginástica e musculação têm se espalhado por todo o país, passando a fazer parte do cotidiano de muitos brasileiros. No entanto, é comum encontrar pessoas que inicialmente se engajam na realização de atividades físicas mas, com o passar o tempo, passam a se sentir desmotivadas e abandonam a prática. Pensando em atrair novos clientes e estimular a realização de exercícios físicos entre eles, uma academia lançou a seguinte promoção:</p> <ul style="list-style-type: none"> - Se o cliente frequentar a academia menos que 10 horas no mês, deve pagar apenas um valor fixo de R\$80. - Se o cliente frequentar a academia 10 ou mais horas no mês, o valor fixo cai para R\$30, e é cobrado um adicional de R\$2 por cada hora utilizada na academia. <p>Construa um programa que determine o valor a ser pago pelo cliente de acordo com a quantidade de horas frequentadas na academia.</p>

Fonte: elaborado pelo autor (2024).

A exploração do corpus de análise permitiu identificar diversos momentos em que os participantes buscavam selecionar as informações que seriam de fato utilizadas na construção da função objetivada. Os trechos a seguir ilustram exemplos neste sentido.

Pesquisador: Entendeu o problema, Adriana?

ADRIANA: Tá, eu acho que eu... tô entendendo...

Pesquisador: Então você tem...

ADRIANA: Ele tem uma taxa fixa...

Pesquisador: Uhum...

ADRIANA: De 2, é... ele paga 2 reais fixo assim de megabytes.

Pesquisador: Exato.

ADRIANA: Se você consumir... é... até... até... “caraca”, eu acabei de ver o valor do megabyte e já esqueci... **40 megabytes, você paga 50 reais. Agora, se você consumir mais ou menos... mais ou menos, né, você continua pagando 2 reais.**

Pesquisador: Uhum.

ADRIANA: Isso que eu entendi, não sei se tá muito viajado.

HELENA: (...) Ok... ler a... o a não é o 1500, o 1500 tem que estar lá no final. Porque... olha só, eu preciso colocar um número que... eu preciso colocar um número... é assim, eu vou digitar o número de tapetes vendidos. Aí o a é o número de tapetes vendidos. O número de tapetes vendidos vai dizer se ele vai ganhar 300 ou não.

HELENA: Não terminei! Aqui... ok, Sandro... não sei não... ler a... aí tipo, ela tem 200 e 35. No caso, 35 é o número que ela vai estar somando a cada mês. Ela tem 200 mais o mês vezes 35. (...)

Decomposição

Conforme argumentam Czismadia et al. (2015), a decomposição é um modo de se pensar na resolução de um problema em termos de suas partes componentes. Com isso, problemas complexos se tornam mais fáceis de se resolver. Nessa perspectiva, esta habilidade do pensamento computacional também foi mobilizada pelos participantes em diversos momentos ao longo dos encontros, na medida em que muitos dos problemas propostos precisavam ser pensados em termos de cenários possíveis.

Os programas a seguir foram construídos pelas participantes Adriana e Helena, respectivamente, já tendo sido mencionados na análise a posteriori do encontro 5. Por mais que as estudantes tenham utilizado estruturas condicionais diferentes para a resolução, em ambos os casos foi necessário analisar o problema em termos de suas partes componentes. Se o número de megabytes consumido fosse menor ou igual a 40, o cálculo do valor total a ser pago se daria de uma forma. Já se fosse consumida uma

quantidade superior a 40 megabytes, um cálculo diferente deveria ser realizado para se chegar à solução esperada. Dessa forma, era necessário decompor o problema em partes menores, utilizando uma estratégia de resolução diferente para cada caso.

```

variaveis
numerico a, b
inicio
escrever "Digite o numero de MB consumido"
ler a
se (a<=40) entao
b=a*2+50
escrever "O nuomero de MB consumido custara ",b
senao
b=a*2+40
escrever "O numero de MB consumido custara",b
fim_se
fim

```

```

variaveis
numerico a, b
inicio
escrever "digite a quantidade"
ler a
se (a<=40) entao
b=50+2*a
escrever b
fim_se
se (a>40) entao
b=40+2*a
escrever b
fim_se
fim

```

Em geral, códigos que demandam a utilização de estruturas condicionais em sua construção estão diretamente associados com a ideia de decomposição. Desse modo, situações-problema que envolvam esse tipo de estrutura apresentam potencialidades significativas para a mobilização desta habilidade do pensamento computacional.

Reconhecimento de padrões (generalização)

Para Czismadia et al. (2015), a habilidade de generalização está associada à identificação e exploração de padrões, semelhanças e conexões já que, de maneira geral, é possível resolver problemas de forma mais rápida recorrendo-se a soluções encontradas para problemas anteriores. De acordo com os autores, fazer perguntas como "*Esse problema é semelhante a algum outro que eu já resolvi?*" ou "*Como esse problema é diferente?*" são estratégias importantes no processo de resolução.

A análise das transcrições dos áudios dos encontros permitiu identificar que variações das perguntas mencionadas anteriormente foram realizadas diversas vezes

pelos participantes. Com grande frequência, participantes buscavam se recordar de problemas semelhantes resolvidos anteriormente no intuito de encontrar soluções possíveis para um novo problema. Dessa forma, os trechos a seguir são exemplos de situações nas quais os estudantes mobilizam a habilidade de generalização.

HELENA: Calma! A gente fez... a gente fez um que era mais ou menos esse! Não tá aqui!

Pesquisador: Aqui tá os que a gente trabalhou, sem ser os que você construiu! Os que você construiu não tão aí.

HELENA: Hum... ah!

Pesquisador: Aqueles primeiros que a gente viu em cada aula.

HELENA: Eu construí um que era igualzinho! Ai, você é cruel [risos]! Você é muito cruel. Eu construí um igualzinho, era só copiar e colar.

Pesquisador: Aí fica muito fácil.

HELENA: Cara, eu tô lembrando, acho que foi o último que a gente fez, que tinha lá, que tinha lá aquele problema lá, que... ai, meu Deus, era um negócio de internet!

Pesquisador: Hum...

HELENA: Lembra?

Pesquisador: Uhum.

HELENA: Como é que eu vou identificar ali que o 200 já é fixo? Aquele negócio que a gente fez! Tá, mas eu coloquei isso antes ou depois? Hum... eu coloco antes ou depois? Eu lembro que tinha que fazer um negócio de, tipo assim, colocar 200 mais a mais... dividido, sei lá, tipo... depende da conta! Isso é antes ou depois [risos]?

ADRIANA: Eu... é... eu escrevia: se o número... aqui tá aqui, se o número é maior, menor do que 10...

Pesquisador: Uhum.

ADRIANA: Agora... é... então... não, per aí... **calma aí, eu tô tentando lembrar o nome em português, calma aí.... senão, acho que seria senão.**

É possível perceber que Helena, em momentos distintos, reconhece similaridades entre o problema que tenta resolver e outros problemas resolvidos anteriormente. A partir de tal reconhecimento, a estudante parece tentar se lembrar de alguma estratégia já utilizada, que poderia ser aplicada ao problema atual. Adriana, por sua vez, parece se recordar de alguma estrutura condicional já utilizada em outra linguagem de programação

Algoritmização

A habilidade de construção de algoritmos foi mobilizada pelos participantes em todos os encontros, na medida em que ao fim de cada sessão era solicitado que os

estudantes construíssem um programa que atendesse a algum objetivo específico. Após definir as informações relevantes para se chegar à solução do problema proposto, os participantes deveriam, então, elaborar uma função que modelasse a situação observada, convertendo as informações em linguagem natural para a linguagem matemática. Em seguida, os participantes deveriam fazer nova operação de conversão, desta vez da linguagem matemática para a algorítmica. Este último processo se relaciona diretamente com a habilidade de construção de algoritmos.

No trecho a seguir é exibido um momento em que a habilidade de algoritmização é mobilizada pelo participante Marcos. No caso, o estudante deveria realizar as alterações necessárias em um programa para que fosse solicitado um novo tipo de dado ao usuário e, além disso, fosse dada uma nova resposta como saída. Marcos compreende que, em virtude da mudança no tipo de dado solicitado (idade ao invés do nome), a estrutura da frase de retorno também deveria ser alterada. Com isso, realiza as mudanças necessárias no algoritmo do programa, conforme ilustrado no Quadro 56.

MARCOS: Eu vou ter que colocar, no caso, é... a minha idade é tanto...

Pesquisador: Isso aí.

MARCOS: Ah, tá, entendi. Tá! Nesse caso, eu acho que pode... não, não pode jogar a variável antes... ou... não, não pode jogar a variável antes porque pode... é, se jogar antes vai falar no início da palavra.

Pesquisador: Uhum.

MARCOS: Então o mais sensato a fazer aqui é jogar ela depois da frase.

Pesquisador: Isso aí.

MARCOS: Tá. Vou abrir umas aspas aqui. Coloca aqui... aí aqui fecha aspas. Espacinho breve. Aí aqui... e aqui o fim. Ah, tá, só que tem que jogar essa linha aqui pra não atrapalhar aquela ali debaixo.

Pesquisador: Uhum.

Quadro 56: Alteração realizada pelo participante Marcos.

Programa inicial	Alteração realizada
<pre> variaveis caracter nome inicio escrever "Digite seu nome" ler nome escrever nome, "e estudante do Colegio Pedro II" fim </pre>	<pre> variaveis caracter idade inicio escrever "Digite sua idade" ler idade escrever "minha idade e ", idade fim </pre>

Fonte: elaborado pelo autor (2024).

Já no trecho ilustrado a seguir, Adriana expõe o processo de pensamento empregado na construção de um código no primeiro encontro avaliativo. Por mais que a

participante tenha compreendido adequadamente o problema e estabelecido uma relação correta entre as variáveis identificadas, por si só, isso não seria suficiente para a construção do programa solicitado. Era preciso que, compreendido o problema e identificada uma solução possível, a participante fosse capaz de criar um algoritmo que de fato traduzisse a resolução imaginada para uma linguagem algorítmica. De fato, Adriana consegue fazer tal conversão com sucesso, como se evidencia no código exibido logo em seguida. Além disso, a participante utiliza corretamente os comandos do ILA para solicitar a informação necessária do usuário, e retornar uma mensagem adequada de acordo com o que havia sido solicitado na situação-problema.

ADRIANA: Eu atribui a variável “a” à quantidade de quilômetros.

Pesquisador: Uhum.

ADRIANA: E a variável “b” eu atribui à... à quantidade de quilômetros que “a” fosse rodado, assim, se fosse 6, 4, assim, seria esse valor vezes cada quilômetro.

Pesquisador: Certo.

ADRIANA: Rodado, que tem um valor fixo de 3 reais.

Pesquisador: Uhum.

ADRIANA: Somado a isso, a bandeirada, que eu entendi lá, era de 6 reais, então tem que ser o valor “a” vezes a quantidade de quilômetros rodados, vezes... é, mais o 6.

```
variaveis
numerico a, b
inicio
escrever"Quantos K foi a corrida"
ler a
b=a*3+6
escrever "O valor da corrida",b
fim
```

Avaliação

Outra habilidade do pensamento computacional mobilizada com grande frequência ao longo dos encontros foi a de avaliação. No primeiro encontro da sequência, os participantes foram orientados a sempre realizarem um teste para verificar se os programas construídos funcionavam adequadamente. Nos encontros seguintes, em geral, os estudantes se recordavam da necessidade de realizar o teste ao fim do processo de construção do código. Além disso, compreendiam que, em alguns casos, deveriam ser feitos diferentes testes para garantir que não havia problemas com o código construído.

No trecho seguinte, por exemplo, Adriana identifica a necessidade de fazer mais de um teste, de modo que fosse possível verificar se o programa estava funcionando adequadamente para todos os cenários possíveis no problema proposto.

ADRIANA:(...) Deu certo.

Pesquisador: Deu certo.

ADRIANA: Tem que fazer outros testes?

Pesquisador: Oi?

ADRIANA: Tem que fazer o teste das duas coisas?

Pesquisador: O que você acha? O que que você faria nesse programa?

ADRIANA: Hum, eu poderia comprar 4 cadernos, poderia comprar, é... 25 cadernos, se decidisse.

Pesquisador: Perfeito. Qual que você acha que seria o teste ideal ou os testes ideais aí nesse caso?

ADRIANA: Comprar 4 cadernos que aí...

Pesquisador: Show. Então bota aí.

ADRIANA: Vamos lá de novo! Dá um enter... ah, não! No último dia eu vou ficar trocando a letra “l” pela letra “k” [risos]... eita! Xi! Calma aí que eu vou consertar... é 26... 26.

Pesquisador: E aí, deu certo?

ADRIANA: Deu certo.

Pesquisador: Deu certo, né? Então você usou dois testes, né? Quais números que você usou nos seus testes?

ADRIANA: Eu usei 2 e 4.

Pesquisador: O que aconteceu quando você usou o 2?

ADRIANA: Apareceu o número 16. Que a compra ficou... cada caderno ficou a 8.

Pesquisador: Perfeito. E no outro teste que você fez?

ADRIANA: Apareceu 26 reais. A compra cada caderno ficou a 6 e 50.

Pesquisador: Perfeito. Então é isso, seu programa tá funcionando direitinho?

ADRIANA: Tá funcionando direitinho!

Pesquisador: Show! Beleza, então é isso!

Outra habilidade relacionada à de avaliação que foi bastante mobilizada nos encontros foi a de depuração. Muitas vezes os participantes cometiam algum erro na construção do código, mas apenas o identificavam após tentarem executar o programa. Sempre que algum equívoco era cometido, uma mensagem de erro era exibida ao participante, indicando em que linha havia se encontrado algum problema. O participante, então, retornava ao código para identificar a linha referida e tentar descobrir a natureza do erro cometido.

Os trechos a seguir representam momentos em que as participantes Helena e Adriana realizam testes que indicam a ocorrência de algum tipo de erro. No caso de Helena, o equívoco cometido fazia com que o programa sequer rodasse. Após a identificação da linha na qual se encontrava o problema, Helena conclui corretamente que o erro cometido residia na forma como havia escrito o comando “escrever” (e crever).

Já o programa construído por Adriana chega a funcionar, o que indica que não foram cometidos equívocos de natureza sintática. No entanto, era necessário que o

participante verificasse se o programa estava semanticamente adequado; ou seja, era preciso avaliar se o programa de fato realizava o que se esperava. Nesse ponto, Adriana identifica corretamente que o valor retornado pelo programa não corresponde ao esperado, uma vez que o objetivo era que se dobrasse um número fornecido pelo usuário. Como a participante digita o número 3, e obtém como resposta o número 9, identifica que ocorreu algum problema no processo de construção. Após analisar novamente o código construído, Adriana identifica o erro cometido e o corrige. Por fim realiza novo teste, concluindo desta vez que o programa funciona corretamente.

HELENA: Alguém me ajuda! Tá certo?

Pesquisador: O que ele falou?

HELENA: Erro na linha 4.

Pesquisador: Então vamos lá!

HELENA: Erro na linha 4... 1, 2... [risos]

Pesquisador: E aí, qual é o erro aí?

HELENA: Tá escrito [risos]...

Pesquisador: Tá escrito a quê?

HELENA: “Ecrever” [risos].

Pesquisador: Ecrever.

Pesquisador: Deu certo?

ADRIANA: A multiplicação é 9.

Pesquisador: O que esse programa tinha que calcular?

ADRIANA: Eita, pior que é! Tinha que dar 6, não 9.

Pesquisador: Então onde tá o problema?

ADRIANA: Ué, pera aí! “B” igual a asterisco “a”, faltou o 2.

(...)

Pesquisador: Agora foi. Dá um “enter”.

ADRIANA: Vai ser 5... 6... sem querer... 12, deu certo!

Pesquisador: Fechou, então?

ADRIANA: Fechou.

7. CONSIDERAÇÕES FINAIS

O trabalho com pessoas com deficiência envolve o respeito às suas limitações, mas também, e sobretudo, a valorização de suas potencialidades. Adaptar uma situação de aprendizagem com o objetivo de abarcar as diferenças e possibilitar a participação plena dos estudantes vai além de uma mera simplificação. Desse modo, a construção de cenários de aprendizagem inclusivos deve promover oportunidades para que todos os estudantes explorem e desenvolvam ao máximo seu potencial de forma criativa e colaborativa, compreendendo e respeitando as diferenças.

A aprendizagem da Matemática por estudantes com deficiência visual se insere nesse cenário. Embora não se possa negar a importância do material concreto na atribuição de significado aos objetos matemáticos, especialmente na educação de pessoas com deficiência visual, é preciso ir além. Em um contexto histórico-social marcado por um desenvolvimento tecnológico acelerado, é preciso indagar, por exemplo, se um objeto de aprendizagem desenvolvido através de tecnologias digitais é acessível a pessoas com deficiência visual. Se uma situação de aprendizagem na qual se utilizará um *software* específico poderá ser plenamente explorada por uma pessoa cega. Se barreiras à aprendizagem e à participação em espaços virtuais explicam a invisibilidade de pessoas com deficiência visual em áreas STEM.

Partindo de tais indagações, buscamos construir um cenário de aprendizagem desafiador, no qual estudantes com deficiência visual pudessem desenvolver habilidades de programação de computadores e de resolução de problemas de forma plena. Considerando o público-alvo da pesquisa, formado por estudantes do Ensino Médio, optamos por nos limitarmos a problemas relacionados à função afim, conteúdo já explorado pelos participantes, além de promissor em termos de modelagem matemática.

Desejávamos, ainda, identificar potencialidades e limitações da programação de computadores enquanto recurso para o desenvolvimento de habilidades do pensamento computacional pelos participantes. Recorremos, então, a uma linguagem algorítmica baseada em português estruturado, dadas as vantagens que esse tipo de recurso apresenta pelo fato de os comandos utilizados na construção dos códigos serem escritos na língua natural do estudante. Era necessário, no entanto, planejar o cenário de aprendizagem de modo criterioso, científico, e para tal utilizamos os pressupostos da Engenharia Didática.

Enquanto referencial metodológico, a Engenharia Didática possibilitou a elaboração, condução e avaliação do cenário de aprendizagem de forma sistemática e rigorosa. Partindo de uma série de análises que precederam a fase de concepção, foi possível planejar ações e estabelecer previsões para o comportamento dos estudantes com base em aspectos epistemológicos, didáticos e cognitivos bem definidos. E em uma constante relação entre planejamento e prática pedagógica, ação e reflexão, utilizamos a própria produção didática como prática de pesquisa na fase de experimentação. Desse modo, o primeiro objetivo específico da tese (OE1) foi alcançado, na medida em que utilizando o referencial teórico da Engenharia Didática, foi possível criar uma sequência didática que possibilitasse utilizar a programação de computadores como recurso para a modelagem da função afim e o desenvolvimento do pensamento computacional em estudantes do Ensino Médio com deficiência visual.

Passamos, então, à condução da sequência didática planejada inicialmente, o que possibilitou o desenvolvimento de habilidades básicas de programação de computadores junto aos participantes, bem como a modelagem e resolução de problemas de Matemática relacionados com a função afim. Assim, o segundo e o terceiro objetivos específicos deste trabalho (OE2 e OE3) também foram atingidos. Além disso, a análise dos dados coletados durante a condução da sequência permitiu identificar uma série de potencialidades e limitações quanto ao uso da linguagem algorítmica ILA para o desenvolvimento do pensamento computacional em estudantes do Ensino Médio com deficiência visual. Desse modo, também consideramos alcançado o quarto objetivo específico do estudo (OE4).

Os dados coletados indicam que, assim como previsto na primeira hipótese de trabalho deste estudo (H1), o uso de uma linguagem algorítmica baseada em português estruturado apresenta vantagens enquanto recurso para que estudantes com deficiência visual desenvolvam habilidades do pensamento computacional. De fato, a análise de conteúdo das transcrições dos áudios dos encontros permitiu identificar diversos momentos em que eram mobilizadas as habilidades de abstração, decomposição, reconhecimento de padrões, algoritmização e avaliação. Além disso, nas entrevistas realizadas após o último encontro da sequência didática, todos os participantes entrevistados apontaram que a existência de comandos em língua portuguesa facilitava o processo de construção dos códigos. Porém, não podemos concluir que o uso deste recurso diminuiu a ocorrência de erros de sintaxe característicos de linguagens de programação textuais, já que erros desse tipo foram frequentes ao longo dos encontros.

No que se refere à segunda hipótese de trabalho do estudo (H2), de fato identificamos limitações relacionadas ao leitor de tela, na medida em que alguns caracteres não eram lidos adequadamente pelo recurso, fato que por vezes dificultava a construção dos códigos e, em alguns casos, induzia a interpretações equivocadas por parte dos participantes. No entanto, também foram identificados outros desafios importantes que devem ser considerados ao se utilizar uma linguagem algorítmica para a resolução de problemas e o desenvolvimento de pensamento computacional por estudantes com deficiência visual. Nesse ponto, a análise lexicográfica e a análise de conteúdo das transcrições dos áudios dos encontros foram essenciais para trazer à tona uma série de aspectos que uma simples observação do rendimento dos estudantes nos encontros avaliativos não seria capaz de revelar.

Em primeiro lugar, estudantes com deficiência visual podem apresentar dificuldades no manuseio do teclado do computador e na identificação de elementos importantes na tela do computador. Além disso, a leitura realizada pelo leitor de telas apresenta algumas inconsistências, principalmente em se tratando de caracteres especiais, o que pode levar a interpretações equivocadas. Todos esses fatores podem provocar uma maior ocorrência de erros de sintaxe na construção dos códigos. De fato, a própria sintaxe da linguagem algorítmica representa um desafio em particular, na medida em que são comuns as dificuldades na introdução aos conceitos de programação de computadores.

Outro desafio é o uso das ferramentas para edição e execução dos códigos. Pessoas com deficiência visual que não utilizam computadores com frequência podem ter dificuldades para acessar as funcionalidades do bloco de notas através do teclado. A necessidade de salvar os arquivos em formatos e locais específicos para que a execução possa ser realizada adequadamente representa um dificultador a mais. Além disso, o uso do *prompt* de comando para executar os programas pode se tornar uma tarefa exaustiva, tanto pela necessidade de memorizar uma série de comandos específicos, quanto pela exigência de se realizar o mesmo procedimento diversas vezes caso seja necessário corrigir erros no programa.

Por sua vez, a mobilização de conceitos matemáticos pode ser desafiadora para alguns estudantes, na medida em que muitos apresentam um histórico de dificuldades com a disciplina. Tanto o uso de notações e operadores específicos, quanto a necessidade de compreender os problemas propostos, construindo funções que

modelassem as situações apresentadas, foram pontos que, muitas vezes, trouxeram dificuldades para os estudantes. Consideramos, no entanto, que a dificuldade e o erro são elementos importantes no processo de aprendizagem, que devem ser explorados no sentido de criação de novos significados.

O planejamento dos encontros também deve ser realizado com cautela, considerando as especificidades do trabalho com estudantes com deficiência visual. É ideal que as atividades sejam conduzidas em um ambiente silencioso, na medida em que os estudantes dependem da audição para construir os códigos. Além disso, é preciso dispor de um intervalo de tempo adequado para a realização das atividades, já que por conta das dificuldades já mencionadas, como com o uso do teclado e das ferramentas para criação e execução dos códigos, o processo de criação dos programas tende a ser mais longo.

Um último desafio identificado se refere às questões de natureza emocional que podem emergir ao longo do processo. É possível que, em alguns momentos, os estudantes manifestem emoções negativas como ansiedade, impaciência, frustração e raiva, em geral por conta de erros cometidos ou dificuldades apresentadas. Embora tais emoções façam parte da experiência humana e possam, por vezes, surgir no contexto escolar, é importante que sejam evitadas de modo que não impactem negativamente o processo de aprendizagem. Pelo contrário, emoções positivas como a alegria e o encantamento, que também foram observadas em diversos momentos nas interações dos participantes com as atividades propostas e as funcionalidades da linguagem algorítmica, devem ser estimuladas.

Em suma, por mais que tenham surgido desafios ao longo dos encontros, nenhum dos pontos-chave identificados representou um impeditivo para a condução da sequência. Foi possível perceber que, de maneira geral, os estudantes puderam participar das atividades propostas com autonomia e qualidade. Além disso, mobilizaram uma série de conceitos e habilidades do pensamento computacional, como abstração, decomposição, reconhecimento de padrões, algoritmização e avaliação. Ademais, foi possível perceber, também, o uso de habilidades metacognitivas a todo instante que expunham seu processo de pensamento, ponderavam se um raciocínio era adequado ou avaliavam se uma solução encontrada era satisfatória. Desse modo, consideramos, por fim, que por mais que existam desafios a serem observados, é possível utilizar uma

linguagem algorítmica em português estruturado para desenvolver o pensamento computacional em estudantes do Ensino Médio com deficiência visual.

Limitações do estudo

Por mais que a pesquisa tenha possibilitado a obtenção de resultados satisfatórios, cabe destacar algumas limitações do trabalho. Em primeiro lugar, o número limitado de participantes que concluíram a sequência didática impediu a realização de uma análise mais ampla. Um maior quantitativo de participantes possibilitaria a expansão dos dados qualitativos obtidos, bem como a construção de um *corpus* textual mais robusto para a análise de conteúdo. Além disso, uma abordagem quantitativa poderia ser realizada em complementação à análise qualitativa conduzida.

Como já mencionado, questões organizacionais também acabaram tendo impacto na coleta de dados. Os ambientes disponíveis para condução dos experimentos muitas vezes eram ruidosos, não raro ocorrendo intervenções de pessoas alheias ao estudo. Tal fato pode ter impactado negativamente a capacidade de concentração e, em consequência, o rendimento dos participantes. Outro ponto de destaque é que por conta de imprevistos ocorridos ao longo do estudo, por vezes o intervalo de tempo entre os encontros foi demasiado longo, o que pode ter comprometido o processo de aprendizagem dos estudantes. Por fim, o tempo destinado à realização de algumas das atividades propostas também se mostrou inadequado. Tais fatores já foram abordados anteriormente, e devem ser considerados com cautela na hipótese de reprodutibilidade do estudo.

Perspectivas

O estudo realizado contribui para a construção de cenários de aprendizagem inclusivos, onde estudantes videntes e não videntes podem trabalhar colaborativamente utilizando os mesmos recursos para a modelagem e resolução de situações-problema. Além disso, apresenta uma proposta desafiadora e estimulante para estudantes com deficiência visual, privilegiando uma abordagem que valoriza potencialidades em lugar de evidenciar limitações.

Cabe, agora, expandir o estudo, identificando como outras linguagens de programação algorítmicas baseadas em português estruturado podem ser utilizadas

para o desenvolvimento de habilidades em estudantes com deficiência visual. É importante, também, trabalhar pela solução de problemas na leitura realizada por leitores de tela, em especial no que se refere a operadores e caracteres especiais, e pela eliminação de barreiras de acessibilidade que as linguagens de programação existentes, algorítmicas ou não, ainda apresentam.

REFERÊNCIAS

ABREU, L. V. F. de. TACTOPI: a playful approach to promote computational thinking to visually impaired children. 2021. 82 f. Dissertação (Mestrado) - Curso de Mestrado em Informática, Universidade de Lisboa, Lisboa, 2021.

AFARI, E.; KHINE, M. S. Robotics as an educational tool: impact of lego mindstorms. *International Journal of Information and Education Technology*, v. 7, p. 6, p. 437-442, 2017.

AHO, A. V. Computation and computational thinking. *Computer Journal*, n. 55, p. 832-835, 2012.

ALMEIDA, E. S.; COSTA, E. B.; SILVA, K. S.; PAES, R. B.; ALMEIDA, A. A. M., BRAGA, J. D. H. AMBAP: Um ambiente de apoio ao aprendizado de programação. In: Workshop sobre Educação em Computação, Florianópolis, 2002.

ALMOULOUD, S. A.; COUTINHO, C. Q. S. Engenharia Didática: características e seus usos em trabalhos apresentados no GT-19 / ANPEd. *Revista Eletrônica de Educação Matemática*, Florianópolis, v. 3, n. 1, p. 62-77, 2008.

ALMOULOUD, S. A.; DA SILVA, M. J. F. Engenharia didática: evolução e diversidade Didactic engineerinSara: evolution and diversity. *Revista Eletrônica de Educação Matemática*, Florianópolis, v. 7, n. 2, p. 22-52, 2012.

ANDRADE, T. M. (Ed.) Matemática interligada: funções afim, quadrática, exponencial e logarítmica. 1ª ed. São Paulo: Scipione, 2020.

ANDRÉ, C. F. O pensamento computacional como estratégia de aprendizagem, autoria digital e construção da cidadania. In: teccogs – Revista Digital de Tecnologias Cognitivas, n. 18, p. 94-109, jul./dez. 2018.

ANTUNES, C.; NETO, I.; CORREIA, F.; PAIVA, A.; NICOLAU, H. Inclusive'R'Stories: An Inclusive Storytelling Activity with an Emotional Robot. In: Proceedings of the 2022 ACM/IEEE INTERNATIONAL CONFERENCE ON HUMAN-ROBOT INTERACTION. Sapporo, Hokkaido, Japão, p. 90-100, 2022.

ARTIGUE, M. Ingénierie didactique. *Recherches en Didactique des Mathémaques*, v.9, n.3, p. 281 - 308, 1988.

BARBOSA, L. L.; COUTO, C. M. S.; TERRA, R. PortuCol: uma pseudolinguagem inspirada em C ANSI para o Ensino de Lógica de Programação e Algoritmos. In XXIV Workshop sobre Educação em Computação (WEI), p. 2343-2352, 2016.

BARCELOS, T. et al. Relações entre o pensamento computacional e a Matemática: uma revisão sistemática da literatura. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 4., 2015, Maceió. Anais... Maceió: Sociedade Brasileira de Computação, 2015.

BARDIN, L. Análise de conteúdo. Lisboa: Edições 70, 1977.

BARROS, J.; SILVA, A. O.; SILVA, G. L. da. Uma releitura histórico-epistemológica para o ensino do conceito de função. Boletim GEPEM, [S. l.], n. 79, p. 50–69, 2021.

BASSIL, Y. Phoenix – the arabic object-oriented programming language. International Journal of Computer Trends and Technology, 67(2), p. 7-11, 2019.

BBC LEARNING, B. What is computational thinking? 2015. Disponível em: <https://www.bbc.co.uk/bitesize/guides/zp92mp3/revision/1>. Acesso em jun 2024.

BELL, E. C. ; SILVERMAN, A. M. The impact of attitudes and access to mentors on the interest in STEM for teens and adults who are blind. J Blind Innovat Res. 8(2), p. 1-14, 2018.

BILABILA, A. M. CompAlg – Ferramenta de ensino e aprendizagem da lógica de programação. 2017. 108 f. Dissertação (Mestrado) - Curso de Mestrado em Ciência de Computadores, Universidade do Porto, Porto, 2017.

BITTAR, M. Contribuições da teoria das situações didáticas e da engenharia didática para discutir o ensino de matemática. In: TELES, R.; BORBA, R.; MONTEIRO, C. Investigações em Didática da Matemática. Editora Universitária UFPE, p. 100-131, 2017.

BLIKSTEIN, P. O pensamento computacional e a reinvenção do computador na educação. 2008. Disponível em: http://www.blikstein.com/paulo/documents/online/ol_pensamento_computacional.html. Acesso em: 15 de maio de 2024.

BORDINI, A. et al. Pensamento Computacional nos ensinos fundamental e médio: uma revisão sistemática. In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 6., 2017, Recife. Anais... Recife: Sociedade Brasileira de Computação, 2017.

BOTELHO, L; REZENDE, W. M. Um breve histórico do conceito de função. Caderno Dá Licença, Niterói, v. 6, p. 64-75, dez. 2007.

BRACKMANN, C. P. Desenvolvimento do pensamento computacional através de atividades desplugadas na educação básica. 2017. 226 f. Tese (Doutorado) - Curso de Informática na Educação, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2017.

BRACKMANN, C.; AUGUSTO, D. B.; CASALI, A.; HERNÁNDEZ, S. Pensamento computacional: Panorama nas américas. In: XVIII Simpósio Internacional de Informática Educativa, SIIIE. 2016. p. 197-202.

BRASIL. Decreto nº 5.296, de 2 de dezembro de 2004. Regulamenta as Leis n. 10.048, de 8 de novembro de 2000, que dá prioridade de atendimento às pessoas que especifica, e 10.098, de 19 de dezembro de 2000, que estabelece normas gerais e critérios básicos para a promoção da acessibilidade das pessoas portadoras de deficiência ou com mobilidade reduzida, e dá outras providências.

BRASIL. Lei nº 9.394, de 20 de dezembro de 1996. Estabelece as diretrizes e bases da educação nacional.

BRASIL. Lei nº 5.692, de 11 de agosto de 1971. Fixa Diretrizes e Bases para o ensino de 1º e 2º graus, e dá outras providências.

BRASIL. Lei nº 4.024, de 20 de dezembro de 1961. Fixa as Diretrizes e Bases da Educação Nacional.

BRASIL. Ministério da Educação. Secretaria de Educação Especial (SEESP). Política Nacional de Educação Especial na Perspectiva da Educação Inclusiva. Brasília: MEC/SEESP, 2008.

BRASIL. Lei nº 13.146, de 6 de julho de 2015. Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência).

BRASIL. Ministério da Educação. Base Nacional Comum Curricular. Brasília, 2018.

BRASIL. Resolução nº 1, de 4 de outubro de 2022. Normas sobre Computação na Educação Básica - Complemento à BNCC.

BRENNAN, K.; RESNICK, M. New frameworks for studying and assessing the development of computational thinking. In: Proceedings of the 2012 annual meeting of the American Educational Research Association, Vancouver, Canada, p. 1-25, 2012.

BRITO, D. S.; ALMEIDA, L. M. W. O conceito de função em situações de modelagem matemática. Zetetiké, Cempem, Unicamp, v.13, n. 23, p. 63-86, 2005.

BUENO, A. J. A. Uma análise por meio do software Iramuteq de teses e dissertações defendidas entre 2007 e 2017 com a temática filmes comerciais no ensino de ciências. 2018. 94 f. Dissertação (Mestrado em Ensino de Ciências e educação Matemática) – Universidade Estadual de Ponta Grossa, Ponta Grossa, 2018.

CALADO, T. V.; REZENDE, V. A generalização da função afim manifestada por estudantes do 9º ano do Ensino Fundamental. Revista Eletrônica de Educação Matemática - REVEMAT, Florianópolis, v. 17, p. 01-22, jan./dez., 2022.

CAMARGO, E.; NARDI, R. O emprego de linguagens acessíveis para alunos com deficiência visual em aulas de Óptica. Revista Brasileira de Educação Especial, v. 14, n. 3, p. 405-426, 2008.

CAMARGO, B. V.; JUSTO, A. M. IRAMUTEQ: Um software gratuito para análise de dados textuais. Temas em Psicologia. 21 (2): p.513–518, 2013.

CAMARGO, B. V.; JUSTO, A. M. Tutorial para uso do software IRAMUTEQ (Interface de R pour les Analyses Multidimensionnelles de Textes et de Questionnaires). [Florianópolis]: Universidade Federal de Santa Catarina, 74p. 2018.

CAMBRANES, E. Supporting novice programmers with natural language in the early stage of programming. In: Proceedings of IEEE Symposium on Visual Languages and Human-Centric Computing, VL/HCC, p. 173-174, 2013.

CAPPELLE, M. C. A.; MELO, M. C. O. L.; GONÇALVES, C. A. Análise de conteúdo e análise de discurso nas Ciências Sociais. *Organizações Rurais e Agroindustriais – Revista de Administração da UFLA*, v. 5, n. 1, p.1-15, 2003.

CARDOSO, G.; PIRES, A. C.; ABREU, L. V.; ROCHA, F.; GUERREIRO, T. LEGOWorld: repurposing commodity tools & technologies to create an accessible and customizable programming environment. In: *Proceedings of the 2021 CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS*. Yokohama, Japan. Association for Computing Machinery, New York, NY, USA, p. 8 - 13, 2021.

CARNEIRO, V. C. G. Engenharia didática: um referencial para ação investigativa e para formação de professores de Matemática. *Zetetiké*, Campinas, UNICAMP, v. 13, n. 23, p. 85-118, 2005.

CHEVALLARD, Y. Sur l'ingénierie didactique. Texte préparé pour la deuxième Ecole d'Été de Didactique des Mathématiques, Orléans, Juillet, 1982.

CODE.ORG. Instructor Handbook - Code Studio Lesson Plans for Courses One, Two, and Three. CODE.ORG, 2015.

COSTA, R.; ARAÚJO, C; HENRIQUES, P. R. Melodic-teaching computational thinking to visually impaired kids. *In: SECOND INTERNATIONAL COMPUTER PROGRAMMING EDUCATION CONFERENCE (ICPEC 2021)*, v. 91, pages 4:1 - 4:14, 2021.

CRICK, T.; SENTANCE, S.: Computing at school: stimulating computing education in the UK. *In: Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, Koli Calling 2011. ACM, New York, 2011.

DALUD-VINCENT, M. Alceste comme outil de traitement d'entretiens semi-directifs : essai et critiques pour un usage en sociologie. *Langage et Société*, 135(1), p. 9-28, 2011.

D'AMBRÓSIO, B. S. Como ensinar matemática hoje? *SBEM*, Brasília, v. 4, n. 2, p. 15-19, 1989.

DOLZ, J. As atividades e os exercícios de língua: uma reflexão sobre a engenharia didática. *DELTA: Documentação e Estudos em Linguística Teórica e Aplicada*, v. 32, n.1, 2016.

DORNELAS, J. J. B. Análise de uma seqüência didática para a aprendizagem do conceito de função afim. 2007. 182 f. Dissertação (Programa de Pós-Graduação em Ensino das Ciências) - Universidade Federal Rural de Pernambuco, Recife.

DREOSSI, R. C. F.; MOMENSOHN-SANTOS, T. O Ruído e sua interferência sobre estudantes em uma sala de aula: revisão de literatura. *Pró-Fono Revista de Atualização Científica*, Barueri (SP), v. 17, n. 2, p. 251-258, maio-ago. 2005.

DUVAL, R. A cognitive analysis of problems of comprehension in the learning of mathematics. *Educational Studies in Mathematics*, v. 61, p. 103-131, 2006.

DUVAL, R. Registres de représentation sémiotique et fonctionnement cognitif de la pensée. Paris, *Annales de Didactiques et de Sciences Cognitives*, v. 5, p. 37-65, 1993.

DUVAL, R. *Semiosis y pensamiento humano: registros semióticos y aprendizajes intelectuales*. Tradução de Myriam Vega Restrepo. 2ª ed. Cali: Universidad del Valle, 2017.

ESQUINCALHA, A. da C. Nicolas Bourbaki e o movimento Matemática Moderna. *Revista de Educação, Ciências e Matemática* v.2 n.3, set/dez, 2012.

ESTEVES, A. M. S.; NOSCHANG, L. F.; RAABE, A. L. A.; HAENDCHEN FILHO, A. *Portugol Studio: Em direção a uma comunidade aberta para pesquisa sobre o aprendizado de programação*. In: *Workshop sobre Educação em Computação (WEI-SBC)*, Belém-PA. *Anais do XXVII Workshop sobre Educação em Computação*, v. 1. p. 513-522, 2019.

EVARISTO, J.; PINTO, S. C. C. S. *Aprendendo a programar programando numa linguagem algorítmica executável (ILA)*. Rio de Janeiro: Book Express, 2000.

FERNANDES, S. H. A. A.; HEALY, L. *Ensaio sobre a inclusão na Educação Matemática*. Unión. *Revista Iberoamericana de Educación Matemática – UNION, Federación Iberoamericana de Sociedades de Educación Matemática – FISEM*, v. 10, p. 59-76, 2007.

FERNANDES, S. H. A. A.; HEALY, L. *Rumo à Educação Matemática inclusiva: reflexões sobre nossa jornada*. *Revista de Ensino de Ciências e Matemática*, São Paulo, v. 7, n. 4, p. 28–48, 2023.

FERRANDIN, M.; STEPHANI, S. L. *Ferramenta para o ensino de programação via internet*. I Congresso Sul Catarinense de Computação: UNESC, Criciúma, 2012.

FERRI, J; ROSA, S. S. *Como o Ensino de Programação de Computadores Pode Contribuir Com a Construção de Conhecimento na Educação Básica Uma Revisão Sistemática da Literatura*. CINTED-UFRG, 2016.

FIGUEIRAS, L.; ARCAVI, A. *Learning to See: The Viewpoint of the Blind*. *Selected Regular Lectures from the 12th International Congress on Mathematical Education*, Springer, p. 175- 186, 2015.

FIGUEIRAS, L.; HEALY, L.; SKOVSMOSE, O. *Difference, inclusion, and mathematics education: Launching a research agenda*. *International Journal of Studies in Mathematics Education*, 9(3), p. 15-35, 2016.

FRANÇA, T. H. *Modelo social da deficiência: uma ferramenta sociológica para a emancipação social*. *Lutas Sociais*, São Paulo, v. 17 n.31, p.59-73, jul./dez., 2013.

FRANÇA, T. H. P. M. *Deficiência e pobreza no Brasil: a relevância do trabalho das pessoas com deficiência*. 2014. 336 f. Tese (Doutorado em Sociologia - Relações de Trabalho, Desigualdades Sociais e Sindicalismo) – Faculdade de Economia da Universidade de Coimbra, Coimbra, 2014.

GIL, A. C. *Métodos e técnicas de pesquisa social*. 6ª ed. São Paulo: Atlas, 2008.

GROVER, S.; PEA, R. Computational Thinking in K-12: A Review of the State of the Field. *Educational Researcher*, v. 42, n. 1, p. 38-43, 2013.

GUARDA, G. F.; PINTO, S. C. C. S. Dimensões do Pensamento Computacional: conceitos, práticas e novas perspectivas. *In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO (SBIE)*, 31., 2020, Online. Anais [...]. Porto Alegre: Sociedade Brasileira de Computação, 2020. p. 1463-1472.

GUEDES, A. M. S.; SOUZA, T. S. A. de; AZEVEDO, I. F. de; NORONHA, W. F. R.; ALVES, F. R. V. A Engenharia Didática como ferramenta para a concepção de um objeto de aprendizagem aplicado ao ensino de probabilidade. *Research, Society and Development*, v. 8, n. 11, p. 1-18, 2019.

GUIMARÃES, A. M.; LAGES, N. A. de C. Algoritmos e Estrutura de dados. Rio de Janeiro: LTC, 1985.

HADWEN-BENNETT, A.; SENTANCE, S.; MORRISON, C. Making Programming Accessible to Learners with Visual Impairments: A Literature Review. *International Journal of Computer Science Education in Schools*, v. 2, n. 2, p. 3-13, 2018.

HEALY, L.; FERNANDES, S. H. A. A. Relações entre atividades sensoriais e artefatos culturais na apropriação de práticas matemáticas de um aprendiz cego. *Educar em Revista (Impresso)*, v. esp., p. 227-244, 2011.

HEALY, L.; POWELL, A. B. Understanding and Overcoming “Disadvantage” in Learning Mathematics. *In: CLEMENTS, M. A. (Ken) et al. (Eds.) Third International Handbook of Mathematics Education*. New York: Springer International Handbooks of Education, v.27, p. 69- 100, 2013.

HORTA, R. F.; FERREIRA, M. A influência da autoestima no desempenho escolar. *Revista Ensin@ UFMS*, v. 2, n. esp., p. 276-286, 15 dez. 2021.

HOSTINS, H.; RAABE, A. Auxiliando a aprendizagem de algoritmos com a ferramenta WebPortugol. *In: Workshop sobre Educação em Computação*, 15., 2007, Rio de Janeiro. Anais do XXVII Congresso da Sociedade Brasileira de Computação, Porto Alegre: Sociedade Brasileira de Computação, p. 96-105, 2007.

INDIA, G.; RAMAKRISHNA, G.; PAL, J.; SWAMINATHAN, M. 2020. Conceptual learning through accessible play: Project Torino and computational thinking for blind children in India. *In: Proceedings of the 2020 INTERNATIONAL CONFERENCE ON INFORMATION AND COMMUNICATION TECHNOLOGIES AND DEVELOPMENT*. Guayaquil, Equador, p. 1 - 11, 2020.

INDIA, G.; RAMAKRISHNA, G.; BISHT, J.; SWAMINATHAN, M. Computational thinking as play: experiences of children who are blind or low vision in india. *In: Proceedings of the 21st ASSETS '19*. Pittsburgh, USA, p. 519–522, 2019.

ISTE/CSTA. Computational Thinking Teacher Resource. 2 ed., 2011. Disponível em: https://cdn.iste.org/www-root/2020-10/ISTE_CT_Teacher_Resources_2ed.pdf. Acesso em maio de 2024.

HORN, M. TopCode: Tangible Object Placement Codes. 2012. Disponível em: <http://users.eecs.northwestern.edu/~mhorn/topcodes/>. Acesso em agosto de 2023.

HU, C. Computational thinkinSara: what it might mean and what we might do about it. *In: ITiCSE '11. Proceedings of ITiCSE 2011. New York: ACM, p. 223-227, 2011.*

JESUS, E. Teaching computer programming with structured programming language and flowcharts. *In: Workshop on Open Source and Design of Communication, 2011, Lisboa. Proceedings of the 2011 Workshop on Open Source and Design of Communication, New York: Association for Computing Machinery, p. 45-48, 2011.*

KANE, S. K.; KOUSHIK, V.; MUEHLBRADT, A. Bonk: Accessible programming for accessible audio games. *In: PROCEEDINGS OF THE CONFERENCE ON INTERACTION DESIGN AND CHILDREN, Trondheim, Norway. Association for Computing Machinery: New York, NY, USA, p. 132-142, 2018.*

KÖLLING, M.; BROWN, N. C. C.; ALTADMRI, A. Frame-Based Editing. *Journal of Visual Languages and Sentient Systems, v. 3, p. 40-67, 2017.*

KÖRBER, N. et al. An experience of introducing primary school children to programming using ozobots. *In: PROCEEDINGS OF THE 16TH WORKSHOP IN PRIMARY AND SECONDARY COMPUTING EDUCATION (WiPSCE '21). Association for Computing Machinery, New York, NY, USA, p. 1 - 6, 2021.*

KOUSHIK, V.; GUINNESS, D.; KANE, S. K. StoryBlocks: a tangible programming game to create accessible audio stories. *In: Proceedings of the 2019 CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS (CHI '19). Glasgow, Scotland, UK. Association for Computing Machinery, New York, NY, USA, p. 1 - 12, 2019.*

KLINGENBERG, O. G.; HOLKESVIK, A. H. Digital learning in mathematics for students with severe visual impairment: A systematic review. *British Journal of Visual Impairment. Vol. 38(1), p. 38-57, 2020.*

LEWIS, A. L. M.; BODNER, G. M. Chemical reactions: what understanding do students with blindness develop? *Chemistry Education Research and Practice, v.14, n.4, p. 625, 2013.*

LIMA, E. L. O Ensino Médio da Matemática. *Revista Gazeta de Matemática, Rio de Janeiro 2003, n° 144, p.44-45, Janeiro 2003.*

LIMA, E. L.; CARVALHO, P. C. P.; WAGNER, E.; MORGADO, A. C. A Matemática do Ensino Médio. Vol 1. 9ª ed. Rio de Janeiro: Sociedade Brasileira de Matemática, 2006.

LIRA, M. C. F.; SCHLINDWEIN, L.M. A pessoa cega e a inclusão: um olhar a partir da psicologia histórico-cultural. *Cadernos Cedes, Campinas, v. 28, n. 75, p. 171-190, 2008.*

LIUKAS, L. Hello Ruby: adventures in coding. Feiwei & Friends, 2015.

LODI, M.; MARTINI, S. Computational Thinking, Between Papert and Wing. *Sci & Educ 30, p. 883-908, 2021.*

LUDI, S.; BERNSTEIN, D.; MUTCH-JONES, K. Enhanced robotics! Improving building and programming learning experiences for students with visual impairments. *In: Proceedings of the 49TH ACM TECHNICAL SYMPOSIUM ON COMPUTER SCIENCE EDUCATION*. Baltimore, MD, USA, p. 372 - 377, 2018.

LORENCINI, P. B. M.; NOGUEIRA, C. M. I.; REZENDE, V. Formas operatória e predicativa de conhecimentos sobre função afim identificadas a partir do diálogo entre alunos videntes e com deficiência visual. *RBECM*, v. 3, n. 2, p. 625-653, jul./dez., 2020.

MACÊDO, J. A. de; BRANDÃO, D. P.; NUNES, D. M. Limites e possibilidades do uso do livro didático de Matemática nos processos de ensino e de aprendizagem. *Educação Matemática Debate*, v. 3, n. 7, p. 68-86, 2019.

MANSO, A.; OLIVEIRA, L.; MARQUES, C. G. Ambiente de aprendizagem de algoritmos – Portugal IDE. *In: Conferência Internacional de TIC na Educação*, 6., 2009, Braga. *Actas da VI Conferência Internacional de TIC na Educação – Challenges 2009*, Braga: Centro de Competência da Universidade do Minho, p. 969-983, 2009.

MANTOAN, M. T. E. A educação especial no Brasil: da exclusão à inclusão escolar. Universidade Estadual de Campinas. Faculdade de Educação. Laboratório de Estudos e Pesquisas em Ensino e Diversidade LEPED/UNICAMP, 2002.

MANZANO, J. A. N. G. *ILA+: Programação de computadores em português*. São Paulo: Propes Vivens, 2017.

MARCELLY, L. Do improviso às possibilidades de ensino: estudo de caso de uma professora de matemática no contexto da inclusão de estudantes cegos. 2015. 192 f. Tese - (doutorado) - Universidade Estadual Paulista, Instituto de Geociências e Ciências Exatas, 2015.

MARCONI, M. A.; LAKATOS, E. M. *Fundamentos de metodologia científica*. 5ª ed. São Paulo: Atlas, 2003.

MEDEIROS, I. L.; VIEIRA, A.; BRAVIANO, G.; GONÇALVES, B. S. Revisão sistemática e bibliometria facilitadas por um Canvas para visualização de informação. *Revista Brasileira de Design da Informação*, v. 12, n. 1, p. 93-110, 2015.

MEDEIROS, A. V. M. Um interpretador online para a linguagem Portugal. 2015. 117 p. Trabalho de Conclusão de Curso (graduação) – Curso de Ciência da Computação, Universidade Federal de Sergipe, São Cristóvão, 2015.

METATLA, O.; BARDOT, S.; CULLEN, C.; SERRANO, M.; JOUFFRAIS, C. Robots for Inclusive Play: co-designing an educational game with visually impaired and sighted children. *In: Proceedings of the 2020 CHI CONFERENCE ON HUMAN FACTORS IN COMPUTING SYSTEMS*, p. 1 - 13, 2020.

MINDSTORMS. Lego, 2023. Disponível em: <https://www.lego.com/pt-br/themes/mindstorms/about>. Acesso em: 6 de setembro de 2023.

MIRANDA, E. M.; FERNANDES, A. M. R.; DAZZI, R. L. S.; SANTIAGO, R. CIFluxproII: ferramenta para auxiliar a avaliação de algoritmos utilizando processamento de linguagem natural. *In: XI Congreso Argentino de Ciencias de la Computación, 2005, Concordia, Entre Ríos. Universidad Nacional de Entre Ríos, 2005.*

MOUNTAPMBEME, A.; LUDI, S. Investigating challenges faced by learners with visual impairments using block-based programming/hybrid environments. *In: Proceedings of the 22ND INTERNATIONAL ACM SIGACCESS CONFERENCE ON COMPUTERS AND ACCESSIBILITY (ASSETS '20). Greece, p. 26 - 28, 2020.*

NASCIMENTO, C. A.; SANTOS, D. A.; TANZI, A. Pensamento computacional e interdisciplinaridade na Educação Básica: um mapeamento sistemático. *In: CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO, 7., 2018, Fortaleza. Anais... Fortaleza: Sociedade Brasileira de Computação, 2018.*

NASCIMENTO, F. P. Classificação da Pesquisa: Natureza, método ou abordagem metodológica, objetivos e procedimentos. *In: NASCIMENTO, F. P.; SOUSA, F. L. L. Metodologia da Pesquisa Científica: teoria e prática – como elaborar TCC. Brasília: Thesaurus, 2016.*

NOSCHANG, L. F. et al. Portugal Studio: Uma IDE para iniciantes em programação. *In: Workshop sobre Educação em Computação, 22., 2014, Brasília. Anais do XXXIV Congresso da Sociedade Brasileira de Computação, Porto Alegre: Sociedade Brasileira de Computação, p. 1287-1296, 2014.*

NUNES, S. S.; LOMÔNACO, J. F. B. Desenvolvimento de conceitos em cegos congênitos: caminhos de aquisição do conhecimento. *Revista Semestral da Associação Brasileira de Psicologia Escolar e Educacional, Campinas, v. 12, n. 1, p. 119-138, 2008.*

OLIVEIRA, O. L.; MONTEIRO, A. M.; ROMAN, N. T. Can natural language be utilized in the learning of programming fundamentals? *In: Proceedings of FRONTIERS IN EDUCATION CONFERENCE, FIE, p. 1851 - 1856, 2013.*

OLIVEIRA, J.D.; CAMPOS, M.D.; BORDINI, R.H.; AMORY, A. GoDonnie: A robot programming language to improve orientation and mobility skills in people who are visually impaired. *In: Proceedings of the 21ST INTERNATIONAL ACM SIGACCESS CONFERENCE ON COMPUTERS AND ACCESSIBILITY. Pittsburgh, PA, USA, p. 28 - 30, 2019.*

ORMELEZI, E. M. Os caminhos da aquisição do conhecimento e a cegueira: do universo do corpo ao universo simbólico. 2000. Dissertação (Mestrado em História e Filosofia da Educação) - Faculdade de Educação, Universidade de São Paulo, São Paulo, 2000.

PAPADAKIS, S. Robots and robotics kits for early childhood and first school age. *International Journal of Interactive Mobile Technologies, v. 14, n. 18, p. 34-56, 2020.*

PAPERT, S.; SOLOMON, C. Twenty Things to do With a Computer, *Educational Technology, v. 12, p. 9-18, april, 1972.*

PAPERT, S. *Mindstorms: children, computers, and powerful ideas.* Basic Books, 1980.

PASTERNAK, E.; FENICHEL, R.; MARSHALL, A. N. Tips for creating a block language with blockly. *In: 2017 IEEE Blocks and Beyond Workshop (B&B), Raleigh, NC, USA, p. 21-24, 2017.*

PINTO, M. S. M.; OSORIO, A. J.; MONTEIRO, A. F. Potencialidades e fragilidades de robôs para crianças em idade pré-escolar: 3 a 6 anos. *Revista Observatório, p. 302-330, 2017.*

PIRES, A. C. et al. Learning maths with a tangible user interface: lessons learned through participatory design with children with visual impairments and their educators. *International Journal of Child-Computer Interaction, 32, 100382, p.1-15, 2022.*

PIRES, A. C. et al. Exploring accessible programming with educators and visually impaired children. *In: Proceedings of the INTERACTION DESIGN AND CHILDREN CONFERENCE, Association for Computing Machinery, p. 148-160, 2020.*

PIRES, A. C. et al. A Tangible Math Game for Visually Impaired Children. *In: Proceedings of the 21th INTERNATIONAL CONFERENCE ON COMPUTERS AND ACCESSIBILITY (ASSETS'19). Pittsburgh, USA, 2019.*

PONTE, J. P. O conceito de função no currículo de Matemática. *Revista Educação e Matemática, APM, Portugal, n.15, p. 3-9, 1990.*

PUZZLETS. The education partnership, 2023a. Disponível em: <https://theeducationpartnership.org/steam-lending-library/puzzlets/>. Acesso em 31 de agosto de 2023.

PUZZLETS. Educational Technology Consultants, Inc., 2023b. Disponível em: <https://www.edtech-consultants.com/puzzlets>. Acesso em 31 de agosto de 2023.

REINERT, M. ALCESTE, une méthodologie d'analyse des données textuelles et une application: Aurélia de Gerard de Nerval. *Bulletin de Méthodologie Sociologique, (28), p.24-54, 1990*

REZENDE, S. M., PINTO, S. C. C. S. O uso de tecnologias digitais em objetos de aprendizagem da matemática para estudantes com deficiência visual: um levantamento de teses e dissertações brasileiras. *Caminhos da educação matemática em revista (online)/IFS, v. 11, n. 3, 2021.*

REZENDE, V., NOGUEIRA, C. M. I.; CALADO, T. V. Função afim na Educação Básica: estratégias e ideias base mobilizadas por estudantes mediante a resolução de tarefas matemáticas. *Alexandria: Revista de Educação em Ciência e Tecnologia, Florianópolis, v.13, n.2, p. 25-50, 2020.*

RICHARDSON, R. J. *Pesquisa social: métodos e técnicas. 3ª ed. São Paulo: Atlas, 1999.*

ROCHA, F. et al. Fostering collaboration with asymmetric roles in accessible programming environments for children with mixed-visual-abilities. *In: Proceedings of the 23rd INTERNATIONAL ACM SIGACCESS CONFERENCE ON COMPUTERS AND ACCESSIBILITY. Association for Computing Machinery, New York, USA, 2021.*

ROCHA, F.; PIRES, A. C.; NETO, I.; NICOLAU, H.; GUERREIRO, T. Assembly at home: accessible spatial programming for children with visual impairments and their families. *In: Proceedings of the IDC '21: INTERACTION DESIGN AND CHILDREN*. New York, NY, USA, 2021.

RODRIGUEZ, C. L.; REIS, R. C. D.; ISOTANI, S. Recursos e estratégias para desenvolvimento e avaliação do Pensamento Computacional na escola. *Tecnologias, Sociedade e Conhecimento*. vol. 4, n. 1, dez. 2017.

RONG, Z.; CHAN, N. F.; CHEN, T.; ZHU, K. CodeRhythm: a tangible programming toolkit for visually impaired students. *In: Proceedings of the EIGHTH INTERNATIONAL WORKSHOP OF CHINESE CHI (CHINESE CHI 2020)*. Honolulu, HI, USA. Association for Computing Machinery, New York, NY, USA, p. 57-60, 2020a.

ROYAL SOCIETY. Shut down or restart: the way forward for computing in UK schools. The Royal Society, 2012.

SÁ, E. D.; SILVA, M. B. C.; SIMÃO, V. S. Atendimento educacional especializado do aluno com deficiência visual. São Paulo: Moderna, 2010.

SABUNCUOGLU, A. Tangible music programming blocks for visually impaired children. *In: Proceedings of the FOURTEENTH INTERNATIONAL CONFERENCE ON TANGIBLE, EMBEDDED, AND EMBODIED INTERACTION, TEI '20*. Association for Computing Machinery New York, NY, USA, p. 423-429, 2020.

SALVADOR, P. T. C. O. et al. Uso do software IRAMUTEQ nas pesquisas brasileiras da área da saúde: uma scoping review. *Rev Bras Promoç Saúde*, 31, p.1-9, 2018.

SALVIATI, M. E. Manual do Aplicativo IRAMUTEQ (versão 0.7 Alpha 2 e R Versão 3.2.3). Compilação, organização e notas de Maria Elisabeth Salviati, 2017. Disponível em: <http://www.iramuteq.org/documentation/fichiers/manual-do-aplicativo-iramuteq-par-maria-elisabeth-salviati>. Acesso em: 17 abr. 2024.

SAMPAIO, R. F.; MANCINI, M. C. Estudos de revisão sistemática: um guia para síntese criteriosa da evidência científica. *Revista Brasileira de Fisioterapia*, São Carlos, v. 11, n. 1, jan./fev., p. 83-89, 2007.

SANTIAGO, R.; DAZZI, R. L. S. Interpretador de Portugol. *In: Congresso Brasileiro de Computação*, 4., 2004, Itajaí. *Anais do IV Congresso Brasileiro de Computação*, Itajaí: UNIVALI, p. 26-29, 2004.

SANTIN, S.; SIMMONS, J. Problemas das crianças portadoras de deficiência visual congênita na construção da realidade. Tradução de Ilza Viega. *Revista Benjamin Constant*, 2ª ed. p.7-11, 1996.

SEO, J.; RICHARD, G. Accessibility, making and tactile robotics: facilitating collaborative learning and computational thinking for learners with visual impairments. *In: Kay, J. and Luckin, R. (Eds.) Rethinking Learning in the Digital Age: Making the Learning Sciences Count*. Presented at ICLS '18: THE 13TH INTERNATIONAL CONFERENCE OF THE LEARNING SCIENCES. London, UK: International Society of the Learning Sciences, 2018.

SILVA, T. G-Portugol: manual da versão v1.0. 2006. Disponível em: <https://lapolli.pro.br/escolas/unicid/tecProg/laboratorio/portugol/portugol.pdf>. Acesso em 21 de maio de 2024.

SILVA, C. R. G. X. da. O conceito de função do contexto universitário ao contexto escolar. 2021. 66 f. Dissertação (Mestrado em Matemática) - Universidade Estadual do Oeste do Paraná, Cascavel - PR.

SILVA, M. H. M.; REZENDE, W. M. Análise histórica do conceito de função. Caderno Da Licença. Instituto de Matemática. Universidade Federal Fluminense. v.2. p. 28-33. Niterói, 1999.

SILVA, S. D.; NOGUEIRA, C. M. I. O processo de generalização de função afim na perspectiva de Jean Piaget. Schème: Revista Eletrônica de Psicologia e Epistemologia Genéticas, v.14, n.1, p.102-144, 2022.

SKOVSMOSE, O. Inclusions, meetings and landscapes. In: KOLLOSCH, D.; MARCONE, R.; KNIGGE, M.; PENTEADO, M. G.; SKOVSMOSE, O. (Eds) Inclusive mathematics education: state-of-the-art research from Brazil and Germany, p. 71-84. Springer, 2019.

SOARES NETO, C. S.; GOMES JÚNIOR, D. L.; MARQUES, L. S.; SOUSA, U. S. Compilador UFMA-CP: uma ferramenta de apoio para o ensino básico de lógica de programação. In: Jornada de Informática do Maranhão, 2008, São Luís. Anais da II Jornada de Informática do Maranhão, São Luís: Universidade Federal do Maranhão, 2008.

SOUSA, Y. S. O. O Uso do Software Iramuteq : Fundamentos de Lexicometria para Pesquisas Qualitativas. Estudos e Pesquisas em Psicologia, 21(4), p.1541-1560, 2021.

SOUSA, U. S. et al. Uso de português estruturado no ensino de alunos de computação com geração automatizada de códigos em linguagem Python. Brazilian Journal of Development, 8(5), p. 38111-38121, 2022.

SOUSA, J. R. de; SANTOS , S. C. M. dos. Análise de conteúdo em pesquisa qualitativa: modo de pensar e de fazer. Pesquisa e Debate em Educação, [S. l.], v. 10, n. 2, p. 1396-1416, 2020.

SOUZA, C. M. de. VisuAlg - Ferramenta de Apoio ao Ensino de Programação. Revista TECCEN, v. 2, n. 2, p.1-9, 2009.

SOUZA, M. S; TAMANINI, P. A; SANTOS, J. M. C. T. Cultura digital: tecnologias, escola e novas práticas educativas. Revista Pedagógica, Chapecó, v. 22, p. 1-19, 2020.

SOUZA, D. M.; BATISTA, M. H. da S.; BARBOSA, E. F. Problemas e Dificuldades no Ensino de Programação: um mapeamento sistemático. Revista Brasileira de Informática na Educação, [S.L.], v. 24, n. 1, p. 39, 2016.

SPINAROVA, G.; VACHALOVA, V. Digital technology as support of cognitive processes of people with visual impairments. In: SOCIETY. INTEGRATION. EDUCATION.

Proceedings of the INTERNATIONAL SCIENTIFIC CONFERENCE, v. 3, p. 195-204, 2021.

SUPALO, C. A.; ISACCON, M. D.; LOMBARDI, M. V. Making hands-on science learning accessible for students who are blind or have low vision. *Journal of Chemical Education*, 91(2), p. 195-199, 2014.

SUPER DOC. Clementoni, 2023. Disponível em: <https://pt.clementoni.com/products/super-doc>. Acesso em 31 de agosto de 2023.

TALL, D; VINNER, S. Concept image and concept definition with particular reference to limits and continuity. *In: Educational Studies in Mathematics*, n. 12, p. 151 – 169, 1981.

THIEME, A.; MORRISON, C.; VILLAR, N.; GRAYSON, M.; LINDLEY, S. Enabling collaboration in learning computer programming inclusive of children with vision impairments. *In: Proceedings of the 2017 CONFERENCE ON DESIGNING INTERACTIVE SYSTEMS*, p. 739-752, 2017.

THOMPSON, P. W. Imagery and the development of mathematical reasoning. *In: STEFFE, L. P.; NESHER, P.; COBB, P.; GOLDIN, G. A.; GREER, B. (eds.), Theories of mathematical learning*, p. 267-283. Hillsdale, NJ: Erlbaum, 1996.

UNESCO. The Salamanca Statement and framework for action on special needs education. *World Conference on Special Needs Education: Access and Quality*. Salamanca: UNESCO, 1994.

VALENTE, J. A. Integração do pensamento computacional no currículo da educação básica: diferentes estratégias usadas e questões de formação de professores e avaliação do aluno. *e- Curriculum*, São Paulo, v. 14, n. 6, p. 34, 2016.

VINNER, S. The role of definitions in teaching and learning. *In: Advanced Mathematical Thinking* (Ed. David Tall). Kluwer publications, 1991.

WERLICH, C.; KEMCZINSKI, A.; GASPARINI, I. Pensamento Computacional no Ensino Fundamental: um mapeamento sistemático. *Nuevas Ideas en Informática Educativa*, vol. 14, p. 375-384, Santiago, 2018.

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

WING, J. M. Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 366(1881), p. 3717-3725, 2008.

WING, J. M. Computational Thinking: What and Why? Unpublished Manuscript, Pittsburgh, PA: Computer Science Department, Carnegie Mellon University, 2010.

WING, J. M. Computational thinking benefits society. 40th Anniversary Blog of Social Issues in Computing, 2014.

ZBOROWSKI, C. A; PIGATTO, A. G. S. Contribuições da engenharia didática como metodologia para o ensino de ciências nos anos iniciais. VIDYA, v. 38, n. 2, p. 71-88, Santa Maria, 2018.

APÊNDICES E ANEXOS

APÊNDICE A - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO (TCLE)

Dados de identificação

Título do Projeto: A programação de computadores como ferramenta para resolução de problemas matemáticos e desenvolvimento do pensamento computacional junto a estudantes com deficiência visual

Pesquisador Responsável: Sandro Miranda de Rezende

Instituição a que pertence o pesquisador responsável: Universidade Federal Fluminense

Telefone para contato do pesquisador: (21) 992450710

E-mail do pesquisador: sdromiranda@gmail.com

Nome do Participante: _____

Responsável legal: _____

O(A) sr.(^a) está sendo convidado(a) a participar do projeto de pesquisa “*A programação de computadores como ferramenta para resolução de problemas matemáticos e desenvolvimento do pensamento computacional junto a estudantes com deficiência visual*”, de responsabilidade do pesquisador Sandro Miranda de Rezende, que tem por objetivo analisar as potencialidades da programação de computadores como recurso para a resolução de problemas matemáticos e desenvolvimento do pensamento computacional junto a estudantes do Ensino Médio com deficiência visual. O estudo é desenvolvido sob orientação do professor Dr. Sergio Crespo Coelho da Silva Pinto, vinculado à Universidade Federal Fluminense, que pode ser contatado através do e-mail screspo@id.uff.br.

Leia cuidadosamente o que se segue e não hesite em perguntar sobre qualquer dúvida que você tiver. Caso se sinta esclarecido(a) sobre as informações que estão neste termo e autorize que o menor pelo qual o(a) sr.(^a) é responsável faça parte do estudo, peço que assine ao final deste documento em duas vias. Ressalta-se que é assegurada uma via deste termo de consentimento ao(à) sr.(^a) responsável, e ainda, que são direitos dos participantes, conforme determinado na Resolução CNS nº 466/12 e na Resolução CNS nº 510/16: ser informado sobre a pesquisa; desistir a qualquer momento de participar da pesquisa, sem qualquer prejuízo; ter sua privacidade respeitada; ter garantida a confidencialidade das informações pessoais; decidir se sua identidade será divulgada e quais são, dentre as informações que forneceu, as que podem ser tratadas de forma pública; ser indenizado pelo dano decorrente da pesquisa, nos termos da Lei; e ser ressarcido por despesas diretamente decorrentes de sua participação na pesquisa. Esclarecemos que a participação do menor no estudo é voluntária e, portanto, o(a) senhor(a) não é obrigado(a) a colaborar com as atividades solicitadas pelo pesquisador. Caso decida não autorizar a participação do menor do estudo, ou resolver a qualquer momento desistir da participação, informamos que este consentimento pode ser retirado a qualquer tempo, sem prejuízos de qualquer natureza ou qualquer tipo de penalização ao(à) participante. Nem você e nem o(a) participante pagarão nada para participar desta pesquisa, e também não receberão nenhum pagamento para a participação, pois é voluntária. Se houver necessidade, as despesas (deslocamento e alimentação) para a participação do(a) menor serão assumidas ou ressarcidas pelos pesquisadores. Fica também garantida indenização em casos de danos, comprovadamente decorrentes da participação do(a) menor na pesquisa, conforme decisão judicial ou extrajudicial. Os pesquisadores estarão à sua disposição para qualquer esclarecimento em qualquer etapa da pesquisa, podendo ser contatados sempre que necessário através do telefone e e-mails

fornecidos.

A participação nesta pesquisa se dividirá em duas etapas. Na primeira, o(a) estudante participará de situações de aprendizagem conduzidas pelo pesquisador, com vistas ao desenvolvimento de conceitos básicos de programação de computadores, bem como de habilidades do pensamento computacional, utilizando para tal uma linguagem de programação acessível. Tais situações de aprendizagem serão conduzidas no Núcleo de Atendimento a Pessoas com Necessidades Específicas (Napne) do campus do Colégio Pedro II no qual o(a) estudante está matriculado(a). Na segunda etapa, o(a) estudante será convidado(a) a construir programas de computador a partir dos conhecimentos desenvolvidos, com o objetivo de resolver um conjunto de problemas matemáticos propostos. Posteriormente, será realizada uma entrevista com os(as) participantes acerca dos processos de construção dos programas e de resolução dos problemas propostos. As atividades conduzidas juntos aos estudantes serão semanais e terão duração de uma hora e meia cada, sendo realizadas ao longo de 10 semanas. Os dois últimos encontros serão destinados à proposta de construção de programas para resolução de problemas matemáticos. Já a entrevista realizada após a condução das atividades semanais terá duração aproximada de trinta minutos. Todos os encontros para desenvolvimento das atividades serão gravados em vídeo, e as entrevistas posteriormente realizadas serão gravadas em áudio.

Toda pesquisa com seres humanos envolve riscos aos participantes. Nesta pesquisa os riscos envolvidos são: cansaço, indisposição, receio ou aborrecimento ao participar das sequências didáticas e/ou entrevistas; vergonha, constrangimento, desconforto ou alterações na autoestima por apresentar dificuldades na realização das atividades propostas; quebra de sigilo e divulgação de dados confidenciais. Com o objetivo de minimizar os riscos mencionados, o pesquisador tomará as seguintes providências: será comunicado inicialmente aos participantes que a pesquisa não tem qualquer relação com as atividades acadêmicas escolares, não fazendo parte da composição de notas de nenhuma disciplina; será comunicado aos participantes que não há qualquer problema se houver dificuldade na realização das atividades propostas, sendo tal fato natural no processo de aprendizagem; o pesquisador interromperá a condução das atividades caso o participante demonstre ansiedade, nervosismo ou cansaço durante a sua realização; as gravações de vídeos e áudios serão marcadas com um código de identificação durante a gravação e vistas e/ou ouvidas unicamente pelo pesquisador responsável e seu orientador, não sendo utilizado o nome do(a) participante. Todos os áudios e vídeos, assim como qualquer documento que possa associar o nome do(a) participante ao código de identificação serão armazenados em computador pessoal do pesquisador. Todo o material de pesquisa será mantido em arquivo digital, sob sua guarda e responsabilidade, por um período de 5 anos após o término da pesquisa, conforme determinado na Resolução nº 466/2012. Por ocasião da publicação dos resultados da pesquisa, o nome do(a) participante será mantido em sigilo absoluto, sendo utilizados os códigos marcados nas gravações para identificação dos(as) participantes. As gravações serão utilizadas somente para coleta de dados, sendo garantida a manutenção do sigilo e da privacidade da participação do(a) menor pelo(a) qual o(a) sr.(ª) é responsável, bem como de todos os seus dados, durante todas as fases da pesquisa e posterior divulgação científica.

Os benefícios esperados da pesquisa para a comunidade são: identificação de estratégias para o desenvolvimento de habilidades básicas de programação de computadores junto a estudantes com deficiência visual; identificação de vantagens e desvantagens na utilização da programação de computadores como recurso para resolução de problemas matemáticos por estudantes com deficiência visual; identificação de vantagens e desvantagens específicas da linguagem de programação utilizada; identificação de caminhos para o desenvolvimento de habilidades do pensamento computacional junto a estudantes com deficiência visual. Para o participante, os benefícios são: o desenvolvimento de habilidades introdutórias de programação de computadores e o desenvolvimento de habilidades para construção de algoritmos computacionais para resolução de problemas matemáticos.

Em caso de dúvidas relacionadas aos aspectos éticos deste estudo, você poderá consultar o Comitê de Ética em Pesquisa em Ciências Sociais, Sociais Aplicadas, Humanas, Letras, Artes e Linguística (CEP Humanas UFF) no endereço: (Rua Passo da Pátria, 156 - São Domingos - Niterói, Rio de Janeiro. Campus da Praia Vermelha da UFF - Instituto de Física (torre nova - 3º andar).

Telefone: (21) 2629-5119
e-mail: cephumanasuff@gmail.com

(assinatura do pesquisador)

CONSENTIMENTO DO(DA) RESPONSÁVEL PELO(A) MENOR DE IDADE

Eu, _____, responsável legal por _____, declaro ter compreendido as informações apresentadas neste termo de consentimento, e concordo com a participação do(a) menor no projeto de pesquisa acima descrito.

Niterói, ____ de _____ de _____

(nome e assinatura do participante ou responsável legal)

APÊNDICE B - TERMO DE ASSENTIMENTO LIVRE E ESCLARECIDO (TALE) *

Dados de identificação

Título do Projeto: A programação de computadores como ferramenta para resolução de problemas matemáticos e desenvolvimento do pensamento computacional junto a estudantes com deficiência visual

Pesquisador Responsável: Sandro Miranda de Rezende

Instituição a que pertence o pesquisador responsável: Universidade Federal Fluminense

Telefone para contato do pesquisador: (21) 992450710

E-mail do pesquisador: sdromiranda@gmail.com

Nome do Participante: _____

Responsável legal: _____

Você está sendo convidado(a) a participar do projeto de pesquisa “*A programação de computadores como ferramenta para resolução de problemas matemáticos e desenvolvimento do pensamento computacional junto a estudantes com deficiência visual*”, de responsabilidade do pesquisador Sandro Miranda de Rezende, que tem por objetivo analisar as potencialidades da programação de computadores como recurso para a resolução de problemas matemáticos e desenvolvimento do pensamento computacional junto a estudantes do Ensino Médio com deficiência visual. O estudo é desenvolvido sob orientação do professor Dr. Sergio Crespo Coelho da Silva Pinto, vinculado à Universidade Federal Fluminense, que pode ser contatado através do e-mail screspo@id.uff.br.

Leia cuidadosamente o que se segue e não hesite em perguntar sobre qualquer dúvida que você tiver. Caso se sinta esclarecido(a) sobre as informações que estão neste termo, peço que assine ao final deste documento em duas vias. Ressalta-se que é assegurada uma via deste termo de assentimento a você, e ainda, que são direitos dos participantes, conforme determinado na Resolução CNS nº 466/12 e na Resolução CNS nº 510/16: ser informado sobre a pesquisa; desistir a qualquer momento de participar da pesquisa, sem qualquer prejuízo; ter sua privacidade respeitada; ter garantida a confidencialidade das informações pessoais; decidir se sua identidade será divulgada e quais são, dentre as informações que forneceu, as que podem ser tratadas de forma pública; ser indenizado pelo dano decorrente da pesquisa, nos termos da Lei; e ser ressarcido por despesas diretamente decorrentes de sua participação na pesquisa. Esclarecemos, dessa forma, que sua participação no estudo é voluntária e, portanto, você não é obrigado(a) a colaborar com as atividades solicitadas pelo pesquisador. Caso decida não participar do estudo, ou resolver a qualquer momento desistir da participação, informamos que este consentimento pode ser retirado a qualquer tempo, sem prejuízos de qualquer natureza ou qualquer tipo de penalização ao(à) participante. Nem você e nem seus pais [ou responsáveis legais] pagarão nada para você participar desta pesquisa, e também não receberão nenhum pagamento para a sua participação, pois é voluntária. Se houver necessidade, as despesas (deslocamento e alimentação) para a sua participação e de seus pais serão assumidas ou ressarcidas pelos pesquisadores. Fica também garantida indenização em casos de danos, comprovadamente decorrentes da sua participação na pesquisa, conforme decisão judicial ou extra-judicial. Os pesquisadores estarão à sua disposição para qualquer esclarecimento em qualquer etapa da pesquisa, podendo ser contatados sempre que necessário através do telefone e e-mails fornecidos.

A participação nesta pesquisa se dividirá em duas etapas. Na primeira, você participará de situações de aprendizagem conduzidas pelo pesquisador, com vistas ao desenvolvimento de conceitos básicos de programação de computadores, bem como de habilidades do pensamento computacional, utilizando para tal uma linguagem de programação acessível. Tais situações de

aprendizagem serão conduzidas no Núcleo de Atendimento a Pessoas com Necessidades Específicas (Napne) do campus do Colégio Pedro II no qual você está matriculado(a). Na segunda etapa, você será convidado(a) a construir programas de computadora partir dos conhecimentos desenvolvidos, com o objetivo de resolver um conjunto de problemas matemáticos propostos. Posteriormente, você participará de uma entrevista acerca dos processos de construção dos programas e de resolução dos problemas propostos. As atividades conduzidas juntos aos estudantes serão semanais e terão duração de uma hora e meia cada, sendo realizadas ao longo de 10 semanas. Os dois últimos encontros serão destinados à proposta de construção de programas para resolução de problemas matemáticos. Já a entrevista realizada após a condução das atividades semanais terá duração aproximada de trinta minutos. Todos os encontros para desenvolvimento das atividades serão gravados em vídeo, e as entrevistas posteriormente realizadas serão gravadas em áudio.

Toda pesquisa com seres humanos envolve riscos aos participantes. Nesta pesquisa os riscos envolvidos são: cansaço, indisposição, receio ou aborrecimento ao participar das sequências didáticas e/ou entrevistas; vergonha, constrangimento, desconforto ou alterações na autoestima por apresentar dificuldades na realização das atividades propostas; quebra de sigilo e divulgação de dados confidenciais. Com o objetivo de minimizar os riscos mencionados, o pesquisador tomará as seguintes providências: será comunicado inicialmente aos participantes que a pesquisa não tem qualquer relação com as atividades acadêmicas escolares, não fazendo parte da composição de notas de nenhuma disciplina; será comunicado aos participantes que não há qualquer problema se houver dificuldade na realização das atividades propostas, sendo tal fato natural no processo de aprendizagem; o pesquisador interromperá a condução das atividades caso o participante demonstre ansiedade, nervosismo ou cansaço durante a sua realização; as gravações de vídeos e áudios serão marcadas com um código de identificação durante a gravação e vistas e/ou ouvidas unicamente pelo pesquisador responsável e seu orientador, não sendo utilizado o nome do(a) participante. Todos os áudios e vídeos, assim como qualquer documento que possa associar o nome do(a) participante ao código de identificação serão armazenados em computador pessoal do pesquisador. Todo o material de pesquisa será mantido em arquivo digital, sob sua guarda e responsabilidade, por um período de 5 anos após o término da pesquisa, conforme determinado na Resolução nº 466/2012. Por ocasião da publicação dos resultados da pesquisa, o nome do(a) participante será mantido em sigilo absoluto, sendo utilizados os códigos marcados nas gravações para identificação dos(as) participantes. As gravações serão utilizadas somente para coleta de dados, sendo garantida a manutenção do sigilo e da privacidade da participação do(a) menor pelo(a) qual o(a) sr.(^a) é responsável, bem como de todos os seus dados, durante todas as fases da pesquisa e posterior divulgação científica.

Os benefícios esperados da pesquisa para a comunidade são: identificação de estratégias para o desenvolvimento de habilidades básicas de programação de computadores junto a estudantes com deficiência visual; identificação de vantagens e desvantagens na utilização da programação de computadores como recurso para resolução de problemas matemáticos por estudantes com deficiência visual; identificação de vantagens e desvantagens específicas da linguagem de programação utilizada; identificação de caminhos para o desenvolvimento de habilidades do pensamento computacional junto a estudantes com deficiência visual. Para o participante, os benefícios são: o desenvolvimento de habilidades introdutórias de programação de computadores e o desenvolvimento de habilidades para construção de algoritmos computacionais para resolução de problemas matemáticos.

Em caso de dúvidas relacionadas aos aspectos éticos deste estudo, você poderá consultar o Comitê de Ética em Pesquisa em Ciências Sociais, Sociais Aplicadas, Humanas, Letras, Artes e Linguística (CEP Humanas UFF) no endereço: (Rua Passo da Pátria, 156 - São Domingos - Niterói, Rio de Janeiro. Campus da Praia Vermelha da UFF - Instituto de Física (torre nova - 3º andar).

Telefone: (21) 2629-5119
e-mail: cephumanasuff@gmail.com

(assinatura do pesquisador)

ASSENTIMENTO DO(DA) MENOR DE IDADE

Eu, _____, declaro ter compreendido as informações apresentadas neste termo de assentimento, e concordo com a participação no projeto de pesquisa acima descrito.

Niterói, ____ de _____ de _____

(nome e assinatura do participante)

* Este termo foi disponibilizado em Braille ou em áudio para participantes cegos

APÊNDICE C – ROTEIRO DAS ENTREVISTAS

A. Identificação:

A.1. Nome:

A.2. Idade:

A.3. Ano de escolarização:

B. Interesses acadêmicos

B.1. Com qual ou quais disciplinas você tem mais facilidade?

B.2. Com qual ou quais disciplinas você tem menos facilidade?

B.3. Você gosta de matemática? Por quê?

B.3.1. *Se respondeu sim ao item B3.* Você sempre gostou, ou passou a gostar com o tempo? Se passou a gostar com o tempo, o que fez com que isso ocorresse?

B.3.2. *Se respondeu não ao item B3.* Você nunca gostou, ou passou a não gostar com o tempo? Se passou a não gostar com o tempo, o que fez com que isso ocorresse?

B.4. Você teve dificuldades com matemática ao longo do Ensino Fundamental? Por quê?

B.5. Você acha que as aulas e os materiais didáticos foram adaptados adequadamente durante o Ensino Fundamental?

C. Utilização do computador

C.1. Você tem computador em casa?

Sim

Não

C.2. Como você avalia o seu uso do teclado do computador?

Consigo utilizar muito bem o teclado do computador.

Consigo utilizar bem o teclado do computador.

Tenho um pouco de dificuldade para utilizar o teclado do computador.

Tenho muita dificuldade para utilizar o teclado do computador.

C.3. Como você avalia o seu uso do leitor de telas no computador?

Consigo utilizar muito bem o leitor de telas no computador.

Consigo utilizar bem o leitor de telas no computador.

Tenho um pouco de dificuldade para utilizar o leitor de telas no computador.

Tenho muita dificuldade para utilizar o leitor de telas no computador.

C.4. Que leitor(es) de telas você utiliza no dia a dia?

NVDA

JAWS

Virtual Vision

Outro(s). Qual? _____

C.5. Você já utilizou o computador nas aulas de matemática?

- Sim
 Não

Se sim, em que tipo de situação o computador foi utilizado?

C.6. Você prefere utilizar instrumentos para escrita em Braille ou o computador para escrever?

- Instrumentos para escrita em Braille
 Computador

C.7. Você prefere utilizar instrumentos para escrita em Braille ou o computador para ler?

- Instrumentos para escrita em Braille
 Computador

C.8. Você tem o costume de utilizar instrumentos para escrita em Braille para resolver e descrever o processo de resolução de problemas matemáticos?

- Sim
 Não

Se sim, o que você acha do uso desses recursos para resolver e descrever o processo de resolução de problemas matemáticos?

C.9. Você tem o costume de utilizar o computador para resolver e descrever o processo de resolução de problemas matemáticos?

- Sim
 Não

Se sim, o que você acha do uso desses recursos para resolver e descrever o processo de resolução de problemas matemáticos?

D. Programação de computadores e ILA

D.1. Você conhecia alguma linguagem de programação antes do ILA?() Sim

- Não

Se sim, qual?

D.2. Você já havia participado de algum curso ou atividade de formação sobre alguma linguagem de programação?

- Sim
 Não

Se sim, qual?

D.3 Após a realização das atividades com o ILA, o que significa programação de computadores para você?

D.4. Você teve dificuldades com o ILA?() Não tive nenhuma dificuldade.

- Tive dificuldade apenas em alguns momentos.() Tive dificuldade em vários momentos.
 Tive dificuldade em todos os momentos.

Se respondeu que teve dificuldade, aponte os aspectos em que teve dificuldade:

- Tive dificuldade com os comandos do programa (variáveis, início, escrever, ler,...) Tive dificuldade com a determinação das variáveis
- Tive dificuldade com os tipos de variáveis (numérico, caracter, logico) Tive dificuldade com comandos de entrada e saída (ler, escrever)
- Tive dificuldade com operadores aritméticos (+, -, *, :)
- Tive dificuldade com o leitor de tela

Fale, de maneira geral, um pouco mais sobre as dificuldades que você teve com o ILA.

D. 5. Você teve problemas com o leitor de tela ao programar utilizando o ILA? Se sim, que tipo de problema?

D.6. Você gostou de programar utilizando o ILA? Por quê?

E. Utilização do ILA para resolução de problemas

E.1. Você teve dificuldade em utilizar o ILA para construir algoritmos para resolução de problemas matemáticos?

- Sim
- Não

Se sim, quais foram as suas dificuldades?

E.2. Ordene, de 1 a 4, as alternativas abaixo de acordo com o seu grau de dificuldade para você, da que você teve mais dificuldade para a que você teve menos dificuldade (1 para a mais difícil até 4 para a menos difícil).

- Selecionar as informações realmente importantes para a resolução do problema
- Dividir o problema em etapas menores
- Lembrar de outros problemas já resolvidos que pudessem ajudar na resolução daquele problema
- Construir o algoritmo para resolução do problema

E.3. Após as atividades desenvolvidas, você acha que a construção de programas de computadores pode ser útil para resolução de problemas matemáticos? Por quê?

E.4. Para você, quais os pontos positivos de utilizar o ILA para construir programas para resolução de problemas matemáticos?

E.5. Para você, quais os pontos negativos de utilizar o ILA para construir programas para resolução de problemas matemáticos?

APÊNDICE D – ROTEIRO DOS ENCONTROS DA SEQUÊNCIA DIDÁTICA

ENCONTRO 1

Temas:

- **Programação de computadores e o ILA;**
- **Entrada, processamento e saída de dados.**

Objetivos:

- Dialogar sobre os conceitos de programação de computadores e linguagens de programação;
- Apresentar o ILA;
- Desenvolver as ideias de entrada, processamento e saída de dados;
- Apresentar os comandos **início**, **fim**, **variáveis**, **ler** e **escrever** do ILA.

Habilidades do pensamento computacional priorizadas:

- Reconhecimento de padrões, decomposição, algoritmos.

Recursos:

- Computadores com ILA e NVDA instalados.

Avaliação:

- Construção de um programa de computador que receba uma informação e apresente na tela uma frase que contenha a informação recebida.

Momento 1: Diálogo sobre programação de computadores, linguagens de programação e o ILA

Duração: 10 minutos

Inicialmente serão desenvolvidas as ideias de programa de computador e linguagem de programação. Nesse momento será apresentado o ILA, interpretador que será utilizado ao longo das sequências didáticas.

Momento 2: Entrada, processamento e saída de dados; comandos ler e escrever; tipos de variáveis

Duração: 20 minutos

Serão desenvolvidos os conceitos de entrada, processamento e saída de dados em programação. Nesse momento, serão apresentados os comandos **ler** e **escrever** do ILA.



Será colocado que, em geral, o computador realizará alguma ação de processamento em relação aos dados de entrada. Para representar a entrada, utilizamos uma variável que pode ser de diferentes tipos. Nesse momento, será trabalhado o conceito de variável, bem como o comando **variáveis** e os tipos de variáveis.

Momento 3: estrutura de um programa em ILA e reconhecimento de padrões

Duração: 20 minutos

Será apresentada a estrutura de um programa em ILA. Nesse momento serão apresentados os comandos **início** e **fim**. Colocar que no ILA os programas são constituídos, em geral, utilizando a seguinte estrutura:

- Declaração de variáveis
- Comando início
- Blocos de instruções
- Comando fim

Em seguida, serão apresentados dois programas:

Programa 1:

```
início
escrever "Eu sou estudante do Colegio Pedro II"
fim
```

Perguntar aos participantes:

- 1- Esse programa tem atividades de entrada, de saída ou as duas?
- 2- O que você acredita que acontecerá nesse programa?

Programa 2:

```
variaveis
caracter nome
início
escrever "Digite seu nome"
ler nome
escrever nome, "e estudante do Colegio Pedro II"
fim
```

Perguntar aos participantes:

- 1- Esse programa tem atividades de entrada, de saída ou as duas?
- 2- O que acontece nas duas primeiras linhas do programa?
- 3- Por que foi utilizada o termo caracter na segunda linha?
- 4- E o termo nome da segunda linha, o que representa no programa?
- 5- Tanto nesse quanto no outro programa, que símbolo veio logo após a palavra escrever?
- 6- O que você acredita que acontecerá nesse programa?

A partir das respostas dos estudantes, reforçar os seguintes pontos:

- Não colocar acentos ou cedilha nas frases que representam saídas (dentro das aspas)
- Se quisermos que a saída seja uma variável, basta utilizar o comando **escrever**, seguido do termo utilizado na declaração da variável
- Se quisermos que a saída seja uma frase definida, basta utilizar o comando escrever “ “. Nesse caso, tudo que está dentro das aspas aparecerá na tela
- Se quisermos que a saída tenha uma variável e também uma frase definida, devemos separá-los por vírgula.
- Reforçar a importância de reconhecer padrões em problemas e programas anteriores, para aplicá-los a novos problemas e programas.

Momento 4 (avaliação): construção de um programa em ILA

Duração: 30 minutos

Tomando por base as estruturas observadas nos programas analisados pelos estudantes, estes serão estimulados a construir um programa que cumpra as seguintes instruções:

- Perguntar ao usuário qual a sua idade.
- Apresentar na tela do computador a seguinte frase “A idade do usuário é ‘tanto’”, onde “tanto” é

a idade do usuário.

Momento 5: desenvolvendo a ideia de avaliação do programa construído (habilidade metacognitiva)

Nesse momento, o estudante será estimulado a refletir sobre o programa construído, avaliando se o mesmo funciona adequadamente. Nesse sentido, o estudante deverá avaliar se:

1- O programa roda?

Caso não rode, verificar onde está o problema. Nesse momento, o estudante deverá refletir sobre a mensagem de erro retornada

2- O programa funciona adequadamente em diferentes casos?

Para tal, deverão ser realizados diferentes testes de modo que se verifique se o retorno do programa é adequado em diferentes cenários. Nesse momento, perguntar aos estudantes que testes eles poderiam realizar para avaliar se o programa funciona adequadamente.

ENCONTRO 2

Operadores matemáticos

Objetivos:

- Apresentar a representação de alguns operadores matemáticos no ILA;
- Construir programas que realizem operações simples utilizando operadores matemáticos.

Habilidades do pensamento computacional priorizadas:

- Decomposição, reconhecimento de padrões, algoritmos

Recursos:

- Computadores com ILA e NVDA instalados

Avaliação:

- Construção de um programa de computador que realize uma operação aritmética simples.

Momento 1: apresentação dos principais operadores matemáticos e suas expressões no ILA

Duração: 10 minutos

Após apresentação dos objetivos do encontro, no primeiro momento serão apresentados os operadores matemáticos utilizados no ILA para realização de operações de soma, subtração, multiplicação, divisão e exponenciação.

Momento 2: estrutura de um programa que envolva operações aritméticas

Duração: 40 minutos

Nesse momento será apresentado aos estudantes o seguinte programa:

```
variaveis
numerico a, b, soma
inicio
escrever "Digite o primeiro numero:"
ler a
escrever "Digite o segundo numero:"
ler b
soma = a + b
escrever "O resultado equivale a: ", soma
fim
```

Perguntar aos participantes:

- 1- Esse programa apresenta comandos de entrada, de saída ou os dois?
- 2- O que isso significa?
- 3- O que significam as duas primeiras linhas do programa?
- 4- O que você acha que esse programa faz?
- 5- O que você acha que acontece na linha 8 do programa?
- 6- O que você acha que acontece na linha 9 do programa?

A partir das respostas dos estudantes, reforçar:

- O programa realiza uma operação de processamento a partir de dados de entrada. Essa operação é representada na linha 8.

- A operação de saída na linha 9 retorna um texto fixo, seguido do valor assumido pela variável (o resultado da soma). Nesse caso, é necessário separar os dois trechos por vírgula.

Perguntar aos participantes:

- 1- A partir da estrutura anterior, o que deveríamos modificar para criar um programa que multiplicasse dois números.
- 2- O que deveríamos modificar para criar um programa que somasse três números. Pedir que implementem a ideia no programa.

Momento 3: construção de um programa que realize uma operação aritmética simples

Duração: 30 minutos

Nesse momento, os estudantes serão estimulados a construir um programa que calcule o quadrado de um número. Reforçar que os estudantes devem avaliar o programa construindo, criando uma série de testes para verificar se o programa funciona adequadamente.

ENCONTRO 3

Desenvolvendo a ideia de algoritmo e retomando o conceito de função

Objetivos:

- Apresentar a representação de novos operadores matemáticos no ILA;
- Rever o conceito de função afim;
- Construir um programa que envolva o conceito de função afim.

Habilidades do pensamento computacional priorizadas:

- Algoritmos, reconhecimento de padrões, abstração, decomposição

Recursos:

- Computadores com ILA e NVDA instalados

Avaliação:

- Construção de um programa de computador que envolva o conceito de função afim

Momento 1: revendo o conceito de função

Duração: 15 minutos

Nesse momento será retomado junto aos estudantes o conceito de função. Inicialmente, buscaremos identificar os conhecimentos prévios dos estudantes em relação ao conteúdo. A partir das interações conduzidas, o objetivo é reforçar a ideia de função enquanto relação entre duas ou mais variáveis. Em seguida, será caracterizada a função do primeiro grau, bem como sua forma padrão $f(x)=ax+ b$, ou $y=ax+b$.

Momento 2: desenvolvendo a ideia de algoritmo

Duração: 15 minutos

Nesse momento será apresentado o conceito de algoritmo aos estudantes. Aqui, buscaremos relacionar a ideia de algoritmo à noção de “receita”, uma sequência de passos que deverão ser seguidos para solucionar um problema ou atingir um objetivo. Inicialmente, será discutido o algoritmo de uma “ida à escola”.

Momento 3: construindo um algoritmo para calcular o dobro de um número

Duração: 20 minutos

Estabelecendo uma relação entre as atividades anteriores, nesse momento os estudantes serão estimulados a construir um algoritmo para a criação de um programa que solicite um número ao usuário um número e retorne o dobro do mesmo. O objetivo é que os mesmos construam uma sequência similar à seguinte:

- Solicitar o número ao usuário
- Armazenar o número
- Realizar a operação de dobrar o número, atribuindo o resultado a uma variável
- Exibir o valor do resultado na tela do computador

Momento 4 (construção do programa/avaliação): realizando a operação de conversão do algoritmo construído para a linguagem ILA

Duração: 30 minutos

Nesse momento, os estudantes serão estimulados a representarem o algoritmo desenvolvido utilizando a linguagem ILA. O objetivo é que os estudantes construam um programa que retorne o dobro de um número fornecido pelo usuário. Na mediação do processo, podem ser realizadas as seguintes perguntas aos estudantes: i) o que é o dobro de um número? ii) que operação devemos realizar para ter como resultado o dobro de um número? iii) você consegue pensar em alguma função que represente essa situação?

Reforçar a ideia de que os estudantes devem testar o programa construído, avaliando se o mesmo roda adequadamente.

ENCONTRO 4

Programação condicional: a instrução se então

Objetivos:

- Apresentar a representação de alguns operadores relacionais no ILA
- Apresentar a instrução **se então**
- Construir programas utilizando a instrução **se então** no ILA

Habilidades do pensamento computacional priorizadas:

- Abstração, decomposição, reconhecimento de padrões, algoritmos, avaliação

Recursos:

- Computadores com ILA e NVDA instalados

Avaliação:

- Construção de um programa de computador que utilize a instrução **se então** no ILA

Momento 1: apresentação dos principais operadores relacionais e suas expressões no ILA

Duração: 15 minutos

Nesse momento serão apresentados os principais operadores utilizados para estabelecimento de uma relação entre elementos (>, <, =, >=, <=).

Momento 2: apresentando a instrução se então

Duração: 40 minutos

Nesse momento será apresentada a instrução **se então** aos participantes. Reforçar que tal instrução será utilizada quando for necessário tomar uma decisão para realizar uma ação. Para tal, inicialmente será apresentado o seguinte programa:

```
variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
fim_se
fim
```

Perguntar aos estudantes:

- 1) Neste programa, quais linhas apresentam comandos que você ainda não conhecia?
- 2) O que você acredita que esse programa faz?
- 3) O que você acha que aparecerá na tela se o usuário digitar o número 7?
- 4) Considerando a maneira como o programa está escrito, o que você acha que irá acontecer se o usuário digitar o número 15?

A partir das interações com os estudantes, perguntar se eles têm alguma sugestão para solucionar o problema ocorrido quando o usuário digitou o número 15. Em seguida, pedir que os estudantes observem o seguinte programa:

```
variaveis
```

```

numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
fim_se
se (a>10) entao
escrever "O numero digitado e maior que 10"
fim_se
fim

```

Perguntar aos estudantes:

- 1) O que você acha que esse programa faz?
- 2) Qual a diferença desse programa em relação ao anterior?
- 3) Você acha que agora o programa está construído da melhor maneira possível? Ou seja, como ele está escrito, existe uma resposta adequada para todos os possíveis valores digitados pelo usuário?

3.1. Se o estudante concluir que sim, perguntar: E se o usuário digitar o número 10? A partir da resposta do estudante, pedir que o mesmo faça a alteração necessária no programa de modo que seja considerada essa possibilidade.

3.2. Se o estudante concluir adequadamente que o programa não leva em conta a possibilidade de o usuário digitar o número 10, pedir que o mesmo faça a alteração necessária no programa de modo que seja considerada essa possibilidade.

A partir das interações com os estudantes, reforçar:

- A estrutura utilizada na instrução se entao é da forma:

```

se (proposição cuja veracidade será avaliada) entao
ação a ser realizada se a proposição for verdadeira
fim_se

```

- A ação que será realizada se a condição for satisfeita pode ser de vários tipos. Aqui, usamos o comando escrever, mas poderíamos ter utilizado uma ação de processamento, ou um bloco de ações.
- É importante considerarmos diferentes possíveis respostas do usuário, e verificar se o programa tem uma resposta adequada para esses diferentes tipos de resposta. Ou seja, é importante realizar uma bateria de testes ao fim do processo de programação e avaliar se o programa funciona adequadamente em diferentes cenários.

Momento 3 (avaliação): construindo um programa utilizando a instrução se entao

Duração: 25 minutos

Neste momento, os estudantes serão estimulados a construir um programa que remeta ao conceito de função afim, utilizando a instrução se entao. Para isso, apresentar a seguinte situação:

Em uma loja onde são vendidos materiais escolares, cada caderno custa R\$8,00. No entanto, a gerência criou uma promoção em que se o cliente compra a partir de 4 cadernos, cada um sai por R\$6,50. Construa um programa que retorne o preço que o usuário irá pagar a partir da quantidade de cadernos comprada.

Para auxiliar no processo de construção do código, perguntar aos estudantes: i) o valor a ser pago por cada

caderno depende da quantidade de cadernos que serão comprados?; ii) nesse caso, podemos pensar em uma única função que poderia representar o valor total a ser pago pelo cliente?; iii) poderíamos dividir a situação descrita em duas ou mais partes? iv) qual seria a função que representaria o valor a ser pago em cada um desses casos?

A partir das interações realizadas, reforçar com os estudantes que:

- Em alguns casos, pode ser útil **decompor** o problema em etapas menores, auxiliando na resolução do problema geral.

ENCONTRO 5

Programação condicional (continuação)

Objetivos:

- Apresentar o comando **senao** no ILA
- Construir programas utilizando a instrução **senao** no ILA

Habilidades do pensamento computacional priorizadas:

- Decomposição, reconhecimento de padrões, algoritmos

Recursos:

- Computadores com ILA e NVDA instalados

Avaliação:

- Construção de um programa de computador que utilize a instrução **se então** no ILA

Momento 1: apresentando a instrução **senao**

Duração: 40 minutos

Nesse momento, será apresentada a instrução **senao** no ILA. Para tal, inicialmente será retomado um programa trabalhado no encontro anterior, com uma pequena alteração incluindo um operador relacional. Nesse sentido, pedir que os estudantes observem o seguinte código:

```
variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
fim_se
se (a>=10) entao
escrever "O numero digitado não e menor que 10"
fim_se
fim
```

Perguntar aos estudantes:

1. Existe algum símbolo ou comando nesse programa que você ainda não tinha utilizado; Se sim, qual você acha que é o significado dele?
2. O que você acha que esse programa faz?

Em seguida, pedir que os estudantes observem o seguinte programa:

```
variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
senao
escrever "O numero digitado nao e menor que 10"
fim_se
fim
```

Perguntar aos estudantes:

1. Existe algum símbolo ou comando nesse programa que você ainda não tinha utilizado? Se sim, qual você acha que é o significado dele?
2. O que você acha que esse programa faz?

A partir das interações com os estudantes, reforçar:

- A estrutura utilizada na instrução se então senao é da forma:

```
se (proposição cuja veracidade será avaliada) entao
ação a ser realizada se a proposição for verdadeira
senao
ação a ser realizada se a proposição for falsa
fim_se
```

- Podemos construir programas que realizam ações semelhantes a partir de comandos diferentes. Não existe uma única maneira correta de programar. No caso, construímos programas parecidos usando as estruturas se entao, e se entao senao.

Em seguida, pedir que os estudantes realizem as alterações necessárias no programa para que o mesmo realize a seguinte ação:

Pedir que o usuário digite um número. Se o número for menor ou igual a 100, deve aparecer na tela “O numero nao e maior que 100”. Se o número for maior que 100, deve aparecer na tela “O numero e maior que 100”

Momento 3 (avaliação): construção de um programa onde pode ser utilizada a instrução senao
Duração: 20 minutos

Nesse momento, os estudantes serão estimulados a construir um programa que resolva à seguinte situação problema:

Uma companhia telefônica realiza a seguinte promoção para seus clientes:

- Se o cliente utilizar até 40MB, é cobrado um valor fixo de RS50, mais RS 2 por Megabite utilizado.
- Se o cliente utilizar mais que 40MB, o cliente segue pagando RS2 por cada Megabite utilizado, mas o valor fixo cai pra RS 40.

Construa um programa que solicite ao usuário a quantidade de megabites utilizada no mês e retorne o valor a ser pago pelo cliente.

ENCONTRO 6

Avaliação (com possibilidade de consulta)

Objetivos:

- Resolver situações problema que se relacionam com a função afim através da construção de programas no ILA

Habilidades do pensamento computacional priorizadas:

- Abstração, decomposição, reconhecimento de padrões, algoritmos

Recursos:

- Computadores com ILA e NVDA instalados

Avaliação:

- Construção de programas de computador com o objetivo de resolver duas situações problema

Momento 1: apresentação da dinâmica de avaliação

Duração: 5 minutos

Neste momento, será apresentada aos participantes a dinâmica da atividade. Será colocado que os participantes terão 1h para a construção de dois programas de computador, com o objetivo de modelar duas situações problema apresentadas. Os participantes poderão consultar um arquivo auxiliar que contém todos os programas trabalhados nas sequências didáticas anteriores.

Momento 2: construção dos programas

Duração: 1h

Nesse momento os participantes deverão construir programas que modelem as seguintes situações problema:

Situação 1

Nos últimos anos, aplicativos para transportes de passageiros em automóveis privados têm se multiplicado, oferecendo serviços voltados tanto para o público geral, quanto para grupos específicos. Ainda assim, em virtude de problemas como excesso de cancelamento de viagens e aumento de tarifas, os táxis por vezes continuam representando uma opção atrativa no transporte de passageiros. Na cidade de Taxilandia, por exemplo, a corrida de táxi é calculada da seguinte maneira:

- É cobrada a taxa de R\$6 referente à bandeirada (valor fixo inicial de qualquer corrida), independente do horário em que seja realizada a corrida;

- É cobrado o valor adicional de R\$3 por quilômetro rodado.

Nessas condições, construa um programa que determine o valor a ser pago pelo usuário de acordo com a quantidade de quilômetros rodados.

Situação 2

A comissão é um benefício muitas vezes utilizado no setor comercial para motivar os empregados e, conseqüentemente, aumentar os lucros da empresa. Tulio é proprietário de uma loja de tapetes, e para tentar

impulsionar as vendas no período de Natal, decidiu conceder, além das comissões, um bônus aos funcionários que atingissem a meta estabelecida para a quantidade de tapetes vendidos. Assim, a renda mensal dos vendedores de sua loja passou a ser composta por uma parte fixa de R\$1500, e uma parte variável que dependia da quantidade de tapetes que o vendedor conseguisse vender. Para o cálculo da parte variável, ficou estabelecido que cada vendedor ganharia uma comissão de R\$ 16 por tapete vendido e, além disso, se o vendedor conseguisse vender mais de 100 tapetes no mês, ganharia adicionalmente um bônus de R\$300.

Nessas condições, construa um programa que Tulio poderia utilizar para calcular o salário dos vendedores da loja de acordo com a quantidade de tapetes vendidos.

Os estudantes poderão consultar um arquivo auxiliar que contém os seguintes programas:

Programa 1

```
inicio
escrever "Eu sou estudante do Colegio Pedro II"
fim
```

Programa 2

```
variaveis
caracter nome
inicio
escrever "Digite seu nome"
ler nome
escrever nome, "e estudante do Colegio Pedro II"
fim
```

Programa 3

```
variaveis
numerico a, b, soma
inicio
escrever "Digite o primeiro numero:"
ler a
escrever "Digite o segundo numero:"
ler b
soma = a + b
escrever "O resultado equivale a: ", soma
fim
```

Programa 4

```
variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
fim_se
se (a>10) entao
escrever "O numero digitado e maior que 10"
fim_se
fim
```

Programa 5

```
variaveis
numerico a
inicio
escrever "Digite o numero"
ler a
se (a<10) entao
escrever "O numero digitado e menor que 10"
senao
escrever "O numero digitado nao e menor que 10"
```

fim_se
fim

ENCONTRO 7

Avaliação (sem possibilidade de consulta)

Objetivos:

- Resolver situações problema que se relacionam com a função afim através da construção de programas no ILA

Habilidades do pensamento computacional priorizadas:

- Abstração, decomposição, reconhecimento de padrões, algoritmos

Recursos:

- Computadores com ILA e NVDA instalados

Avaliação:

- Construção de programas de computador com o objetivo de resolver duas situações problema

Momento 1: apresentação da dinâmica de avaliação

Duração: 5 minutos

Neste momento, será apresentada aos participantes a dinâmica da atividade. Será colocado que os participantes terão 1h para a construção de dois programas de computador, com o objetivo de modelar duas situações problema apresentadas. Os participantes não terão nenhuma fonte de consulta.

Momento 2: construção dos programas

Duração: 1h

Nesse momento os participantes deverão construir programas que modelem as seguintes situações problema:

Situação 3

Mariana ganhou de sua mãe como presente de Natal um cofre que já continha uma nota de R\$200. Animada com o presente, decidiu, a partir do mês seguinte, depositar R\$35 no cofre a cada mês. Construa um programa que determine quantos reais haverá no cofre de acordo com a quantidade de meses passados.

Situação 4

Academias de ginástica e musculação têm se espalhado por todo o país, passando a fazer parte do cotidiano de muitos brasileiros. No entanto, é comum encontrar pessoas que inicialmente se engajam na realização de atividades físicas mas, com o passar o tempo, passam a se sentir desmotivadas e abandonam a prática. Pensando em atrair novos clientes e estimular a realização de exercícios físicos entre eles, uma academia lançou a seguinte promoção:

- Se o cliente frequentar a academia menos que 10 horas no mês, deve pagar apenas um valor fixo de R\$80.

- Se o cliente frequentar a academia 10 ou mais horas no mês, o valor fixo cai para R\$30, e é cobrado um adicional de R\$2 por cada hora utilizada na academia.

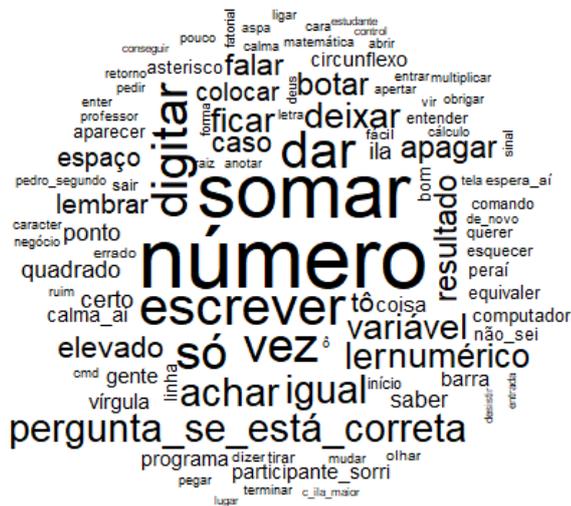
Construa um programa que determine o valor a ser pago pelo cliente de acordo com a quantidade de horas frequentadas na academia.

Figura 63 – Nuvem de palavras – Encontro 1



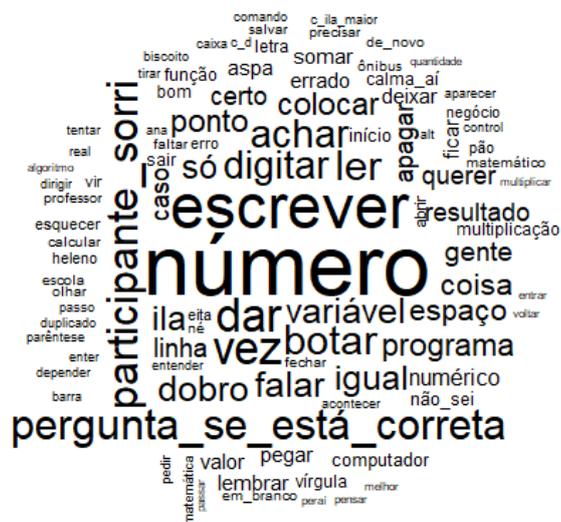
Fonte: elaborado pelo autor (2024).

Figura 64: Nuvem de palavras – Encontro 2



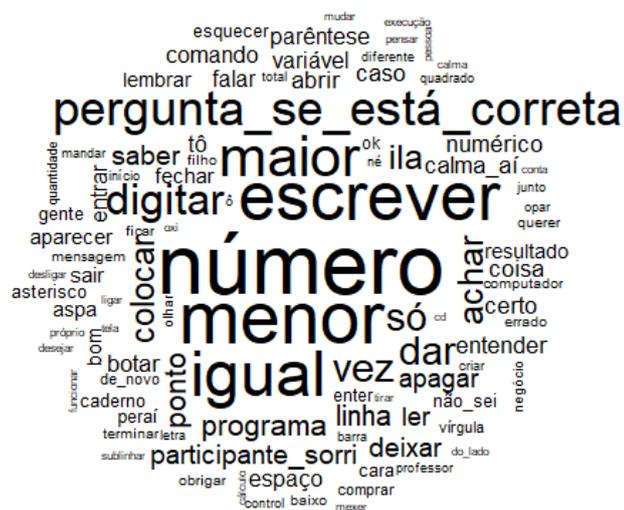
Fonte: elaborado pelo autor (2024).

Figura 65: Nuvem de palavras – Encontro 3



Fonte: elaborado pelo autor (2024).

Figura 66: Nuvem de palavras – Encontro 4



Fonte: elaborado pelo autor (2024).

