



**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
**CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA**  
**PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO**

**ELAINE CRISTINA JUVINO DE ARAÚJO**

**UMA ABORDAGEM PARA CRIAÇÃO DE INSTRUMENTOS PEDAGÓGICOS**  
**NO CONTEXTO DO ENSINO DE PROGRAMAÇÃO PARA PESSOAS**  
**NEURODIVERGENTES**

**CAMPINA GRANDE – PB**

**2024**

**ELAINE CRISTINA JUVINO DE ARAÚJO**

**UMA ABORDAGEM PARA CRIAÇÃO DE INSTRUMENTOS PEDAGÓGICOS  
NO CONTEXTO DO ENSINO DE PROGRAMAÇÃO PARA PESSOAS  
NEURODIVERGENTES**

Tese apresentada ao Programa do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I, pertencente à linha de pesquisa Educação em Ciência da Computação e área de concentração Ciência da Computação, como requisito para obtenção do grau de Doutor em Ciência da Computação.

Orientador: Prof. Dr. Wilkerson de Lucena Andrade

**CAMPINA GRANDE – PB**

**2024**

A663a Araújo, Elaine Cristina Juvino de.  
Uma abordagem para criação de instrumentos pedagógicos no contexto do ensino de programação para pessoas neurodivergentes / Elaine Cristina Juvino de Araújo. – Campina Grande, 2024.  
183 f. : il. color.

Tese (Doutorado em Ciência da Computação) – Universidade Federal de Campina Grande, Centro de Engenharia Elétrica e Informática, 2024.  
"Orientação: Prof. Dr. Wilkerson de Lucena Andrade".  
Referências.

1. Educação em Computação. 2. Metodologia e Técnicas da Computação. 3. Ensino de Programação. 4. Neurodiversidade. I. Andrade, Wilkerson de Lucena. II. Título.

CDU 004(07)(043)



MINISTÉRIO DA EDUCAÇÃO  
**UNIVERSIDADE FEDERAL DE CAMPINA GRANDE**  
POS-GRADUACAO EM CIENCIA DA COMPUTACAO  
Rua Aprígio Veloso, 882, Edifício Telmo Silva de Araújo, Bloco CG1, - Bairro Universitário, Campina Grande/PB, CEP 58429-900  
Telefone: 2101-1122 - (83) 2101-1123 - (83) 2101-1124  
Site: <http://computacao.ufcg.edu.br> - E-mail: [secpg@computacao.ufcg.edu.br](mailto:secpg@computacao.ufcg.edu.br)

## FOLHA DE ASSINATURA PARA TESES E DISSERTAÇÕES

**ELAINE CRISTINA JUVINO DE ARAÚJO**

### UMA ABORDAGEM PARA CRIAÇÃO DE INSTRUMENTOS PEDAGÓGICOS NO CONTEXTO DO ENSINO DE PROGRAMAÇÃO PARA PESSOAS NEURODIVERGENTES

Tese apresentada ao Programa de Pós-Graduação em Ciência da Computação como pré-requisito para obtenção do título de doutor em Ciência da Computação.

Aprovada em: 02/09/2024

Prof. Dr. WILKERSON DE LUCENA ANDRADE, UFCG, Orientador

Profa. Dra. LÍVIA MARIA RODRIGUES SAMPAIO CAMPOS, UFCG, Examinadora Interna

Profa. Dra. JOSEANA MACÊDO FECHINE RÉGIS DE ARAÚJO, UFCG, Examinadora Interna

Profa. Dra. YUSKA PAOLA COSTA AGUIAR, UFPB, Examinadora Externa

Prof. Dr. ECIVALDO DE SOUZA MATOS, UFBA, Examinador Externo



Documento assinado eletronicamente por **WILKERSON DE LUCENA ANDRADE, PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 30/09/2024, às 14:05, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **JOSEANA MACEDO FECHINE, PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 02/10/2024, às 21:07, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).



Documento assinado eletronicamente por **LIVIA MARIA RODRIGUES SAMPAIO CAMPOS, PROFESSOR(A) DO MAGISTERIO SUPERIOR**, em 07/10/2024, às 08:55, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).

---



Documento assinado eletronicamente por **Yuska Paola Costa Aguiar, Usuário Externo**, em 14/10/2024, às 09:20, conforme horário oficial de Brasília, com fundamento no art. 8º, caput, da [Portaria SEI nº 002, de 25 de outubro de 2018](#).

---



A autenticidade deste documento pode ser conferida no site <https://sei.ufcg.edu.br/autenticidade>, informando o código verificador **4759082** e o código CRC **F30CBCFF**.

---

Universidade Federal de Campina Grande  
Centro de Engenharia Elétrica e Informática  
Coordenação de Pós-Graduação em Ciência da Computação

Uma Abordagem para Criação de Instrumentos  
Pedagógicos no contexto do Ensino de Programação  
para Pessoas Neurodivergentes

Elaine Cristina Juvino de Araújo

Tese submetida à Coordenação do Curso de Pós-Graduação em Ciência da Computação da Universidade Federal de Campina Grande - Campus I como parte dos requisitos necessários para obtenção do grau de Doutor em Ciência da Computação.

Área de Concentração: Ciência da Computação  
Linha de Pesquisa: Metodologia e Técnicas da Computação

Wilkerson de Lucena Andrade  
(Orientador)

Campina Grande, Paraíba, Brasil

©Elaine Cristina Juvino de Araújo, 02/09/2024

## Resumo

Este estudo é uma Pesquisa em Design Educacional que visa favorecer a inclusão de estudantes neurodiversos no ensino de programação, disponibilizando meios para isso. Diante do exposto, foi fornecido aos professores de programação um artefato educacional inclusivo para o público neurodiverso. Para atingir esse objetivo, esta tese de doutorado centra-se nas seguintes questões de pesquisa (QP): QP1 - Quais os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?, QP2 - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória? e, QP3 - Como tornar o ensino de programação inclusivo para pessoas neurodivergentes? Para isso realizamos estudos identificando os impactos que as deficiências cognitivas podem gerar nas habilidades necessárias para programar, investigando também as percepções de professores e estudantes neurodiversos em aulas de programação introdutória. Por fim, realizamos uma pesquisa para elaboração do artefato. O artefato proposto foi construído na forma de um guia para que professores de programação introdutória possam tornar suas aulas inclusivas ao público neurodiverso. O guia possui orientações para adaptação de aulas teóricas, exercícios e avaliações. Além disso, apresenta modelos para criação de planos e roteiros de aula pensados para esse público alvo. O método Delphi foi utilizado para validação deste artefato por meio da avaliação por treze juízes especialistas, sendo nove professores de programação e quatro profissionais de pedagogia. Além disso, conduzimos um estudo com aluno neurodiverso para aplicação dos artefatos, tendo o ponto de vista do aluno. Diante do exposto, a tese ora descrita, possui como principal contribuição, a proposta de um artefato educacional, para construção de aulas de programação introdutória, inclusivo à estudantes com deficiência cognitiva ou neurodiversos. Este artefato surge como uma solução prática para que professores possam tornar suas aulas mais inclusivas, para favorecer a aprendizagem desses estudantes. A fim de gerar um artefato que fosse de fato viável para uso do professor, foi feita uma validação atenciosa, inserindo professores de programação no processo, além de profissionais de Pedagogia que dão suporte a estudantes com deficiência. Diante disso, esta pesquisa almejou proporcionar aos estudantes neurodiversos, igualdade de oportunidades para aprender programação, principalmente se tratando de programação introdutória, disciplina tão importante para que eles avancem em

um curso de Computação, seja como técnico de nível médio ou graduado, e que possam obter êxito no curso.

## **Abstract**

This study is an Educational Design Research that aims to promote the inclusion of neurodiverse students in programming education, providing means for the inclusion of these students in introductory programming classes. To this end, programming teachers were provided with an educational artifact that is inclusive of neurodiverse students. To achieve this goal, this doctoral thesis focuses on the following research questions (RQ): RQ1 - What possible impacts can cognitive disabilities have on the skills involved in teaching programming? RQ2 - What perception of difficulties do neurodiverse students and teachers face in the context of introductory programming? and, RQ3 - How can programming education be made inclusive for neurodivergent people? To this end, we conducted studies to identify the impacts that cognitive disabilities can have on the skills needed to program and investigate the perceptions of neurodiverse teachers and students in introductory programming classes. Finally, we conducted research to develop the artifact. The proposed artifact was built in the form of a guide so that introductory programming teachers can make their classes inclusive to neurodiverse audiences. The guide contains guidelines for adapting theoretical courses, exercises, and assessments. In addition, it presents models for creating lesson plans and scripts designed for this target audience. The Delphi method was used to validate this artifact through evaluation by expert judges. In addition, we conducted a study with neurodiverse students to validate the artifacts, taking the student's point of view. In view of the above, the thesis described here has as its main contribution the proposal of an educational artifact for the construction of introductory programming classes inclusive to students with cognitive disabilities or neurodiverse individuals. This artifact emerges as a practical solution so that teachers can make their classes more inclusive to favor these students' learning. To generate an artifact that was actually viable for teacher use, a careful validation was carried out, including programming teachers in the process, in addition to Pedagogy professionals who support students with disabilities. In view of this, this research aimed to provide neurodiverse students with equal opportunities to learn to program, especially when it comes to introductory programming, a subject that is so important for them to advance in a Computer Science course, whether as a mid-level technician or graduate and to be successful in the course.

## Agradecimentos

Chegar até aqui foi uma longa jornada. Sempre tive o desejo de fazer um doutorado, mas, após o término do mestrado, a perda da minha mãe fez com que esse sonho adormecesse por vários anos.

Após realizar o sonho da maternidade e com o crescimento da minha amada Olívia, voltei a desejar o doutorado.

Quando finalmente decidi que era hora de reacender esse sonho, precisei de apoio, que veio de várias formas, pelas quais sou profundamente grata...

A Deus, de onde vem toda a minha força. Sem fé, eu nada seria.

Aos meus pais, minha fortaleza, que sempre apoiaram meus estudos, mesmo não tendo tido a oportunidade de seguir com os deles.

Ao meu esposo, que sempre acreditou no meu potencial, suportando comigo todos os desafios, na alegria e na tristeza, na saúde e na doença, por todos os dias de nossas vidas. Todo o meu amor a você, Lucas!

À minha filha Olívia, que todos os dias me lembra o significado de amar alguém mais do que a mim mesma. Mamãe quer que você aprenda a nunca desistir dos seus sonhos, estarei sempre aqui por você!

Camila, minha prima filha do coração, obrigada pela companhia e o carinho de sempre.

A toda minha família e aos meus amigos, que sempre se orgulharam das minhas conquistas e torceram por mim.

Ao meu estimado orientador Wilkerson, que exerceu seu papel com maestria, sempre acreditando em mim, muitas vezes mais do que eu mesma. Não tenho palavras para expressar minha gratidão por essa jornada.

À minha querida coorientadora Ana Liz, que, apesar de não ter sido coorientadora oficialmente (um pequeno detalhe), me orientou com excelência, sempre contribuindo de forma valiosa para esse processo. Muito obrigada por tudo!

Ana Liz e Mirna foram as pessoas que mais me incentivaram a fazer o doutorado na UFCG e a escolher Wilkerson como orientador. Vocês foram essenciais nesse processo. Obrigada pelo apoio e amizade!

Aos queridos colegas que fiz durante o doutorado, em especial Isabelle, Narallyne e Hanna, com quem compartilhei os desafios desses quatro anos, incluindo uma pandemia. Vocês tornaram essa jornada muito melhor com todo o apoio e torcida.

# Conteúdo

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Contexto e Motivação . . . . .	1
1.2	Identificação do Problema . . . . .	4
1.3	Objetivos . . . . .	5
1.4	Metodologia . . . . .	7
1.5	Contribuições . . . . .	8
1.6	Aspectos Éticos . . . . .	9
1.7	Estrutura do Documento . . . . .	9
<b>2</b>	<b>Referencial Teórico</b>	<b>11</b>
2.1	Ensino de Programação . . . . .	11
2.2	Deficiências Cognitivas e Neurodiversidade . . . . .	14
2.2.1	Autismo . . . . .	15
2.2.2	Transtorno de Deficit de Atenção e Hiperatividade . . . . .	17
2.2.3	Transtornos de Aprendizagem . . . . .	18
2.2.4	Deficiência Intelectual . . . . .	19
2.3	Educational Design Research . . . . .	19
2.4	O Método Delphi . . . . .	21
2.5	Considerações Finais do Capítulo . . . . .	22
<b>3</b>	<b>Trabalhos Relacionados</b>	<b>24</b>
3.1	Revisão Sistemática de Literatura . . . . .	25
3.1.1	Protocolo e Planejamento . . . . .	25
3.1.2	Condução . . . . .	25

3.1.3	Triagem . . . . .	26
3.1.4	Resultados . . . . .	26
3.2	Considerações Finais do Capítulo . . . . .	31
<b>4</b>	<b>Metodologia</b>	<b>32</b>
4.1	Metodologia Aplicada . . . . .	32
4.1.1	Fase 1: Análise e Exploração . . . . .	35
4.1.2	Fase 2: Design e Construção . . . . .	35
4.1.3	Fase 3: Avaliação e Reflexão . . . . .	36
4.2	Considerações Finais do Capítulo . . . . .	36
<b>5</b>	<b>Influência da Neurodiversidade no Ensino e Aprendizagem de Programação</b>	<b>37</b>
5.1	QP1 - Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação? . . . . .	37
5.1.1	Metodologia . . . . .	38
5.1.2	Resultados . . . . .	39
5.1.3	Discussão . . . . .	45
5.1.4	Ameaças a Validade . . . . .	46
5.2	QP2 - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória? . . . . .	47
5.2.1	Metodologia . . . . .	48
5.2.2	Resultados . . . . .	52
5.2.3	Discussão . . . . .	56
5.2.4	Ameaças a Validade . . . . .	59
5.3	Considerações Finais do Capítulo . . . . .	59
<b>6</b>	<b>Instrumentos Pedagógicos Adaptados para Pessoas Neurodivergentes</b>	<b>62</b>
6.1	O guia para aulas de programação inclusiva ao público neurodivergente . . . . .	62
6.1.1	Orientações para Plano de Aula . . . . .	63
6.1.2	Orientações para Roteiro de Aula . . . . .	64
6.1.3	Orientações para Adaptação de Conteúdo, Exercícios e Avaliações . . . . .	67
6.2	Validação do guia . . . . .	71

6.2.1	Painel de Juízes Especialistas . . . . .	71
6.2.2	Os formulários de Avaliação . . . . .	74
6.2.3	Análise dos Dados . . . . .	75
6.2.4	A primeira rodada de Delphi . . . . .	75
6.2.5	Segunda Rodada de Delphi . . . . .	81
6.2.6	Terceira Rodada de Delphi . . . . .	90
6.3	Estudo com estudantes . . . . .	93
6.3.1	Participantes . . . . .	93
6.3.2	Materiais e Métodos . . . . .	93
6.3.3	Protocolo . . . . .	94
6.3.4	Análise e Resultados . . . . .	95
6.3.5	Ameaças a Validade . . . . .	97
6.4	Considerações Finais do Capítulo . . . . .	97
<b>7</b>	<b>Considerações Finais</b>	<b>98</b>
7.1	Conclusões . . . . .	98
7.1.1	Implicações dos Achados da QP1 . . . . .	99
7.1.2	Implicações dos Achados da QP2 . . . . .	100
7.1.3	Implicações dos Achados da QP3 . . . . .	101
7.2	Contribuições . . . . .	101
7.3	Trabalhos Futuros . . . . .	103
<b>A</b>	<b>Um Mapeamento Sistemático sobre o Ensino de Programação para Pessoas com Deficiência</b>	<b>115</b>
<b>B</b>	<b>A Systematic Literature Review on Teaching Programming to People with Cognitive Disabilities</b>	<b>126</b>
<b>C</b>	<b>Identifying Programming Skills Impacted in Students with Cognitive Disabilities</b>	<b>135</b>
<b>D</b>	<b>Parecer Comitê de Ética</b>	<b>144</b>
<b>E</b>	<b>Termo de Consentimento Livre e Esclarecido</b>	<b>149</b>

---

<b>F</b>	<b>Termo de Consentimento Livre e Esclarecido Responsável Menor</b>	<b>153</b>
<b>G</b>	<b>Perceptions about Teaching Programming in the Neurodiverse Students' Context</b>	<b>157</b>
<b>H</b>	<b>Guia de Elaboração de Material de Programação Acessível para Pessoas Neurodivergentes</b>	<b>167</b>

# Lista de Símbolos

ACM - *Association for Computing Machinery*

CC - *Ciência da Computação*

DI - *Deficiência Intelectual*

EDR - *Educational Design Research*

IDE - *Integrated Development Environment*

IEEE - *Institute of Electrical and Electronic Engineers*

IVC - *Índice de Validade de Conteúdo*

NAI - *Núcleo de Acessibilidade e Inclusão*

NAPNE - *Núcleos de Apoio a Pessoas com Necessidades Específicas*

TDAH - *Transtorno do Déficit de Atenção e Hiperatividade*

TEA - *Transtorno do Espectro Autista*

# Lista de Figuras

2.1	Educational Design Research Adaptado de McKenney e Reeves [59]	20
3.1	Publicações por Origem	27
3.2	Publicações por Ano	28
3.3	Publicações por Tipo	28
4.1	Projeto da Pesquisa segundo a Educational Design Research	33

# Lista de Tabelas

2.1	Um exemplo de exercício de programação . . . . .	13
2.2	Níveis de Autismo pelo DSM-5 [6] . . . . .	16
2.3	Estilo Cognitivo Autista por Stuurman [2][1][79] . . . . .	17
2.4	Características do TDAH por Galeos [33] . . . . .	18
2.5	Algumas Características da Deficiência Intelectual no DSM-5 [6] . . . . .	19
3.1	Sumário dos estudos retornados em cada base . . . . .	26
4.1	Questões de Pesquisa e Objetivos distribuídos nas fases do EDR . . . . .	34
5.1	Deficiência Cognitivas em relação às Habilidades . . . . .	43
5.2	Perfil dos Estudantes . . . . .	49
5.3	Perfil dos Professores . . . . .	50
5.4	Roteiro de Entrevista Professores . . . . .	51
5.5	Roteiro Entrevista Estudantes . . . . .	52
5.6	Dúvidas em Programação . . . . .	53
5.7	Dúvidas em Programação por Tópicos . . . . .	54
5.8	Estratégias do Professor . . . . .	55
6.1	Plano de Aula . . . . .	64
6.2	Roteiro de Aula . . . . .	65
6.3	Perfil dos Juízes Especialistas - Professores de Programação . . . . .	73
6.4	Perfil dos Juízes Especialistas - Profissionais . . . . .	74
6.5	Avaliações dos Juízes Professores de Programação na Primeira Rodada de Delphi . . . . .	76
6.6	Avaliações dos Juízes Profissionais na Primeira Rodada de Delphi . . . . .	77

---

6.7	Considerações Finais dos Juízes Professores Primeira Rodada de Delphi . .	80
6.8	Considerações Finais dos Juízes Professores Primeira Rodada de Delphi . .	81
6.9	Sugestões dadas pelos Juízes Professores de Programação na Primeira Rodada de Delphi . . . . .	83
6.10	Sugestões dadas pelos Juízes Profissionais na Primeira Rodada de Delphi .	84
6.11	Avaliações dos Juízes Professores de Programação na Segunda Rodada de Delphi . . . . .	85
6.12	Avaliações dos Juízes Profissionais na Segunda Rodada de Delphi . . . . .	86
6.13	Considerações Finais dos Juízes Professores JP1 a JP6 da Segunda Rodada de Delphi . . . . .	88
6.14	Considerações Finais dos Juízes Professores JP7 e JP8 da Segunda Rodada de Delphi . . . . .	89
6.15	Considerações Finais dos Juízes Profissionais na Segunda Rodada de Delphi	90
6.16	Sugestões dadas pelos Juízes Professores de Programação na Segunda Rodada de Delphi . . . . .	91
6.17	Avaliações dos Juízes Professores de Programação na Terceira Rodada de Delphi . . . . .	92
6.18	Respostas do Aluno ao Questionário sobre o Roteiro de Aula - Estrutura Condicional . . . . .	96

# Capítulo 1

## Introdução

Este capítulo apresenta a introdução do trabalho. Na Seção 1.1 mostramos qual o contexto em que a pesquisa se enquadra e qual a motivação para desenvolvê-la. Na Seção 1.2, temos uma breve discussão sobre o cenário no qual identificamos o problema. Em seguida, definimos os objetivos e as questões de pesquisa na Seção 1.3. A metodologia utilizada nesta pesquisa é apresentada na Seção 1.4. Na sequência, mostramos as contribuições da pesquisa na Seção 1.5 e as informações referentes ao Comitê de Ética na Seção 1.6. Por fim, a estrutura da tese é indicada na Seção 1.7.

### 1.1 Contexto e Motivação

A área de ensino de introdução à programação compreende o processo de projetar, criar e testar algoritmos. Essa área possui consideráveis índices de evasão [60], mesmo para estudantes sem deficiência. A abordagem utilizada em cursos de programação envolve muitas vezes a exposição de situações-problema onde o estudante deve criar um algoritmo que chegue à solução esperada. Quando o estudante não consegue entender o problema que precisa resolver, dificilmente terá sucesso no exercício [18; 84; 82].

Na minha trajetória como professora de programação, me deparei várias vezes com estudantes com deficiência. Esses estudantes que tinham interesse em aprender, mas tinham necessidades específicas que dificultavam o processo. Cada tipo de deficiência gerava demandas específicas, e, a instituição de ensino não possuía processos bem definidos para amparar a aprendizagem desses alunos, como, por exemplo, a adaptação de aulas, do am-

biente, de instrumentos de avaliação. Partindo desse ponto, vislumbramos que a criação de contextos motivadores para a aprendizagem e métodos de ensino adequados poderiam proporcionar um apoio adequado aos estudantes, particularmente aqueles com algumas dificuldades de aprendizagem, como estudantes com deficiência.

As instituições estão recebendo os alunos com deficiência da melhor forma, mas, na prática, observamos que ainda há muito o que melhorar nesse processo. Nas instituições de ensino que foram alvo desta pesquisa, observamos que existe um setor para apoio a recepção e acompanhamento de alunos com deficiência ao longo do curso. Esses setores contam com uma equipe multidisciplinar que geralmente é composta por pedagogos, psicopedagogos, intérpretes de libras, psicólogo, entre outros. Na Universidade Federal de Campina Grande (UFCG), o setor é o Núcleo de Acessibilidade e Inclusão (NAI)<sup>1</sup> e tem como propósito a “promoção de inclusão e acessibilidade a pessoas com deficiência, Transtornos do Espectro Autista (TEA), altas habilidades/superdotação e transtornos específicos, em toda a UFCG”. No Instituto Federal de Educação, Ciência e Tecnologia da Paraíba o setor específico é a Coordenação de Ações Inclusivas<sup>2</sup>, no nível da reitoria, e os Núcleos de Apoio a Pessoas com Necessidades Específicas (NAPNE) em cada campus. Os NAPNE “atuam no apoio e atendimento de alunos e contam, de acordo com suas necessidades locais, com tradutores e intérpretes de LIBRAS, transcritores e revisores de Braille, cuidadores, letores, alfabetizadores de jovens e adultos e psicopedagogos contratados, além de servidores efetivos do quadro de pessoal do IFPB”.

Existem estudos que buscam tornar o ensino de programação inclusivo [5; 19; 41; 77]. Entretanto, o que vimos em nossas pesquisas prévias é, que a maioria das iniciativas é voltada para as deficiências visíveis, como a deficiência auditiva e a visual [20]. Além destas deficiências, temos várias outras, incluindo aquelas que podem ser chamadas de “não visíveis”, o que pode dificultar mais a forma de condução do ensino para esse perfil de estudante. As deficiências não visíveis, são geralmente aquelas ligadas aos transtornos do neurodesenvolvimento, sendo as mais conhecidas o Autismo e o Transtorno do Déficit de Atenção e Hiperatividade (TDAH). Esses transtornos podem afetar os indivíduos apresen-

<sup>1</sup><https://www.cartadeservicos.ufcg.edu.br/catalogo-de-servicos-ufcg/221-nucleo-de-acessibilidade-e-inclusao-nai-a.html>

<sup>2</sup><https://www.ifpb.edu.br/prae/acoes-inclusivas/apresentacao>

tando, por exemplo, falta de atenção, déficit na comunicação, comportamentos repetitivos, entre outros [6].

No contexto das deficiências “não visíveis” existe uma linha de pensamento que considera os indivíduos com alguma destas como pessoas neurodiversas [75]. Outra forma de se referir a esse público é neuroatípico.

No contexto da neurodiversidade e do ensino de programação, existem poucos estudos e, os que existem são, em sua maioria, voltados para cursos extracurriculares e para crianças [21]. Essas experiências com crianças são válidas, porém esse público-alvo não tem necessariamente as mesmas demandas de estudantes mais velhos, gerando a necessidade de realizar pesquisas com o público adulto e adolescente. No caso dos cursos extracurriculares, podemos ter diversos vieses, como duração variada, material variado, público de idade variável entre outras questões. Diante disso, não temos uma visualização fidedigna das demandas de estudantes desse contexto, em cursos técnicos e de graduação, na área de Computação.

O que identificamos na prática docente, é que grande parte dos estudantes não tem contato com programação ao longo de sua vida escolar, o que pode tornar esse conteúdo mais complexo. Além disso, com os nossos estudos, identificamos que os estudantes com deficiências cognitivas podem ter dificuldades com algumas das competências necessárias para aprender a programar, como raciocínio lógico e abstração, por exemplo [43]. Portanto, ao compreender quais as necessidades dos estudantes neurodiversos, poderemos atender aos desafios desse público-alvo e então estabelecer estratégias que possam apoiar professores e estudantes e favorecer o processo de ensino e aprendizagem de programação.

Os professores de programação enfrentam desafios importantes para ministrar suas aulas, já que em uma sala de aula temos estudantes com diversos perfis de aprendizagem. Um dos desafios mais urgentes diz respeito ao professor atender uma turma que possui estudantes neuro atípicos e neuro típicos. Esse grande desafio pode demandar do professor a busca por capacitação adequada, adequação de práticas pedagógicas e adaptação de seus materiais, como descrito no artigo apresentado no apêndice G. Diante disso, para atender adequadamente todos esses perfis em suas especificidades, é necessário não só fornecer suporte, mas também temos que considerar a demanda de tempo para preparar materiais de apoio inclusivos. Portanto, fornecer instrumentos que podem agilizar esse processo de adaptação de conteúdo é uma estratégia promissora.

## 1.2 Identificação do Problema

É desafiador dar acesso adequado à educação para estudantes com algum tipo de deficiência. Em se tratando de ensino de programação, existe um esforço para trazê-lo para os currículos escolares desde o ensino fundamental. Porém, estas iniciativas não consideram as diferenças entre os estudantes, especialmente aqueles com dificuldades de aprendizagem ou comprometimento cognitivo. Tornar os currículos de informática acessíveis a todos é um grande desafio, além disso, em se tratando dos professores, estes precisam aprender a lidar com estudantes com diferentes necessidades. As instituições buscam novas metodologias, tecnologias e cursos para professores para auxiliar no amparo aos estudantes, com necessidades específicas ou não [87; 88].

Pessoas com deficiência podem cursar qualquer curso técnico ou superior que desejarem, inclusive cursos da área de Computação, segundo a Lei Brasileira da Inclusão de n.º 13.146/2015<sup>3</sup>. Ao iniciar um curso nesta área, eles enfrentarão disciplinas de introdução à programação, que compreendem o aprendizado de projetar, criar e testar algoritmos.

Além de lidar com as dificuldades já inerentes ao ensino de programação, os professores também precisam lidar com a diversidade de estudantes em sala de aula. Particularmente, alguns estudantes apresentam necessidades específicas, como dificuldades de aprendizagem ou déficit de atenção relacionados a distúrbios do neurodesenvolvimento (como o transtorno de déficit de atenção e hiperatividade (TDAH) e estão incluídos no ensino regular. Contudo, o processo educativo não está preparado para recebê-los. A literatura mostra que esse público continua sub-representado e “escondido” na Ciência da Computação, apesar dos esforços [88].

Ainda são poucas as pesquisas sobre programação de práticas de ensino para turmas com diferentes necessidades educacionais. Este desafio a ser enfrentado, envolve o apoio às necessidades específicas de diferentes estudantes em uma sala de aula [81]. Ainda temos algumas iniciativas voltadas para pessoas com diversas habilidades cognitivas e ensino de programação. Conseqüentemente, temos poucos recursos pedagógicos adaptados para esse público, dificultando encontrar boas estratégias que tenham funcionado na prática [50].

Diante deste contexto, o problema que esta tese visou resolver é o de auxiliar profes-

---

<sup>3</sup>[https://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2015/lei/l13146.htm? = undefined](https://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm? = undefined)

sores na prática docente com relação aos estudantes neurodiversos que cursam disciplinas de introdução à programação através da disponibilização de instrumentos pedagógicos para melhorar a qualidade do ensino para esse público alvo.

### **1.3 Objetivos**

No contexto de introdução à programação, que já é historicamente considerado complexo, as especificidades das deficiências cognitivas podem tornar o aprendizado ainda mais complexo quando não há nenhum tipo de adaptação [41; 51].

Diante do problema apresentado na Seção 1.2, esta pesquisa visou disponibilizar instrumentos para auxiliar professores na prática docente com relação aos estudantes neurodiversos nas aulas introdutórias de programação. Para isso, fornecemos aos professores de programação instrumentos pedagógicos adaptados para pessoas neurodivergentes, como planos de aula, roteiros, orientações para construção de planos de aulas, materiais de aula, exercícios e avaliações. Dentro desse público, consideramos pessoas com deficiências cognitivas como Autismo, TDAH, Deficiência Intelectual e Transtorno de Aprendizagem. Esse escopo foi definido por serem as deficiências cognitivas mais apresentadas por estudantes que entraram por meio de cotas de Pessoa com Deficiência, nos cursos da área de Computação das instituições de ensino que tivemos contato nessa pesquisa. Por exemplo, segundo informações fornecidas pelo IFPB, no período 2024.1 dois alunos estão enquadrados nas cotas PCD e ambos possuem laudo de TEA. Já no curso de Engenharia de Computação, foram identificados cinco alunos nas cotas PCD, sendo dois com laudo de TDAH e um com laudo de TDAH e TEA.

Esta tese de doutorado almejou que a proposta de instrumentos pedagógicos adaptados para pessoas neurodivergentes, para aulas de programação, possa favorecer significativamente o aprendizado de estudantes neurodiversos em disciplinas introdutórias de programação. Ao adotar o guia de preparação de aulas, espera-se que os professores consigam adaptar seus roteiros de aula e exercícios de programação de forma ágil, fornecendo aos seus estudantes neurodiversos a possível igualdade de oportunidades de aprendizado nesse contexto.

Para isso, esta pesquisa partiu da premissa de que, fornecendo aos professores instrumentos para adaptação de materiais, em consequência, possibilitando aos estudantes neurodiver-

tos, de disciplinas introdutórias de programação, materiais adaptados as suas necessidades específicas. Assim terão a possibilidade de romper barreiras de acessibilidade, buscando dessa forma o êxito.

Para melhor compreender se é possível criar instrumentos pedagógicos direcionado às necessidades específicas dos estudantes neurodiversos, esta tese de doutorado centrou-se nas seguintes questões específicas de pesquisa (QP):

- **QP1** - Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?

*Esta questão de pesquisa foi respondida através da realização de uma revisão sistemática de literatura [21] e um estudo exploratório [43]. Primeiramente investigamos o estado da arte envolvendo o ensino de programação e pessoas com deficiência cognitiva. Em seguida, visamos investigar como as características específicas das deficiências poderiam impactar nas habilidades necessárias para aprender programação.*

- **QP2** - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da programação introdutória?

*Já para responder a QP2, visamos localizar estudantes neurodiversos em ambiente escolar no contexto da primeira disciplina de programação e seus respectivos professores. Ao localizá-los, realizamos uma pesquisa para identificar as percepções dos estudantes neurodiversos e seus professores em relação a como funcionam as aulas de programação atualmente. Tivemos em vista compreender quais as dificuldades enfrentadas por professores e estudantes neurodiversos nesse contexto. Este estudo foi aceito para publicação no IEEE Frontiers in Education 2024 e será apresentado em outubro de 2024.*

- **QP3** - Como tornar o ensino de programação inclusivo para pessoas neurodivergentes?

*Buscando responder a QP3, elaboramos instrumentos pedagógicos, na forma de um guia, para criação de roteiros de aulas de programação, com foco em estudantes neurodiversos. O guia foi pensado utilizando as principais recomendações relacionadas às demandas desse perfil de estudante, bem como as especificidades de aulas de programação.*

Diante disso, para responder às questões de pesquisa, foram estabelecidos os seguintes objetivos específicos:

- **OBJ1** - Catalogar as habilidades que os estudantes precisam desenvolver para programar e como essas habilidades podem ser influenciadas pelas características dos estudantes com deficiência cognitiva;
- **OBJ2** - Investigar como se dá atualmente o processo de ensino e aprendizagem de programação no contexto de estudantes neurodiversos;
- **OBJ3** - Mapear informações sobre abordagens e práticas gerais recomendadas para o público neurodiverso;
- **OBJ4** - Investigar como os materiais de aula de programação poderiam ser adaptados, associando competências às necessidades específicas do comprometimento cognitivo;
- **OBJ5** - Elaborar instrumentos pedagógicos para criação de aulas de programação adaptado para estudantes neurodiversos;
- **OBJ6** - Avaliar o uso dos instrumentos, na prática.

## 1.4 Metodologia

Esta pesquisa de doutorado se enquadra na área de **Educação em Computação**, e teve em vista abordar um problema complexo que é fornecer instrumentos pedagógicos para auxiliar professores na prática docente com relação à estudantes neurodiversos em disciplinas introdutórias de programação de cursos regulares da área de Computação. Por ser uma pesquisa que almeja contribuir para esse contexto de forma prática, escolhemos um método científico que amparasse esse processo adequadamente pela natureza da pesquisa.

Podemos caracterizar a pesquisa científica de diversas formas, aqui caracterizamos esta pesquisa quanto a natureza, quanto à abordagem e quanto ao método [35].

Quanto à natureza, esta pesquisa está situada na **Pesquisa Aplicada**, já que possui como interesse a aplicação, utilização e consequências práticas do conhecimento gerado [35]. Esta tese propôs instrumentos pedagógicos para criação de aulas de programação adaptado para

estudantes neurodiversos para auxiliar professores na prática docente voltada para estudantes com deficiência cognitiva ou neurodiversos em disciplinas introdutórias de programação, possibilitando a adaptação das suas aulas e aplicação para esse público-alvo.

Quanto a abordagem, esta pesquisa se enquadra na **Pesquisa Qualitativa**, pois buscou conhecimento baseando-se no sentido e significado do fenômeno para as pessoas, e não apenas na frequência em que ele ocorre [37]. Como o público-alvo desta pesquisa é relativamente pequeno em relação ao universo, se faz necessária a obtenção de dados descritivos, de estudantes neurodiversos e de professores de programação, buscando compreender o fenômeno como ocorre atualmente e quais as lacunas existentes que os instrumentos propostos poderiam buscar solucionar.

Quanto ao método e profundidade de objetivos, esta é uma **Pesquisa Exploratória**, pois o fenômeno estudado ainda possui pouco conhecimento acumulado [35; 37]. A pesquisa propôs e validou os instrumentos pedagógicos gerados, buscando proporcionar aos professores um guia prático para auxiliá-los a adequar suas aulas ao público neurodiverso.

Buscando o rigor científico para resolver o problema proposto, escolhemos partir da abordagem Design Science Research (DSR), que foi criada para amparar a criação de artefatos computacionais projetados para a Educação [66]. Indo além da DSR, identificamos que esta abordagem possui uma categoria denominada **Educational Design Research (EDR)**, que tem em vista sistematizar a criação de artefatos educacionais como abordagens educacionais, materiais de aula, estratégias pedagógicas entre outros, para resolver problemas educacionais complexos [24; 58; 59].

## 1.5 Contribuições

Esta tese possui como principal contribuição, a proposta de instrumentos pedagógicos para construção de aulas de programação introdutória com foco em estudantes com deficiência cognitiva ou neurodiversos. Estes instrumentos surgem como uma solução prática para os professores melhorarem a qualidade do ensino de programação, para estudantes neurodiversos. A fim de gerar algo que fosse de fato viável para uso do professor, foi feita uma validação, inserindo professores de programação no processo, além de profissionais de Pedagogia que dão suporte a estudantes com deficiência.

No processo de projeto e construção dos instrumentos, esta pesquisa também gerou outras contribuições secundárias, como a discussão sobre as habilidades necessárias para aprender programação, afetadas pelas deficiências cognitivas [43]. Essa discussão é importante, ao poder embasar novas pesquisas que possibilitarão melhoria de desempenho de estudantes neurodiversos.

Diante disso, esta pesquisa almeja proporcionar aos estudantes neurodiversos, igualdade de oportunidades para aprender programação, principalmente se tratando de programação introdutória, disciplina tão importante para que ele avance em um curso de Computação, seja técnico de nível médio ou de graduação. Desta forma esperamos que o estudante possa obter êxito.

## 1.6 Aspectos Éticos

Esta pesquisa de doutorado seguiu os princípios éticos estabelecidos pelo Conselho Nacional de Saúde do Brasil (CNS), conforme a Resolução n.º 510/2016, que ampara pesquisas com seres humanos sendo aprovada pelo conselho de ética em pesquisa (CEP) da Universidade Federal de Campina Grande (UFCG) sob o número 46773221.9.0000.5182.

## 1.7 Estrutura do Documento

Esta tese está organizada da seguinte forma.

O **Capítulo 2** apresenta a fundamentação teórica. Descrevemos alguns conceitos acerca do ensino de programação, o conjunto de habilidades envolvidas e algumas abordagens existentes para ensinar esse conteúdo. Discutimos também o comprometimento cognitivo, alguns dos principais tipos de deficiência cognitiva e como se chegou à nomenclatura neurodiversidade. Em seguida, apresentamos os trabalhos relacionados a esta pesquisa no **Capítulo 3**, encontrados mediante revisão de literatura.

O **Capítulo 4** especifica a metodologia da pesquisa. Delimitamos a área de investigação e descrevemos e planejamos os passos a seguir nesta investigação de doutoramento para atingir o objetivo.

O **Capítulo 5** apresenta a discussão sobre a influência da neurodiversidade no ensino e

aprendizagem de programação. Para isso respondemos as duas primeiras questões de pesquisa.

O **Capítulo 6** descreve os instrumentos pedagógicos propostos. Apresentamos o guia para criação de aulas de programação inclusivas ao público neurodiverso. Em seguida, mostramos o processo de validação dos instrumentos, seguindo os passos da EDR.

O **Capítulo 7** apresenta as considerações finais da tese. Temos as conclusões e as contribuições geradas pelo trabalho. Também falamos sobre as ameaças a validade da pesquisa e suas limitações. Finalmente discutimos trabalhos futuros que podem ser gerados a partir desta pesquisa.

# Capítulo 2

## Referencial Teórico

Neste capítulo, são abordados conceitos essenciais que amparam esta pesquisa de doutorado. Inicialmente, discute-se o ensino de programação, incluindo estratégias e abordagens empregadas para ensinar esse conteúdo, parte integrante do pensamento computacional. Posteriormente, são apresentadas as características e singularidades dos principais tipos de deficiência cognitiva, também conhecidos como neurodiversidade. Por fim, são introduzidos conceitos essenciais relacionados à condução da pesquisa, a saber, o Educational Design Research e o Método Delphi.

### 2.1 Ensino de Programação

É necessário entender o conceito de Algoritmos para aprender a programar. Um algoritmo é uma sequência de etapas bem definidas que podem receber valores como entrada e gerar uma saída [18]. Podemos criar diferentes soluções para resolver um determinado problema, mas algumas podem ser mais simples ou mais rápidas que outras.

A programação é parte essencial da Ciência da Computação (CC) e não foi tradicionalmente aprendida ao longo da vida escolar dos estudantes brasileiros. Atualmente temos esforços para inserir esse e outros conteúdos da área de Computação na educação Básica. No Brasil, o Ministério da Educação definiu algumas normas para definir o ensino de Computação no contexto da Base Nacional Comum Curricular<sup>1</sup>. No contexto da educação infantil,

---

<sup>1</sup><https://www.gov.br/mec/pt-br/acao-a-informacao/participacao-social/conselhos-e-orgaos-colegiados/cne/base-nacional-comum-curricular-bncc>

por exemplo, é definido que haja a criação e testes de algoritmos de forma lúdica, utilizando brincadeiras e objetos do ambiente, bem como a movimentação do próprio corpo. Já no ensino fundamental, a computação passa a ser explorada como um agente de transformação e, no ensino médio, existe o foco em desenvolvimento de projetos, articulando conceitos, procedimentos e linguagens próprias da computação preferencialmente de maneira colaborativa.

A Sociedade Brasileira de Computação também elaborou diretrizes para o ensino de Computação na educação básica. Nos anos iniciais do ensino fundamental, é esperado que o estudante compreenda como criar algoritmos de forma básica, utilizando, por exemplo, operações aritméticas simples. Já nos anos finais ele deve conseguir criar algoritmos mais complexos, dominando as principais técnicas para construir soluções algorítmicas. No ensino médio, o foco é no desenvolvimento de projetos aplicando o que foi aprendido no ensino fundamental, inserindo a análise mais aprofundada de algoritmos no tocante a correção e eficiência [22].

Os professores de programação buscam constantemente estratégias e abordagens para gerar melhor desempenho dos estudantes. Em geral, as dificuldades enfrentadas estão relacionadas à compreensão de conceitos que não são claros na teoria, mas tornam-se através da prática [84].

Alguns métodos ou abordagens para o ensino de programação preocupam-se com como os professores transmitem conhecimento ao estudante. Uma das abordagens mais difundidas de ensinar programação introdutória é a abordagem de Thompson [82]. Esta abordagem trata de ensinar Algoritmos através da resolução de problemas. Diante disso, apresenta os seguintes passos para resolver um exercício de programação: (1) compreender o problema, (2) planejar a solução, (3) implementar o programa e (4) revisar. Essas etapas podem ser fáceis e naturais para os estudantes resolverem seus problemas de programação. Porém, todo o processo pode ser desafiador para alguns estudantes iniciantes, aumentando a necessidade de orientação e assistência.

A Tabela 2.1 <sup>2</sup> mostra um exercício de programação típico no qual a instrução é dividida em as seguintes partes: (1) Descrição: onde o estudante encontra um texto narrativo explicando, de forma geral, o problema a ser resolvido com programação; (2) Entrada: conjunto

---

<sup>2</sup><https://www.thehuxley.com/problem/2931>

de informações que o programa deve receber como entrada; (3) Saída: informação que o programa deve retornar ao dispositivo de saída padrão; (4) Exemplos de entradas e saídas: forma de auxiliar o estudante a encontrar uma solução para o problema proposto mostrando alguns exemplos de entradas e saídas correspondentes a uma solução compatível com o problema.

Tabela 2.1: Um exemplo de exercício de programação

Descrição
<p>No Brasil, muitas crianças aguardam a chance de ter uma família. Para efetivar uma adoção, alguns critérios precisam ser cumpridos segundo o Estatuto da Criança e do Adolescente, são eles:</p> <p>Art. 42. Podem adotar os maiores de 18 (dezoito) anos, independentemente do estado civil.</p> <p>§1º Não podem adotar os ascendentes e os irmãos do adotando.</p> <p>§2º. Para adoção conjunta, é indispensável que os adotantes sejam casados civilmente ou mantenham união estável, comprovada a estabilidade da família.</p> <p>§3º. O adotante há de ser, pelo menos, dezesseis anos mais velho do que o adotando.</p> <p>Art. 45. A adoção depende do consentimento dos pais ou do representante legal do adotando.</p> <p>§1º. O consentimento será dispensado em relação à criança ou adolescente cujos pais sejam desconhecidos ou tenham sido destituídos do poder familiar.</p> <p>§2º. Em se tratando de adotando maior de doze anos, será também necessário o seu consentimento.</p>
Entrada
<p>A Idade do Adotante</p> <p>Se o adotante é irmão ou ascendente</p> <p>Se é Adoção Conjunta</p> <p>Adotantes são casados ou união estável</p> <p>Idade do Adotando</p> <p>Pais Desconhecidos ou Adotando Destituído do Poder Familiar?</p> <p>Consentimento dos pais quando não desconhecidos</p> <p>Consentimento do adotando maior de doze anos</p>
Saída

Se pode ou não adotar	
Exemplos de Execução	
Entrada	Saída
45	Pode adotar
N	
N	
N	
8	
N	
S	
N	

Verificando um exercício como esse, quais são as características encontradas que podem gerar dificuldades aos estudantes com deficiência? Por exemplo, uma descrição extensa do exercício pode distrair um estudante com Transtorno de Deficit de Atenção e Hiperatividade (TDAH), dificultando a busca de uma solução para o problema. Estas e outras questões motivam a pesquisa proposta neste documento.

## 2.2 Deficiências Cognitivas e Neurodiversidade

Os Transtornos do Neurodesenvolvimento são condições que podem acometer as crianças desde a sua formação na gestação. Geralmente os sintomas já surgem desde a infância, podendo apresentar deficit em áreas como habilidades sociais e aprendizagem. Esses prejuízos podem variar de pessoa para pessoa, havendo casos em que os deficit são sutis e em outros podem ter prejuízos globais em habilidades sociais ou inteligência, mas não limitados a estes. Existem vários tipos de transtornos de Neurodesenvolvimento catalogados, e é bastante comum que uma pessoa possua mais de um transtorno combinadamente. O diagnóstico é feito mediante avaliações clínicas com profissionais capacitados, como neurologistas, psiquiatras e neuropsicólogos [6]. Detalhamos aqui os mais comuns, os quais são o Transtorno

do Espectro Autista (TEA), Transtorno do Deficit de Atenção e Hiperatividade (TDAH) e Transtornos de Aprendizagem.

Deficiência cognitiva é um termo genérico na literatura para discutir deficiências mentais. As deficiências cognitivas mais comuns incluem Transtornos do Neurodesenvolvimento, como TEA, TDAH e Transtornos Específicos de Aprendizagem, como Dislexia [55]. Koushik e Kane [50] declaram que poucas pesquisas ainda se concentram no ensino de informática para pessoas com diversas habilidades cognitivas. Também há poucos relatos de estratégias para esse público, o que nos mostra uma lacuna considerável a ser explorada.

Por estes serem transtornos que atingem o cérebro, existe uma teoria tratada pela primeira vez pela psicóloga Judy Singer [75], que considera as características de transtornos do neurodesenvolvimento como variações naturais do cérebro humano, não considerando isso como doenças, mas sim como diversidade, daí o termo neurodiversidade.

### **2.2.1 Autismo**

O Transtorno do Espectro Autista (TEA) faz parte do conjunto de transtornos do neurodesenvolvimento e pode afetar os indivíduos principalmente na comunicação e nos comportamentos repetitivos [6]. É um distúrbio com três níveis diferentes de gravidade, como podemos ver no resumo da Tabela 2.2. O nível mais grave é o nível 3, e o mais leve, o nível 1. De acordo com cada nível, o autista pode apresentar maior ou menor resistência, principalmente nas interações sociais. Outra característica importante é a necessidade de manter uma rotina bem definida, tornando as mudanças bruscas dolorosas para o autista.

Tabela 2.2: Níveis de Autismo pelo DSM-5 [6]

Nível	Comunicação Social	Comportamentos Restritos e Repetitivos
Nível 1	Limitação em iniciar interações sociais	Inflexibilidade, dificuldades em lidar com mudanças e problemas organizacionais
Nível 2	Limitação moderada nas interações sociais	Inflexibilidade e resistência considerável à mudança
Nível 3	Limitação severa para interações sociais	Inflexibilidade e resistência severa a mudança

O autismo pode se manifestar de diferentes maneiras e não prejudicará permanentemente a inteligência dos indivíduos. Pesquisas indicam que aproximadamente metade dos autistas têm inteligência média ou acima da média [25]. Com o passar dos anos, temos cada vez mais diagnósticos de autismo, o que implica que cada vez mais pessoas autistas ingressem no ambiente escolar, inclusive no ensino superior [34].

Stuurman et al. [79] apresentou um conjunto de características que ele chamou de “estilo cognitivo autista”, como podemos ver na Tabela 2.3. Essas características mostram, por exemplo, que pessoas autistas são muito detalhistas, o que pode ser uma boa habilidade para testes de software. Além disso, eles precisam de regras para categorizar as coisas em grupos. Pessoas autistas apresentam dificuldades em interpretar informações ambíguas, tornando-as mais adeptas a linguagens que não permitem dupla interpretação, como lógica e matemática. Assim, os autistas costumam tomar decisões mais racionais do que os neurotípicos, tendo dificuldade de generalização e dedução. Por outro lado, são indivíduos sistemáticos, tomando decisões indutivamente. Além dessas características, os autistas podem apresentar deficiências em funções executivas, como planejamento e controle de impulsos.

Tabela 2.3: Estilo Cognitivo Autista por Stuurman [79]

---

Características
Detalhistas
Categorização
Dificuldade com ambiguidade
Raciocínio racional
Generalização fraca
Sistematização
Dificuldade em funções executivas

---

### 2.2.2 Transtorno de Deficit de Atenção e Hiperatividade

O TDAH é outro transtorno do neurodesenvolvimento caracterizado por desregulação na forma de desatenção, desorganização, hiperatividade e impulsividade. Uma pessoa com TDAH pode ter apenas desatenção, hiperatividade ou ambos como disfunção central. Esse transtorno geralmente aparece na infância e continua na idade adulta, o que pode causar prejuízos no funcionamento social, acadêmico e profissional [6].

Os estudantes com TDAH são mais propensos a ter dificuldades de aprendizagem do que os neurotípicos e déficit em leitura e matemática. Essa comorbidade comum pode atingir até 45% das pessoas com esse transtorno [62].

O TDAH pode se manifestar de três maneiras [6; 33]: (1) apenas hiperatividade, (2) apenas deficit de atenção e (3) hiperatividade e desatenção combinadas. Na Tabela 2.4 vemos algumas características de cada tipo.

Tabela 2.4: Características do TDAH por Galeos [33]

Tipo	Características
Hiperativo	Energia excessiva, impaciência, dificuldade em esperar, fazer coisas sem pensar, dificuldade em corrigir erros
Desatento	Falta de concentração, dificuldade em memorização, dificuldade em seguir ordens, desorganização
Combinado	Possui características tanto do tipo hiperativo quanto do tipo desatento

As características não se limitam ao que está descrito na Tabela 2.4, mas são essenciais para compreender como as pessoas com TDAH reagem ao que está acontecendo ao seu redor. Para pensar em como ensinar estudantes com TDAH, devemos lembrar que é mais difícil para eles manterem o foco em um assunto que não lhes interessa muito.

### 2.2.3 Transtornos de Aprendizagem

Os distúrbios de aprendizagem podem ser apresentados por diversas dificuldades apresentadas ao longo da vida escolar, como a dislexia, que apresenta características de comprometimento no reconhecimento e ortografia de palavras. Além destes, quem possui esse transtorno pode ter problemas com leitura e raciocínio matemático [6].

A dislexia é uma deficiência que dificulta no processamento de informações. Além disso, pessoas com dislexia podem ter problemas para organizar e realizar tarefas, assimilar conteúdos e escrever textos [78].

Embora as dificuldades de aprendizagem possam resultar de distúrbios de aprendizagem específicos, como a dislexia, ou outros distúrbios, como o TDAH [87], o distúrbio de aprendizagem específico pode afetar pessoas sem outro distúrbio. Em alguns casos, podem até ter altos níveis de inteligência, mas apresentam dificuldades acadêmicas devido a deficit psicológico.

### 2.2.4 Deficiência Intelectual

Deficiência Intelectual, ou Transtorno do Desenvolvimento Intelectual, é um tipo de deficiência que pode dificultar nas capacidades mentais genéricas, como, por exemplo, raciocínio lógico, planejamento e abstração. Pode se apresentar em diferentes níveis de deficit (leve, moderada, grave ou profunda) e dificultar na independência do indivíduo [6]. Na Tabela 2.5, temos algumas características de deficiência intelectual, dentro de cada nível de gravidade.

Tabela 2.5: Algumas Características da Deficiência Intelectual no DSM-5 [6]

Tipo	Características
Leve	Dificuldades em aprender habilidades acadêmicas, como leitura e escrita
Moderada	Dificuldades nas habilidades acadêmicas, necessitando de apoio
Grave	Alcance limitado de habilidades conceituais, com pouca compreensão da escrita e matemática
Profunda	Compreensão muito limitada na comunicação, geralmente possui prejuízos sensoriais e físicos

## 2.3 Educational Design Research

A Pesquisa em Design Educacional (EDR) é um gênero de pesquisa baseado na investigação científica e no esforço prático para resolver problemas educacionais [58]. Com EDR podemos criar qualquer artefato acadêmico, como produtos, processos e metodologias. O processo de construção de pesquisa em EDR pode ser flexível, colaborativo e inclusivo, mas também pode ser teórico e transformador e tem tido excelentes resultados ao longo dos anos. Para utilizar a EDR, devemos entender que se trata de um gênero iterativo, onde o pesquisador pode criar seus produtos educacionais a partir de fases de investigação, avaliação e divulgação, podendo refazê-los quantas vezes forem necessárias até seu resultado.

Podemos usar a EDR para criar intervenções educacionais via design, desenvolvimento e avaliação apoiados pela teoria e prática [67]. Essas etapas sistemáticas são cíclicas. Faremos quantas iterações forem necessárias para atingir nosso objetivo.

Uma abordagem para EDR é proposta por McKenney e Reeves [59], com foco nas etapas de pesquisa e desenvolvimento e como elas podem ser interligadas para gerar resultados. Vemos na Figura 2.1 a proposta genérica, o que nos permite fazer adaptações para cada pesquisa.

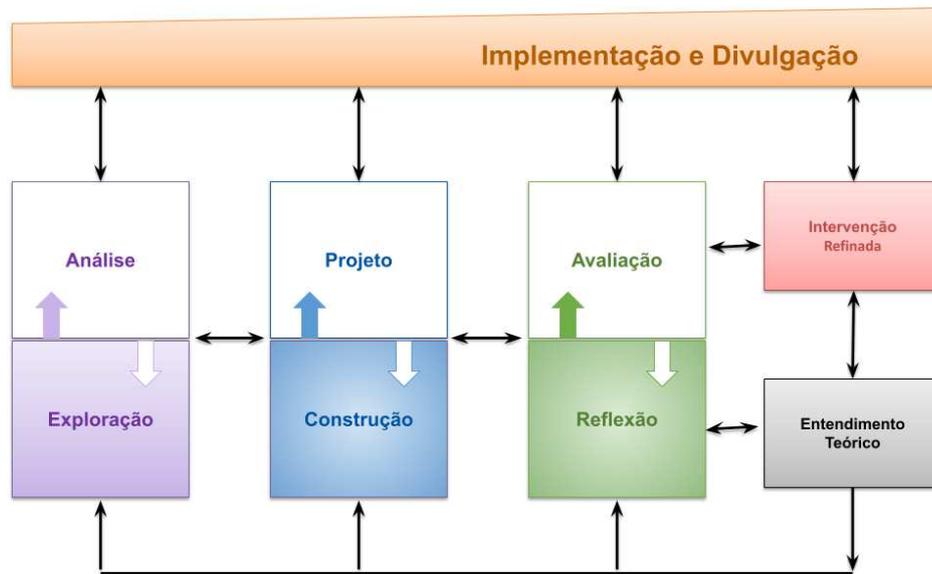


Figura 2.1: Educational Design Research Adaptado de McKenney e Reeves [59]

Como podemos observar, o modelo proposto apresenta três fases interligadas com setas que indicam como todo o processo é interativo e flexível. Na fase de **análise e exploração**, devemos compreender e contextualizar o problema a ser resolvido. Nesta fase, costumamos realizar pesquisas bibliográficas e coleta de dados, que podem nos ajudar a entender o problema e propor uma solução. Na fase de **projeto e construção**, começamos a ter ideias de como resolver o problema com base nas teorias obtidas na etapa anterior, criando possíveis soluções que refinam. Na fase de **avaliação e reflexão**, temos algum produto de pesquisa que precisa ser testado e refinado quantas vezes for necessário para chegar a uma solução estável.

É fundamental salientar que podemos ter a união entre teoria e prática para criar um artefato educacional inovador em todas as etapas.

Após sucessivas iterações entre as três fases, teremos os resultados da investigação, os

quais são parte práticos e parte teóricos. A parte prática seria o produto educacional, que pode ser um processo, um programa ou uma política. A parte de compreensão teórica seriam resultados que descrevem ou explicam um fenômeno educacional, que podem ser, por exemplo, artigos científicos gerados por pesquisas.

## 2.4 O Método Delphi

A coleta de dados é uma parte importante da pesquisa científica, e pode ser feita de variadas maneiras. Um método interessante de ser aplicado em pesquisas em educação é o Delphi. O método ou técnica Delphi se baseia na ideia de consenso para um planejamento estratégico [3].

Esse método iniciou-se nos Estados Unidos, na década de 1950, e tinha o objetivo de gerar consenso de especialistas sobre a área militar para defesa nacional. Foi visto posteriormente, que Delphi poderia ser utilizado em diversas áreas, como tecnologia e educação [40].

A aplicação do Delphi acontece de forma que, para se resolver um problema complexo, convocam-se especialistas na área e busca-se gerar um consenso sobre o tema. Os especialistas não precisam estar geograficamente no mesmo lugar, facilitando o acesso [3; 40]. Para isso, deve-se seguir alguns princípios, a saber [4]:

- Escolher adequadamente os especialistas, de forma que estes tenham domínio relevante do conhecimento que queremos pesquisar;
- Escolher especialistas heterogêneos, de forma que o conhecimento agregado deles se complemente;
- Recrutar entre cinco e vinte especialistas é interessante para garantir bons resultados, sem comprometer a qualidade da análise com um número muito baixo de especialistas, ou torná-la muito complexa com um número muito alto de especialistas.
- É importante dar o feedback entre as rodadas, divulgando o consenso e as respostas dos especialistas, já que eles não conversam entre si.
- Geralmente são suficientes três rodadas para as respostas terem estabilidade.

Para a implementação do Delphi, primeiramente precisamos **escolher os especialistas** que participarão da pesquisa. Selecionar bem quem irá participar do estudo é uma parte fundamental para o sucesso. Precisa haver o comprometimento dos escolhidos, para obtermos dados de qualidade. É normal e esperado que nem todos os recrutados no início perdurem até a última rodada, então, ao se recrutar deve-se pensar em uma quantidade maior do que a que se espera de fato [3; 40; 57].

Além da escolha dos especialistas, outra parte importante do estudo é **elaborar os questionários**, que serão adaptados a cada nova rodada de respostas. Deve-se ter em mente o que queremos como contribuição dos especialistas. Uma estratégia bastante utilizada é inserir perguntas com escalas de importância, como a escala de Likert. Além dos dados de escala, é importante que os juízes tenham questões abertas, para haver espaço para as observações. Essas respostas irão ser cruciais para o processo qualitativo e a interação entre os juízes a cada rodada [3; 40; 57].

As **rodadas** são criadas a partir do que se deseja validar e a quantidade dependerá das respostas dos especialistas. A partir da primeira rodada, as rodadas seguintes devem considerar as respostas da rodada anterior para os ajustes e é importante que se divulguem os resultados da rodada anterior para os avaliadores saberem o que foi respondido. A tendência é que ao longo das rodadas as respostas tendem a convergir para um consenso. Não existe um número exato de rodadas, o que é recomendado é que se parem as rodadas quando as respostas atinjam estabilidade, o que não necessariamente significa um consenso completo [4].

A partir das rodadas, é necessário realizar a **análise dos dados**, sendo feita entre uma rodada e a próxima, para fazer com que os participantes sobre a evolução de todo o processo. Esta análise deve ser não apenas quantitativa, mas também qualitativa [3; 40; 57].

## 2.5 Considerações Finais do Capítulo

Neste capítulo, visamos fornecer o arcabouço teórico para fundamentar a compreensão da pesquisa apresentada nesta tese. A Seção 2.1 abordou os principais conceitos e abordagens envolvidas na programação introdutória. Já a Seção 2.2 apresentou as principais deficiências cognitivas e a neurodiversidade, mostrando suas principais características e demandas.

Na Seção 2.3 apresentamos o Educational Design Research, método utilizado na condução desta pesquisa. Por fim, na Seção 2.4 apresentamos o Delphi, método utilizado para validação dos instrumentos pedagógicos propostos nesta tese.

# Capítulo 3

## Trabalhos Relacionados

A principal motivação para esta pesquisa foi buscar formas de incluir pessoas com deficiência no ensino de programação. Porém, sabemos que cada tipo de deficiência possui suas especificidades e demandas. Diante disso, não conseguimos gerar soluções que atendam todas elas de forma geral. Portanto, inicialmente tivemos em vista delimitar nosso escopo de pesquisa no tocante ao tipo de deficiência que seria atingido.

Realizamos e publicamos um mapeamento sistemático feito para identificar o estado da arte sobre o ensino de programação para pessoas com deficiência [20], que está em sua íntegra no apêndice A. Nesse estudo, vislumbramos uma maior lacuna no contexto das deficiências cognitivas. Diante disso, delimitamos nossa pesquisa para esse público-alvo.

Delimitando o escopo para deficiências cognitivas, realizamos e publicamos uma revisão sistemática de literatura [21], para conhecermos os estudos que estavam sendo feitos relacionando ensino de programação e pessoas com deficiência cognitiva. Esse estudo encontra-se em sua totalidade no apêndice B. Posteriormente tivemos em vista atualizar nossa revisão, realizando uma nova seleção de trabalhos. Assim, este capítulo apresenta os resultados da revisão atualizada até 2024. Esse estudo faz parte da fase 1 da EDR, de Análise e Exploração. Lembrando que, como o método é iterativo, podemos passar mais de uma vez pelas fases. Portanto, discutimos os resultados dos principais trabalhos relacionados a esta temática, na Seção 3.1. Por fim, expomos as conclusões deste capítulo, na Seção 3.2.

## 3.1 Revisão Sistemática de Literatura

Para condução da Revisão Sistemática de Literatura, adotamos o protocolo definido por Kitchenham e Charters [47], que possui três fases: (1) planejamento, (2) condução e (3) divulgação. Diante disso, esta revisão partiu dos resultados da revisão realizada em 2021 [21] que foi atualizada até 2024, e aspirou buscar estudos primários sobre ensino de programação e pessoas neurodiversas ou com deficiência cognitiva.

### 3.1.1 Protocolo e Planejamento

O protocolo RSL define que precisamos de uma questão a ser respondida, e a questão desta revisão é identificar “quais são os esforços e contribuições existentes para ensinar pessoas com deficiência cognitiva a programar”?

### 3.1.2 Condução

Para condução da RSL, utilizamos a seguinte chave, além da revisão feita em 2021 [21].

**(“comput\*” OR “cs” OR “program\*”) AND (“teac\*” OR “learn\*” OR “student\*”) AND (“autism” OR “cognitive disabilities” OR “neurodivers\*” OR “ADHD” OR “Attention Deficit”)**

Os critérios de inclusão definidos para a seleção foram:

- (CI1) - Artigos em inglês;
- (CI2) - Artigos publicados nos últimos 5 anos.
- (CI3) - Os artigos deverão conter as palavras-chave no título e/ou no resumo e/ou no palavras-chave do artigo selecionado;
- (CI4) - Artigos completos que apresentem metodologias, sugestões ou práticas para ensinar programação a pessoas com algum tipo de deficiência cognitiva.

E a exclusão:

- (CE1) - Trabalhos repetidos;

- (CE2) - Artigos de revisão ou mapeamento de literatura.

A busca foi realizada nas bases do Institute of Electrical and Electronic Engineers (IEEE), da Association for Computing Machinery (ACM) e da Springer, aplicando-se as alterações necessárias para a chave em cada base.

Tabela 3.1: Sumário dos estudos retornados em cada base

<b>Base</b>	<b>Resultados</b>
ACM	2030
IEEE	235
Springer	578
<b>Total</b>	<b>2843</b>

### 3.1.3 Triagem

O processo de triagem dos artigos foi feito aplicando-se os critérios de inclusão e exclusão, bem como realizando a leitura dos títulos, resumos e palavras-chave de cada artigo.

Primeiramente organizamos os artigos encontrados nas bases pelos critérios de inclusão CI1, CI2 e CI3. Essa triagem reduziu a lista de trabalhos a serem analisados. A partir desse ponto iniciamos a leitura dos resumos, aplicando o critério CI4 e os critérios de exclusão CE1 e CE2.

Diante disso, obtivemos um total de 21 trabalhos que tem relação com nosso tema de pesquisa e analisamos os achados a seguir.

### 3.1.4 Resultados

A RSL foi realizada e publicada em 2021, sendo revista e atualizada entre os meses de abril e julho de 2024. Nesta seção, apresentamos e discutimos os resultados encontrados nesta revisão atualizada.

## Resultados Gerais

Quanto a disposição geográfica dos trabalhos, podemos ver na Figura 3.1 que a maior prevalência de trabalhos é oriunda dos Estados Unidos, com doze trabalhos. Esta quantidade expressiva destoa do restante da distribuição, mostrando uma hegemonia de trabalhos na área vinda desta origem. Esse resultado é esperado, já que nos Estados Unidos existem diversas iniciativas voltadas para a equidade.

Um fato interessante é que o segundo país com mais trabalhos é o Brasil. Isso mostra que o país está emergindo nas iniciativas voltadas para a inclusão.

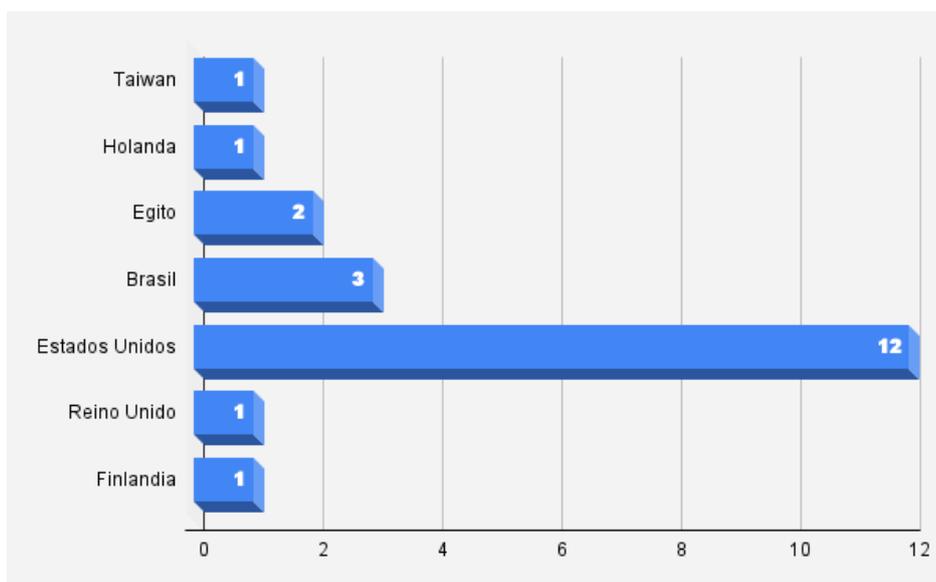


Figura 3.1: Publicações por Origem

Sobre a disposição de trabalhos anuais, vemos na Figura 3.2 que, os anos de 2020 e 2022, foram os que tiveram maior quantidade de publicações. Fato interessante, já que enfrentamos uma pandemia de COVID-19, fechamento de instituições de ensino e adaptação de aulas para o contexto remoto.

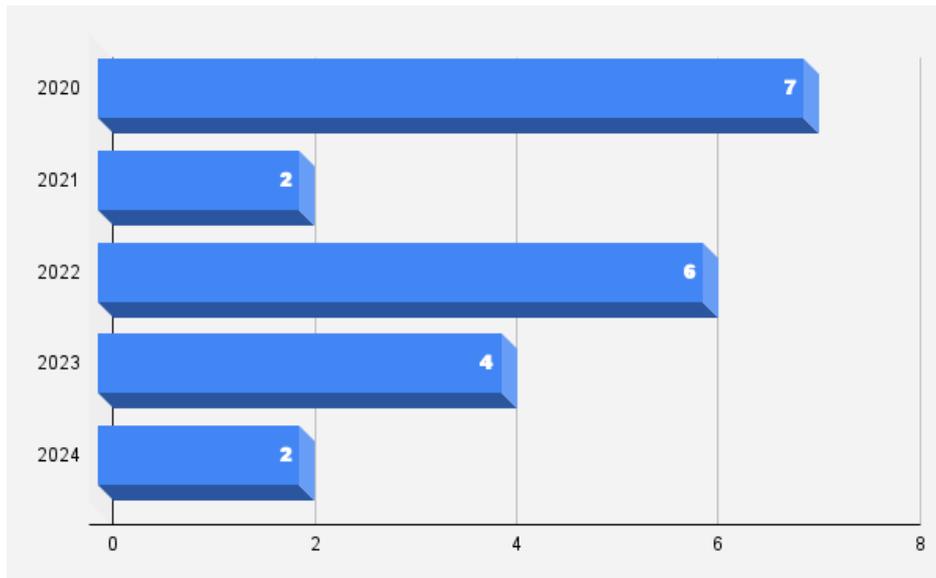


Figura 3.2: Publicações por Ano

Os trabalhos encontrados mostram os esforços que estão sendo feitos ao redor do mundo para promover a inclusão de pessoas neurodiversas no contexto do ensino de programação, como vemos na Figura 3.3. Identificamos diversos tipos de trabalhos, destacando-se Estudos de Caso e Relatos de Experiência. O que podemos observar é, que, em se tratando de um tema emergente com muitas lacunas a serem preenchidas, as iniciativas são motivadas pela demanda, enquanto os resultados são obtidos através da experimentação.

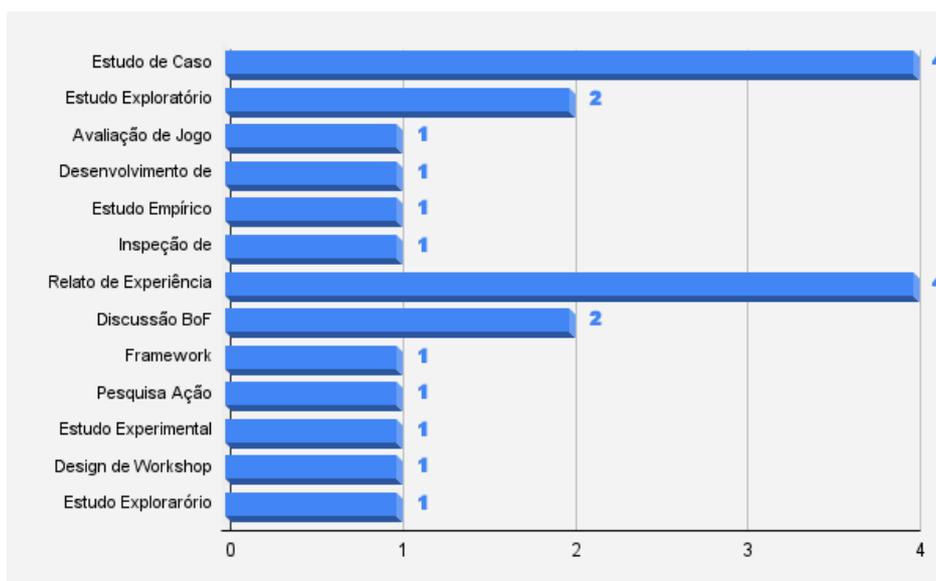


Figura 3.3: Publicações por Tipo

### **Ensino de Programação e Neurodiversidade**

Ao longo da busca por trabalhos relevantes e em consonância com esta pesquisa, observamos que os esforços atuais contam com uma parte considerável dedicada ao público-alvo infantil, seguindo a linha do contato precoce com a programação, e da possibilidade deste contato ser também benéfico para crianças neurodiversas [5; 23; 26; 27; 52; 73].

Ainda são poucos os trabalhos que focam em estudantes de graduação, o que mostra uma lacuna considerável a ser explorada [10; 11]. No estudo apresentado em [10], foi analisada a experiência de aulas de Programação com três estudantes com autismo, cada um cursando uma disciplina diferente no momento da pesquisa: Programação Introdutória, Programação Orientada a Objetos e Banco de Dados. Essa diversidade de disciplinas pode ser um viés nos resultados. Em cada caso, foi aplicada uma metodologia diferente, adaptada à respectiva disciplina. Já [11] apresenta um BoF que trata de quais suportes adicionais instrutores de CC deveriam ter para atender estudantes neurodiversos. O aspecto comum em [10] e [11] foi a ratificação da necessidade de os professores estarem preparados para lidar com estudantes autistas, pois as abordagens tradicionais podem não ser eficazes para esse perfil de estudante.

Ensinar programação a adultos fora de um contexto acadêmico também ainda é pouco explorado [41; 49]. Em [41], explora-se um curso extracurricular de programação aplicado a adultos com autismo, utilizando Problemas de Parsons. Essa estratégia consiste em apresentar aos estudantes um conjunto de linhas de código embaralhadas para que eles reorganizem corretamente para resolver um problema. Já [49], trata de tornar programação em blocos, acessível a pessoas com deficiência cognitiva, projetando técnicas de interação acessíveis.

Estudantes de Ensino Médio foram contemplados em alguns estudos [19; 63; 86]. Os trabalhos apresentam estratégias realizadas com cursos extracurriculares. Apesar de ter sido aprovado na Base Nacional Comum Curricular no Brasil um currículo para Ciência da Computação na educação básica, ainda não é implementado oficialmente por todas as escolas. Um nicho de pesquisa interessante que pode ser explorado é o de estudantes dos cursos técnicos de nível médio da área de Informática. Nos Institutos Federais, por exemplo, existem cursos de nível médio onde os estudantes possuem disciplinas de Programação.

Em [19] foi aplicado um curso extra para um estudante com TDAH, trabalhando conceitos de Pensamento Computacional com 34 semanas de duração. Após a aplicação do curso, foi feita uma avaliação para verificar a taxa de sucesso do estudante, que foi definida em

80% de acertos. O estudante obteve uma taxa de sucesso de 84%, sugerindo a eficácia do experimento. Embora relevante, a experiência retrata um curso extracurricular e para apenas um estudante. Já em [63], temos a aplicação de um curso voltado para desenvolvimento de jogos e que exercitou também o trabalho em equipe. Esse curso chamado de acampamento, foi realizado com 14 estudantes autistas. Nesse curso, os pesquisadores geraram algumas recomendações com base na experiência deles, como, por exemplo, evitar distrações, trabalhar o comportamento entre o grupo e favorecer o trabalho em equipe. São boas estratégias, mas estão focadas na parte operacional do curso em si e não nos materiais aos quais os estudantes têm acesso. Experiência parecida foi relatada em [86], que também trabalhou a ideia de acampamento para ensinar programação, mas de forma totalmente remota. O público-alvo tem diferentes tipos de deficiência como autismo, deficiência intelectual, deficiência visual e cegueira. Dessa forma vemos que são deficiências que tem demandas bem específicas, o que pode atrapalhar os resultados.

O uso de ferramentas visuais para ensinar programação é bastante difundido na comunidade acadêmica. É interessante pensar em formas de avaliar se essas ferramentas atendem ao público neurodiverso, ou quais seriam as adaptações necessárias [73]. Além disso, um programador tem a sua disposição diversas ferramentas de desenvolvimento integrado, então seria interessante saber se estas ferramentas são acessíveis a esse público.

O uso de gamificação também é uma abordagem considerada por uma quantidade significativa de trabalhos. A aplicação de jogos é considerada benéfica pela sua natureza desafiadora, estimulando o desenvolvimento de habilidades como resolução de problemas e concentração. Além disso, existem os benefícios sociais como a aprendizagem de regras e trabalho em equipe [26; 27; 52; 63].

A Robótica é uma vertente interessante, que vem ganhando espaço nas aulas de programação, o que pode ser uma boa estratégia para motivar os estudantes [52; 23; 68]. As iniciativas encontradas que usam essa estratégia focaram em crianças. Em [23], a aplicação de atividades de Robótica Educacional com crianças autistas apontou impactos positivos para concentração, interação social e cumprimento de regras. Nesse caso, o trabalho utiliza as aulas de programação como motivador para o desenvolvimento de outras habilidades, o que é muito interessante, mas aprender programação não seria o foco principal.

Outra discussão pertinente foi encontrada em [79], onde temos um estudo que cruza as características cognitivas de pessoas com TEA com algumas características desejáveis para a área de Engenharia de Software, incluindo Pensamento Computacional, Acadêmico e de Engenharia. Nesse estudo foram gerados alguns guidelines que poderiam ajudar pessoas autistas nesta área, como, por exemplo, ser claro no contexto do texto, dar instruções claras e específicas, ter sempre exercícios associados. As recomendações são relevantes, mas precisam ser mais aprofundadas no tocante a serem aplicadas por professores, por exemplo. Um trabalho de nossa autoria [43], busca fazer um estudo aprofundando a discussão ao nível das habilidades específicas para aprender a programar e os impactos que as deficiências cognitivas podem gerar nelas.

Nem todos os trabalhos investigaram e propuseram o conteúdo e materiais apropriados para pessoas com deficiência. Em [86], houve algumas adaptações nas atividades, incluindo a redução de enunciado, identificadores mais simples, instruções mais curtas.

## **3.2 Considerações Finais do Capítulo**

Este capítulo apresentou o estado da arte relacionado a esta pesquisa de doutorado. Realizamos uma revisão de literatura, onde encontramos trabalhos relevantes que tem relação com o tema e que podem contribuir com a discussão aqui realizada.

Observamos que boa parte dos estudos atuais focam no uso da programação para auxiliar crianças neurodiversas na aquisição de habilidades importantes, como atenção e socialização. Vimos também que a inclusão de pessoas neurodiversas na graduação em cursos de Computação ainda é pouco explorada, abrindo um grande campo de pesquisa a ser atingido.

O que encontramos sobre experiências em ensinar de fato programação nos traz de aprendizado que existe uma demanda muito clara por iniciativas apropriadas para esse público-alvo, porém ainda existe uma grande lacuna de como isto pode ser feito.

# Capítulo 4

## Metodologia

Conforme apresentado na Seção 1.3, esta pesquisa teve como objetivo principal favorecer a inclusão de estudantes neurodiversos nas aulas introdutórias de programação. Para isso, fornecemos aos professores de programação instrumentos pedagógicos adaptados para o público neurodiverso. O método empregado nesta pesquisa foi o gênero Educational Design Research (EDR). Este gênero engloba abordagens para gerar produtos que podem impactar no ensino e aprendizagem para contextos bem definidos [7].

### 4.1 Metodologia Aplicada

EDR é derivada do método emergente Design Science Research, que visa dar rigor científico a criação de artefatos para resolver problemas contextualizados [66]. A EDR foi escolhida por ser focado na resolução de problemas através da criação de artefatos que podem ser educacionais [58]. Na EDR, pode-se chamar de intervenção um produto que tem em vista resolver um problema educacional.

Esta tese apresenta um processo iterativo de análise, projeto, desenvolvimento e validação de instrumentos pedagógicos para aulas de programação adaptados ao público neurodiverso. Diante desse contexto, esta tese se utiliza de uma metodologia apropriada para geração de projetos educacionais. Dessa forma esta pesquisa se baseia no modelo de projeto definido na EDR, sendo adaptado a este contexto.

Na EDR, o processo de desenvolvimento do produto é cíclico e iterativo, portanto, podemos ter diversas iterações de investigação, desenvolvimento, testes e refinamento [58]. Além

do produto gerado, a compreensão teórica do fenômeno em questão também é considerada resultado da EDR, pois esse resultado pode ser útil para além do contexto da pesquisa. Na Figura 4.1, podemos ver o projeto desta pesquisa baseado em EDR.

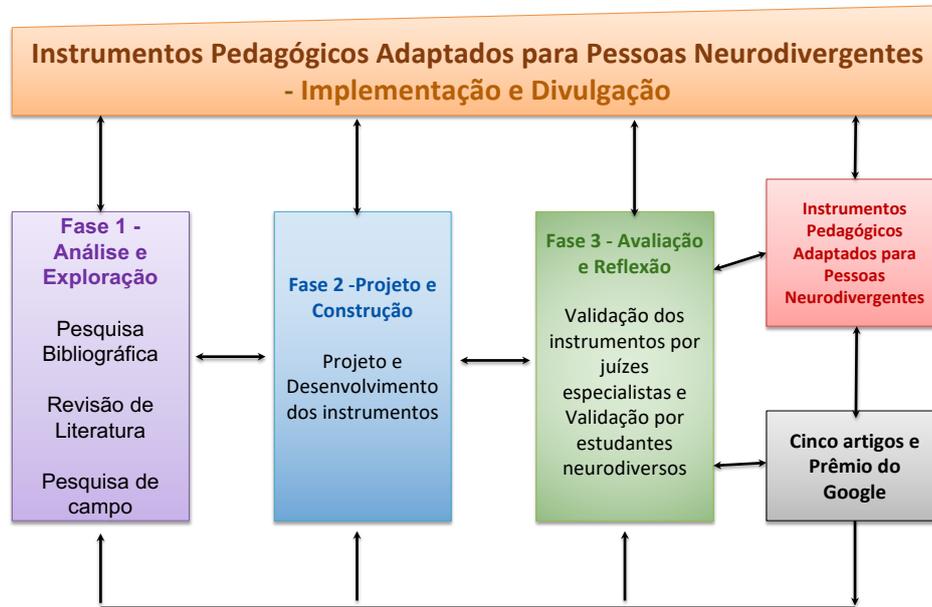


Figura 4.1: Projeto da Pesquisa segundo a Educational Design Research

Para responder as QP, esta pesquisa necessitou ser amparada pela literatura atual, sendo feitas investigações em ambiente de ensino. Esse processo fez parte da fase de análise e exploração. Com o arcabouço obtido na fase anterior, foram construídos os instrumentos pedagógicos, compreendendo a fase de projeto e construção. Durante a fase de avaliação e reflexão, utilizamos estratégias para validar o que foi proposto, gerando dados para seu refinamento. Como o processo é cíclico, foram necessárias algumas iterações nessas três fases para ter como resultado os instrumentos refinados, os quais são o produto principal desta pesquisa. Com o refinamento, obtivemos subprodutos do processo que foram a produção de três artigos que estão publicados [20; 21; 43] e um que está aceito e aguardando publicação. Além destes publicaremos mais um posteriormente. Além dos artigos, tivemos a conquista de um prêmio <sup>1</sup>.

Para respondermos nossas questões de pesquisa, organizamos os passos de execução conforme a EDR, a saber:

<sup>1</sup><https://research.google/programs-and-events/award-for-inclusion-research-program/recipients/?filtertab=2022>

- **QP1** - Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?
  - **Fase 1 - Análise e exploração:** O que se sabe sobre os estudantes com deficiência cognitiva na primeira disciplina de programação? Que impactos positivos e negativos o estudante neurodiverso pode ter e por que isso acontece? (**OBJ1**)
- **QP2** - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória?
  - **Fase 1 - Análise e exploração:** O que se sabe sobre as dificuldades percebidas por estudantes neurodiversos e professores de programação? (**OBJ2 e OBJ3**)
- **QP3** - Como tornar o ensino de programação inclusivo para pessoas neurodivergentes?
  - **Fase 2 - Projeto e construção:** Geração de um guia para criação de aulas de programação para auxiliar professores na prática docente com relação à estudantes neurodiversos. ; (**OBJ3, OBJ4 e OBJ5**)
  - **Fase 3 - Avaliação e reflexão:** Como professores e profissionais de apoio avaliam os instrumentos gerados na fase 2? quais resultados encontrados? (**OBJ6**)

Na Tabela 4.1, vemos a distribuição dos objetivos e questões de pesquisa nas fases do EDR.

Tabela 4.1: Questões de Pesquisa e Objetivos distribuídos nas fases do EDR

QP	OBJ	Análise e Exploração	Design e Construção	Avaliação e Reflexão
QP1	1	X		
QP2	2	X		
	3	X	X	
QP3	4		X	
	5		X	
	6			X

### 4.1.1 Fase 1: Análise e Exploração

Inicialmente foi feita uma busca para identificar o estado da arte no ensino de programação para pessoas com deficiência. Para isso, utilizamos um mapeamento sistemático observando o estado da arte das pessoas com deficiência em um contexto generalista [20].

Entendemos que atender as demandas de todos os tipos de deficiência é utópico, diante disso visamos identificar um escopo mais restrito. Com os resultados do mapeamento sugerindo falta de pesquisas focadas em comprometimento cognitivo, um novo estudo foi realizado com revisão sistemática para identificar iniciativas que envolviam o ensino de programação e pessoas com deficiência cognitiva [21];

Diante disso, visamos analisar o impacto que as deficiências cognitivas podem gerar nas competências necessárias para aprender a programar [43]. Esta etapa correspondeu a QP1 - “Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?”.

Por fim, tivemos em vista observar as percepções de alunos neurodiversos e seus respectivos professores sobre como ocorre o processo de ensino e aprendizagem de programação. Esta etapa correspondeu a QP2 - “Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória?”.

### 4.1.2 Fase 2: Design e Construção

Nesta fase, focamos na QP3 - “Como tornar o ensino de programação inclusivo para pessoas neurodivergentes?”. Desenvolvemos uma proposta de solução para o problema identificado na fase 1, gerando instrumentos pedagógicos e validando ele através do método Delphi.

Os instrumentos foram criados no formato de guia para os professores poderem produzir materiais de aulas de programação considerando as necessidades específicas do público neurodiverso. Utilizamos como base para as orientações os estudos realizados na fase 1 e documentos oficiais de órgãos de saúde que tratam das especificidades das deficiências cognitivas.

Este resultado visa auxiliar os professores de programação, favorecendo positivamente o processo de aprendizagem de estudantes com deficiência cognitiva ou neurodiversos, em uma primeira disciplina de programação;

### **4.1.3 Fase 3: Avaliação e Reflexão**

Nesta fase validamos os instrumentos produzidos na fase 2. Utilizamos o método Delphi com especialistas. A coleta de dados foi realizada com professores de programação e profissionais de pedagogia que apoiam pessoas neurodiversas. O recrutamento dos participantes foi realizado por e-mail, e a validação foi feita por meio de três rodadas de Delphi com os juízes especialistas respondendo formulários do Google Forms.

## **4.2 Considerações Finais do Capítulo**

Este capítulo apresentou a metodologia adotada para a execução desta pesquisa de doutorado. Apresentamos a distribuição das atividades nas fases do EDR conforme os objetivos e questões de pesquisa.

## Capítulo 5

# Influência da Neurodiversidade no Ensino e Aprendizagem de Programação

Para responder às questões de pesquisa, realizamos estudos, considerando as necessidades específicas dos estudantes neurodivergentes e das aulas de programação.

### **5.1 QP1 - Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?**

Aprender a programar requer diversas habilidades cognitivas, que podem ser barreiras para estudantes com dificuldades de aprendizagem [81].

Para responder a “QP1 - Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?” realizamos uma pesquisa exploratória para identificar e comparar as habilidades envolvidas na aprendizagem de programação e as dificuldades que estudantes com deficiência cognitiva podem enfrentar, discutindo seus impactos. O artigo completo foi publicado no IEEE Frontiers in Education (FIE) 2022 [43] e está disponível na íntegra no Apêndice C.

### 5.1.1 Metodologia

O objetivo principal do estudo foi verificar os impactos nas habilidades necessárias para aprender programação e as características específicas das deficiências cognitivas. Para isso foram definidas duas questões de pesquisa:

- QP1.1 - Quais as habilidades envolvidas no ensino de programação?
- QP1.2- Como as deficiências cognitivas podem impactar nas habilidades necessárias para programar?

Diante disso, na QP1.1 identificamos as habilidades importantes para aprender programação. Na QP2 analisamos o impacto nessas habilidades no contexto dos estudantes com deficiência cognitiva.

Para responder a QP1.2, realizamos uma revisão sistemática de literatura, adotando o protocolo de Kitchenham and Charters [47], utilizando a seguinte chave de busca:

**(“programming” AND “teaching” AND “skills”)**

As bases utilizadas foram a ACM Digital Library e a IEEE Xplore, resultando num total de 874 trabalhos.

Para a triagem aplicamos os critérios de inclusão a seguir:

- (CI1) - Artigos em inglês;
- (CI2) - Artigos publicados entre 2018 e 2022 (até janeiro);
- (CI3) - Artigos que contenham os termos da sequência de busca pelo menos no título, resumo ou palavras-chave;
- (CI4) - Artigos que tratem de habilidades envolvidas no ensino de programação.

Também foram aplicados alguns critérios de exclusão, a saber:

- (CE1) - Artigos repetidos;
- (CE2) - Artigos que não aderem ao tema;

- (CE3) - Estudos que não abordem habilidades no contexto de programação.

Após a aplicação dos critérios de inclusão e exclusão, obtivemos oito trabalhos relevantes para a pesquisa.

Para responder à QP1.2, utilizamos os achados da QP1.1 e a revisão realizada em 2021 [21], além de documento oficial de agências de saúde [6]. Nesse momento, identificamos algumas das principais características de cada deficiência cognitiva e relatos de como pessoas que têm uma delas podem reagir a situações que envolvem o ensino de programação. Partindo desta investigação, propusemos um cruzamento entre essas informações e as habilidades necessárias para programar que vimos na QP1, buscando encontrar evidências de como ajudar melhor esses estudantes.

### **5.1.2 Resultados**

Para alcançar nosso objetivo de pesquisa, partimos de uma revisão de literatura para identificar as principais habilidades para qualquer pessoa aprender a programar. Isso nos permitiu analisar posteriormente o impacto em estudantes com deficiências cognitivas utilizando as descobertas dessa revisão [21]. Selecionamos artigos que apresentavam habilidades necessárias para aprender a programar, compilamos os resultados e respondemos a primeira pergunta de pesquisa na próxima seção.

#### **QP1.1 - Quais habilidades estão envolvidas no ensino de programação?**

Para responder a QP1.1, nós buscamos na literatura um conjunto de habilidades que seriam necessárias para quem quer aprender programação.

A partir desta pesquisa selecionamos oito trabalhos que tinham relação com o tema [28; 39; 45; 44; 60; 61; 56; 76]. Com a triagem e análise desses artigos pudemos identificar e catalogar um conjunto de habilidades que foram usadas como base para esta pesquisa. Portanto, as principais habilidades que um estudante precisa para aprender a programar são:

- Resolução de problemas
- Habilidades Matemáticas
- Abstração

- Pensamento Crítico
- Compreensão de Texto
- Conhecimento em inglês
- Atenção aos detalhes
- Criatividade
- Gerenciamento de Tempo

A **resolução de problemas** está contida no Pensamento Computacional (PC) [26], assim como a programação. Essa habilidade é importante no aprendizado de programação, ao dizer respeito a planejar a solução de um problema, dividi-lo em etapas menores e buscar a solução mais eficiente."

É considerado desafiador para estudantes de Ciência da Computação aprender conceitos de programação, incluindo habilidades de resolução de problemas. Muitos estudantes têm dificuldades em entender o problema e convertê-lo em um programa [45].

Em programação, frequentemente chamamos um software de solução, justamente porque programar é resolver problemas. A resolução de problemas é uma habilidade muito importante na programação de computadores. Os estudantes precisam aprender a resolver problemas, entendendo que não existe uma solução única para todos, e que é necessário criar uma solução específica para cada problema [28]. Considerando que não existe uma única solução para um problema, podemos afirmar que estudantes com diferentes perfis de aprendizagem podem chegar a soluções diversas para o mesmo problema.

Outro ponto interessante é a possibilidade de o programador aprender a resolver problemas colaborativamente, já que, em um ambiente de trabalho formal, muitas vezes ele enfrentará a criação de soluções com uma equipe de programadores[56].

As **habilidades matemáticas** estão diretamente ligadas a programação. É muito comum que os exercícios de programação utilizem fórmulas matemáticas em sua composição. O estudante além de entender o problema, precisa também identificar qual seria a lógica matemática que caberia ali para chegar à sua solução.

A lógica matemática envolve compreender a relação de causa e efeito entre elementos. A partir de uma pré-condição, infere-se uma conclusão lógica. Esse processo também inclui

o sequenciamento de etapas para resolver um problema específico, muitas vezes utilizando procedimentos e fórmulas matemáticas [45].

**Abstração** é um conceito que pode auxiliar o estudante a criar uma estrutura para resolver um determinado problema. Normalmente, nessa fase, ao final da solução, ele refinará os passos em seu algoritmo [14]. O pensamento abstrato consiste em pensar sobre um problema de várias maneiras, detalhando-as [45].

O **Pensamento crítico** diz respeito a contextualizar, analisar e sintetizar sobre algo obtendo uma conclusão para resolver um problema. Ele pode representar um desafio para muitas pessoas. Compreender seu significado e inferir conclusões a partir das informações disponíveis é considerado uma tarefa difícil.

Outra habilidade muito importante e desafiadora é a de **compreensão de texto**, que envolve identificar a ideia principal, suas implicações e relacioná-la com o conhecimento prévio do leitor, facilitando assim a compreensão [45]. Na programação precisamos entender o problema a ser resolvido. Para isso, o programador deve ter uma boa habilidade de leitura e compreensão. Além disso, a maioria das abordagens para o ensino de programação foca em habilidades de resolução de problemas, muitas vezes negligenciando outras habilidades essenciais, como as linguísticas [30].

Quando falamos em programação de computadores, é evidente que o **conhecimento em inglês** pode ser um facilitador. As linguagens de programação utilizam o inglês em sua sintaxe [60], de modo que compreender o inglês ajuda a entender o significado de cada comando.

É evidente que a **atenção aos detalhes** é uma habilidade fundamental para aprender qualquer coisa, mas não é simples de desenvolver. Em um mundo dinâmico e cheio de estímulos, concentrar-se em algo específico torna-se desafiador. No contexto da programação, a atenção exige um alto nível de concentração do programador para analisar problemas ou código-fonte. Para programar, é essencial focar na produção, teste e correção de soluções, considerando tanto o fluxo ideal quanto as possíveis falhas [45].

Temos que a **criatividade** é uma habilidade muito importante, permitindo ao indivíduo gerar ideias, o que é especialmente benéfico na criação de algoritmos, já que não existe uma solução única para resolver um problema. A programação de computadores contribui para o desenvolvimento de habilidades como a criatividade [32], e ser criativo pode ser de grande

ajuda para o programador.

**Gerenciamento de tempo** eficazmente é considerado uma habilidade essencial para a programação [60]. Ao desenvolver um projeto, é crucial saber particionar o tempo de maneira adequada para a execução de cada etapa de resolução de um problema.

### **QP1.2- Como as deficiências cognitivas podem impactar nas habilidades necessárias para programar?**

Utilizamos o termo deficiência cognitiva como um termo guarda-chuva, englobando diversas deficiências que, de alguma forma, limitam as funções mentais do indivíduo [64]. Para responder à QP1.2, inicialmente trabalhamos com características associadas aos comprometimentos cognitivos extraídas da literatura e de agências de saúde [6], relacionando-as ao aprendizado de programação. Em seguida analisamos o conjunto de características de cada deficiência para fazer um cruzamento com a lista de habilidades que possuímos, conforme observamos na Tabela 5.1.

Existem vários tipos de deficiência que podem afetar a cognição dos indivíduos. Nesta pesquisa, focamos nas mais conhecidas; são elas [6]:

- **Autismo** - O Transtorno do Espectro Autista é caracterizado por déficits na comunicação, interação social e padrões repetitivos de comportamento.
- **TDAH** - O Transtorno do Déficit de Atenção e Hiperatividade é caracterizado por desatenção, impulsividade ou ambos que persistem ao longo do tempo.
- **Dislexia** - Na Dislexia, temos um padrão de dificuldades de aprendizagem, principalmente no reconhecimento e compreensão de palavras.
- **Deficiência Intelectual** - Pessoas com deficiência intelectual geralmente apresentam déficits em funções mentais como raciocínio, resolução de problemas e pensamento abstrato, entre outros.

As deficiências cognitivas possuem níveis de gravidade: leve, moderada, grave e profunda. Estes níveis podem variar de indivíduo para indivíduo, existindo inclusive casos de indivíduos com mais de uma deficiência simultaneamente, por exemplo, Autismo e Deficiência Intelectual [6].

Tabela 5.1: Deficiência Cognitivas em relação às Habilidades

<b>Deficiências</b>	<b>Habilidade Impactada</b>
Intelectual, Dislexia	Resolução de Problemas
Intelectual	Habilidades Matemáticas
Intelectual, TDAH	Abstração
Autismo	Pensamento Crítico
Intelectual, Dislexia, TDAH	Compreensão de Leitura
Deficiências Cognitivas	Conhecimento em inglês
TDAH	Atenção aos Detalhes
TDAH	Criatividade
Intelectual, Autismo	Gerenciamento de Tempo

Sobre **Resolução de Problemas** ainda há pouca pesquisa relacionando estudantes com Deficiência Intelectual e habilidades de programação. Esses estudantes têm mais dificuldade em funções adaptativas diretamente relacionadas a habilidades como resolução de problemas, adquiridas mais lentamente do que um estudante neuro típico [80]. Pessoas com Dislexia, possuem dificuldade em interpretar a língua escrita, o que pode dificultar o entendimento de enunciado de problemas [6].

Analisando as habilidades identificadas na QP1.1, podemos identificar que estudantes com Deficiência Intelectual terão dificuldades em habilidades principalmente ligadas a conceitos de lógica e **matemática** [80].

Estudantes com Deficiência Intelectual têm mais facilidade em entender conceitos concretos do que conceitos abstratos, então **abstração** é uma habilidade que pode ser impactada para esse público [80].

O **pensamento crítico** pode ser complexo, por exemplo, para aqueles que têm Autismo e comportamentos desafiadores, então atividades de programação que incorporam essa habilidade, devem ser projetadas para favorecer essa habilidade [48]. No TDAH, o pensamento crítico também pode ser prejudicado [6].

A habilidade de **compreensão de texto** pode ser impactada em pessoas com deficiência cognitiva por diversos aspectos. Indivíduos autistas podem ter atrasos na linguagem e

compreensão, falta de fala e deficiências na comunicação social. Pessoas com Deficiências Intelectuais moderadas podem ter progresso lento na leitura e escrita. Pessoas com Dislexia podem ter dificuldades de reconhecer palavras e ortografia [6].

As linguagens de programação são baseadas na língua inglesa, então qualquer pessoa que não tenha **conhecimento em inglês** pode ter dificuldades. Pessoas com deficiências cognitivas que não são nativos de países de língua inglesa podem precisar de suporte adicional [16]. Dentre os vários fatores que podem afetar o desempenho de programadores [54], especialmente iniciantes, a linguagem é um deles, tendo um impacto significativo no desempenho e no aprendizado de programação.

A **atenção aos detalhes** é uma habilidade essencial para a programação. Precisamos estar atentos quando estamos criando código, pois, além de criar a solução para o problema, também devemos obedecer à sintaxe da linguagem utilizada. Pessoas com TDAH podem ter dificuldade com atenção, hiperatividade ou ambos [87].

Habilidades que envolvem maior concentração, como a **criatividade**, seriam mais impactadas em estudantes com TDAH, pois geralmente é mais difícil para algumas pessoas com TDAH se concentrar [87].

Pessoas com deficiência intelectual podem ter dificuldade em compreender o tempo. **Gerenciar o tempo** pode ter impacto em pessoas autistas, já que estes têm dificuldade na alternância de atividades, organização e planejamento do seu tempo [6]. Diante disso pode haver danos ao cumprimento de prazos, por exemplo.

Estudos indicam que aprender habilidades de programação de computadores pode melhorar competências como sequenciamento, investigação e resolução de problemas em estudantes com deficiência intelectual [80]. Além disso, pesquisas mostram que estudantes com dislexia podem resolver problemas rapidamente, e a programação é uma área que pode fortalecer suas habilidades e ajudar a superar suas fraquezas [87; 29]. Alguns desenvolvedores profissionais com dislexia afirmaram que sua condição não representa obstáculos significativos em suas carreiras [81].

Uma das características que pessoas com autismo e TDAH podem ter é o hiper foco, que faz com que a pessoa dedique intensa concentração a algo que lhes interessa. Essa característica pode ser vantajosa se o estudante estiver interessado em aprender programação. Professores podem explorar maneiras de despertar esse interesse, como utilizar jogos para

atrair a atenção dos estudantes que gostam de jogar [19].

Portanto, é importante que ao invés de pensar apenas nas dificuldades associadas às deficiências cognitivas, também extraíamos seu potencial, identificando características de cada deficiência que podem facilitar o aprendizado da programação.

### 5.1.3 Discussão

Esta pesquisa permitiu observar que as habilidades envolvidas no ensino de programação podem gerar impacto em estudantes que tenham algum tipo de deficiência cognitiva. Como essa condição pode variar para cada indivíduo, podemos inferir que cada estudante pode precisar de um suporte específico. Diante disso, o papel do professor será essencial para entender em quais habilidades cada estudante está tendo dificuldades utilizando as recomendações adequadas e buscar extrair seu potencial para auxiliá-lo nos estudos. O professor sem capacitação adequada pode não ter sucesso, então como ele poderia identificar essas dificuldades e habilidades? E sobre abordagens, qual utilizar: ferramentas, jogos, testes específicos?

Diante disso, precisamos entender o que será necessário para dar esse suporte em pesquisas futuras. Portanto, a perspectiva do professor pode ser uma questão chave. As tecnologias estão em constante evolução, mas a escola resiste a abordar o currículo tradicionalmente, levando ao desinteresse dos estudantes [17]. Melhorar a didática para cursos de programação sempre foi um desafio para os professores [74]. Os professores muitas vezes não dão suporte adequado por falta de conhecimento sobre abordagens práticas [42]. Os professores devem dar um suporte constante e personalizado a cada estudante, permitindo que eles adquiram e aprimorem suas habilidades específicas de programação [31].

Outro ponto de vista seria o da perspectiva do ensino. O ensino de programação pode girar em torno de diferentes teorias de ensino, como o construtivismo [74]. Essas teorias amplamente difundidas podem facilitar ou dificultar o apoio adequado aos estudantes com deficiência cognitiva. É necessário aprofundar a discussão com o que já temos como alternativa e propor outras possibilidades buscando apresentar o que funciona para esse público.

Os desafios não se limitam a estes. Além disso, também podemos acrescentar a questão avaliativa, sendo muito importante e precisa ser justa nas limitações e potencialidades de cada estudante.

O Design Universal para Aprendizagem [42] é declarado como estratégia para o ensino

de pessoas com deficiência. Ele é norteado por três princípios básicos: (1) Múltiplos meios de engajamento, (2) Múltiplos meios de representação e (3) Múltiplos meios de ação e expressão. A ideia de oferecer ao estudante múltiplas formas para aprender parece ser uma abordagem promissora. Entretanto, ele não foi criado explicitamente para atender às especificidades do ensino de programação, nem para um tipo específico de deficiência. Ainda é necessário pesquisar mais sobre ele, especialmente para pessoas com deficiências cognitivas.

O W3C [83] possui um grupo de trabalho dedicado à criação de diretrizes de conteúdo web para pessoas com deficiência. Essas diretrizes são complementares às recomendações de Acessibilidade Web existentes e consolidadas [46]. É vasta a disponibilidade de conteúdo e materiais sobre programação na web, especialmente após a pandemia da COVID, que aumentou a necessidade de explorar novas oportunidades de ensino, especialmente a distância [2]. Essas notas são interessantes porque algumas podem ser aplicadas à criação de conteúdo, por exemplo, para aulas de programação.

As dificuldades existem, e os estudantes podem até levar mais tempo para aprender, entretanto, as evidências mostram que eles podem aprender. Não devemos olhar só para os desafios apresentados, mas também extrair o potencial dos estudantes, buscando seu melhor para favorecer seu aprendizado. O apoio constante é essencial para o êxito de pessoas com deficiência cognitiva em atingir seus objetivos [16].

#### **5.1.4 Ameaças a Validade**

Como qualquer pesquisa, esta tem ameaças à sua validade. Realizamos um estudo bibliográfico por meio de uma revisão sistemática da literatura de habilidades e posterior análise. Houve limitações devido aos termos e características de cada base de busca. Como o trabalho de triagem e análise é manual, fatores humanos podem afetar a extração e análise de dados e a interpretação dos artigos. Ainda assim, tentamos minimizá-los com mais de uma verificação.

## **5.2 QP2 - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória?**

Com a inclusão de estudantes com deficiência em cursos técnicos e de graduação, enfrentam-se diversos desafios, tanto para estes estudantes quanto para os professores. Especificamente tratando-se do ensino de Programação, temos diversas peculiaridades que podem impactar nesse processo. Para responder à questão de pesquisa “QP2 - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória?” retratamos as percepções de estudantes neurodiversos e seus respectivos professores, quais as dificuldades e como professores e estudantes estão tentando resolvê-las. A partir de um estudo de caso, observamos que os estudantes possuem dificuldades em iniciar seus novos programas sem auxílio e que professores não recebem capacitação adequada para atender as necessidades educacionais específicas desses estudantes.

O objetivo deste estudo de caso foi reportar como uma situação real de ensino de programação já acontece para pessoas com deficiência cognitiva, quais os principais problemas identificados por estudantes e professores e como eles estão tentando resolvê-los. Mais especificamente, discutimos os principais problemas de aprendizagem identificados em aulas de programação por alunos neurodiversos e seus professores e como eles estão tentando resolvê-los.

Para atingir esse objetivo, estabelecemos três subquestões de pesquisa a serem respondidas:

- QP2.1 - Quais são as dificuldades dos alunos neurodiversos em relação à aula de programação?
- QP2.2 - Qual é a percepção dos professores de programação em relação ao seu trabalho com alunos neurodiversos?
- QP2.3 - Como os professores de programação estão agindo em relação aos alunos neurodiversos?

O artigo referente a este estudo foi submetido e aceito para apresentação no IEEE Fron-

tiers in Education (FIE) 2024, que acontecerá em outubro de 2024. A versão final do artigo a ser publicada nos anais do FIE 2024 encontra-se na íntegra no apêndice G.

### **5.2.1 Metodologia**

Esta pesquisa de caráter exploratório e descritivo [36], tem o objetivo de compreender o estado do ensino de programação para estudantes neurodiversos atualmente. Os participantes deste estudo pertencem a dois perfis: (1) estudantes neurodiversos cursando sua primeira disciplina de programação e (2) seus respectivos professores. Os perfis foram escolhidos devido à complexidade do fenômeno em estudo, focando apenas em estudantes com diagnóstico confirmado de algum transtorno do neurodesenvolvimento, apesar de sabermos que existem vários estudantes que podem apresentar características de neurodiversidade sem diagnóstico formal. Este estudo constitui um estudo de caso exploratório de caso único [89; 15; 37].

Consideramos o ensino de programação o fenômeno central deste estudo. O contexto inclui a primeira turma de programação em dois cursos de graduação e um curso técnico de ensino médio. As unidades de análise são (1) os estudantes neurodiversos e (2) seus respectivos professores dentro deste contexto.

#### **Perfil dos Participantes**

Para este estudo, investigaram-se as impressões de estudantes neurodiversos e seus respectivos professores no contexto do primeiro curso de programação. Os participantes da pesquisa foram, portanto, estudantes com algum diagnóstico de neurodiversidade matriculados em um curso da área de Computação de nível superior ou em um curso técnico de nível médio que estivessem cursando a primeira disciplina de programação. Além destes, também selecionamos seus respectivos professores na primeira disciplina de programação.

Os estudantes participantes foram selecionados por meio de contato com duas instituições de ensino, o Instituto Federal de Educação, Ciência e Tecnologia da Paraíba – Campus Campina Grande e a Universidade Federal de Campina Grande – Campus Campina Grande. Nesse contato tivemos em vista localizar estudantes que tivessem diagnósticos emitidos por profissionais especializados. As instituições possuíam esses diagnósticos, que eram recebi-

dos no momento da admissão dos estudantes nos respectivos processos seletivos. Sabemos que alguns estudantes podem apresentar características compatíveis com algum transtorno de neurodesenvolvimento, porém não têm laudo ou preferem não ser identificados como tal. Como não possuímos qualificação profissional para diagnosticá-los, optamos por selecionar apenas aqueles que já têm laudo e, por livre e espontânea vontade, concordaram em participar da pesquisa.

Como este campo de pesquisa é muito restrito, com poucas pessoas em relação ao universo de estudantes, estabelecemos que seriam selecionados todos os estudantes que se enquadrassem no perfil e estivessem regularmente matriculados em uma primeira disciplina de programação. Foram localizados, portanto, seis estudantes e três professores: cinco estudantes de graduação e um estudante de curso técnico do ensino médio. Para preservar a privacidade dos estudantes, os identificamos por meio de uma sigla, onde os estudantes de graduação são identificados pela letra G seguida de um número de controle, e a sigla T1 identifica o estudante de curso técnico.

Em se tratando dos diagnósticos, observamos uma diversidade, embora todos se enquadrem na categoria de deficiência cognitiva ou neurodiversidade, conforme vemos na Tabela 5.2.

Tabela 5.2: Perfil dos Estudantes

<b>Pseudônimo</b>	<b>Diagnóstico</b>
G1	Síndrome de Asperger (TEA)
G2	Síndrome de Asperger (TEA)
G3	Síndrome de Asperger (TEA) e TDAH
G4	TDAH
G5	TDAH e Transtorno de Humor
T1	Síndrome Noonan, TDAH, e Deficiência Intelectual

O que antes era conhecido como Síndrome de Asperger atualmente está inserido no Transtorno do Espectro Autista, segundo o Manual Diagnóstico e Estatístico de Transtornos Mentais (DSM-5) [6]. Foram utilizados os laudos dos estudantes para triagem, feitos em épocas distintas, por isso mantemos o termo. Além disso, o Transtorno de Humor está

incluído no Transtorno do Déficit de Atenção e Hiperatividade (TDAH). Entretanto, a Síndrome de Noonan é uma condição física, mas como estudante em questão também possui TDAH e Deficiência Intelectual (DI), o qualificou para participar do estudo.

Após selecionar estudantes elegíveis, entramos em contato com o professor de cada estudante na primeira disciplina de programação. Como podemos ver na Tabela 5.3, os professores participantes da pesquisa possuem formação em Computação e ampla experiência em ensino.

Tabela 5.3: Perfil dos Professores

<b>Pseudônimo</b>	<b>Formação</b>	<b>Maior Titulação</b>	<b>Experiência de Ensino</b>
PG1	Ciência da Computação	Mestre	11 Anos
PG2	Ciência da Computação	Doutor	17 Anos
PT1	Informática	Especialização	16 Anos

### **Coleta de Dados**

A coleta de dados foi realizada por meio de entrevistas semiestruturadas, dessa forma, foi obtido informações sob a perspectiva do participante [53]. Criamos um roteiro para cada unidade de análise, agrupando as perguntas em partes, sendo um roteiro para os estudantes neurodiversos e outro para seus professores. Diante disso, foi orientado aos entrevistados apresentar o máximo de informações relevantes possíveis para esta pesquisa. As entrevistas foram realizadas individualmente em momentos diferentes, fora da sala de aula, para que os participantes se sentissem confortáveis durante todo o processo.

O roteiro de entrevista dos professores foi dividido em três partes, conforme vemos na Tabela 5.4. A parte 1 teve em vista coletar os perfis dos participantes com seus dados de formação e se alguma formação apoiou suas aulas envolvendo estudantes com deficiência. Na parte 2, foi detalhada a experiência do professor com estudantes com deficiência em programação introdutória. Na parte 3, finalizamos a entrevista coletando dados sobre a disciplina.

O roteiro de entrevista com os estudantes foi organizado em seis partes, conforme vemos na Tabela 5.5. A parte 1 coleta dados de nome e idade, que servem apenas como controle. O nome do participante não será divulgado, preservando sua privacidade. Na parte 2, abrimos

espaço para o estudante falar sobre suas principais dúvidas sem o relembrar do conteúdo para que ele possa tentar explicar por conta própria. A parte 3 especifica os principais conteúdos abordados na primeira disciplina de programação. A oportunidade permitiu que os estudantes relembassem o conteúdo e relatassem suas dúvidas e entendimentos sobre cada um deles. A parte 4 teve em vista entender as dificuldades dos estudantes ao fazer exercícios de programação. A parte 5 teve como objetivo entender as dificuldades nas avaliações. Por fim, na parte 6, permitimos que os estudantes se expressassem livremente para concluir a entrevista.

Tabela 5.4: Roteiro de Entrevista Professores

---

<b>Parte 1 Perfil do Participante</b>
Formação Acadêmica; Quantos anos de experiência em sala de aula. Experiência envolvendo estudantes com deficiência em sala de aula; Treinamento em Educação Inclusiva.
<b>Parte 2 Experiência com Estudantes com Deficiência</b>
Estudantes com deficiência na disciplina de programação em semestres anteriores; Tipos de Deficiências; Sobre os estudantes com deficiência neste semestre; Adaptações das aulas para os estudantes; Adaptação dos materiais e atividades para os estudantes com deficiência; Interesse em conhecer formas de auxiliar esses estudantes em aulas de programação.
<b>Parte 3 Condução da Disciplina</b>
Detalhe como a disciplina é conduzida atualmente.

---

Tabela 5.5: Roteiro Entrevista Estudantes

---

<b>Parte 1 Perfil do Participante</b>
Nome; Idade.
<b>Parte 2 Dúvidas sobre Programação</b>
Detalhar quais seriam suas dúvidas em programação
<b>Parte 3 Conteúdos</b>
Definição de Algoritmo; Variáveis; Entrada e Saída; Estruturas Condicionais; Estruturas de Repetição; Vetores; Funções.
<b>Parte 4 Exercícios</b>
Enunciado; Entrada; Saída.
<b>Parte 5 Avaliação</b>
<b>Parte 6 Informações Gerais</b>

---

## 5.2.2 Resultados

Realizamos uma análise de conteúdo dos dados coletados durante as entrevistas [8; 13; 71]. As entrevistas foram realizadas no segundo semestre de 2022. Nosso objetivo foi observar como ocorre o processo de ensino e aprendizagem no contexto de estudantes neurodivergentes da primeira disciplina de programação.

Entrevistamos seis estudantes que possuíam laudo de algum transtorno do neurodesenvolvimento e seus três respectivos professores da primeira disciplina de programação. Como um dos professores tinha mais de um estudante que se encaixava no estudo, observamos que o número de professores era menor que o número de estudantes.

A análise dos dados foi manual porque a amostra era pequena. Lemos cuidadosamente as respostas das entrevistas e compilamos os resultados segundo as perguntas de pesquisa abaixo.

### **QP2.1 Quais são as dificuldades dos estudantes neurodiversos em relação à aula de programação?**

A primeira pergunta aos estudantes (Guia de Entrevista - Parte 2) foi para buscar informações sobre quais seriam as dúvidas que eles tinham em programação, por exemplo, se seria na sintaxe da linguagem ou em algum conceito específico. Nosso objetivo era extrair as dificuldades neutramente, sem mencionar nenhum conteúdo específico. Com esta estratégia, os estudantes informariam ao entrevistador se tivessem alguma dúvida específica. A partir das respostas, identificamos os principais termos relacionados ao conteúdo e categorizamos conforme a Tabela 5.6. A partir desta tabela observamos como os estudantes responderam sobre suas dificuldades na programação sem nenhum lembrete sobre os tópicos de programação.

Tabela 5.6: Dúvidas em Programação

<b>Dificuldades</b>	<b>G1</b>	<b>G2</b>	<b>G3</b>	<b>G4</b>	<b>G5</b>	<b>T1</b>
Exercícios	X					
Materiais	X					
Aplicação dos Algoritmos		X				
Laços				X		
Sem dúvidas			X		X	
Dúvidas sobre tudo						X

Observando as respostas sobre dificuldades em geral, os estudantes variaram em suas respostas, porém vemos que, do total de estudantes entrevistados, dois acreditam não ter dificuldades em programação. Apenas um estudante acredita ter dificuldade em todos os tópicos mencionados.

Após cada estudante expressar espontaneamente suas dúvidas sobre programação, perguntamos a ele sobre suas dificuldades em relação aos tópicos de programação. Com base em cada tópico mencionado, o estudante disse se teria e qual seria sua dúvida. Na Tabela 5.7, vemos como os estudantes responderam suas perguntas nos tópicos de programação. É marcado com “X” se o estudante tiver dúvidas sobre o tópico perguntado.

Tabela 5.7: Dúvidas em Programação por Tópicos

<b>Tópicos</b>	<b>G1</b>	<b>G2</b>	<b>G3</b>	<b>G4</b>	<b>G5</b>	<b>T1</b>
Definição de Algoritmos					X	X
Variáveis						X
Entrada/Saída						X
Condicional	X			X		X
Laços	X			X	X	X
Vetores	X			X	X	X
Funções	X					

Podemos observar que, dos estudantes que responderam que tinham dúvidas, nenhum indicou ter dúvidas sobre os tópicos iniciais, entrada e saída, e variáveis. Em relação ao conteúdo das funções, os estudantes G2,G3,G4,G5 e T1 ainda não estudaram esse tópico.

Em seguida, os estudantes responderam sobre suas dificuldades nos exercícios e avaliações (Guia de Entrevista Partes 5 e 6). Cinco dos seis estudantes entrevistados relataram que sua maior dificuldade foi entender o enunciado das questões. Essa dificuldade também é relatada em relação às avaliações. Um dos estudantes também destacou que muitos exercícios envolvem cálculos matemáticos como uma dificuldade.

### **QP2.2 Qual a percepção dos professores de programação em relação ao seu trabalho com estudantes neurodiversos?**

Entrevistamos três professores de programação para responder a essa pergunta. Na entrevista, perguntamos a cada professor sobre sua experiência com estudantes com deficiência. Todos os professores entrevistados eram professores de pelo menos um dos estudantes neurodiversos que também entrevistamos. Todos os entrevistados responderam que tiveram estudantes com deficiência em semestres anteriores, receberam estudantes com diferentes tipos de necessidades e não tiveram nenhuma formação específica para ajudar esses estudantes.

Para identificar os professores, utilizamos a sigla PG seguida de um número de controle para os professores da graduação, e a sigla PT1 para o professor do curso técnico.

O professor PG1 percebeu a dificuldade do seu estudante neurodiverso, principalmente em atenção e memorização, buscando estratégias para ajudá-lo. O professor do PG2, que tem 4 estudantes neurodiversos, percebeu que cada estudante tem diferentes níveis de interação e atenção à disciplina. Da mesma forma que há estudantes neurodiversos que apresentam bom desempenho e interagem bem, também há estudantes neurodiversos que não interagem e não buscam responder perguntas, prejudicando seu desempenho na disciplina. O professor PT1 notou que seu estudante neurodiverso tem muitas dificuldades, baixa interação e falta de atenção.

Finalmente, todos os professores entrevistados acreditam que precisam de treinamento adequado para lidar com a neurodiversidade, que se enquadra em deficiências não visíveis, e que as instituições educacionais ainda precisam melhorar o suporte aos professores neste contexto.

### **QP2.3 Como os professores de programação estão agindo em relação aos estudantes neurodiversos?**

Com base na percepção dos professores sobre os estudantes neurodiversos, relatada na QP2, surge outra questão importante, que é como esses professores tentam dar suporte aos estudantes neurodiversos, mesmo sem terem formação adequada. Diante disso, temos informações essenciais que nos ajudam a refletir sobre como mais do que práticas tradicionais podem ser necessárias para ensinar esses estudantes, conforme mostrado na Tabela 5.8.

Tabela 5.8: Estratégias do Professor

<b>Estratégia</b>	<b>PG1</b>	<b>PG2</b>	<b>PT1</b>
Aumento da carga horária do curso	X		
Aumento do tempo para realização de atividades ou avaliações	X	X	X
Uso do Monitor	X	X	X
Intervenção constante	X		
Repetição da explicação		X	X
Uso de Imagens			X

O professor PG1 relatou uma tentativa de aumentar a carga horária dos estudantes, fa-

zendo com que eles usassem dois semestres em vez de um para concluir todo o conteúdo da programação.

Todos os professores indicaram aumentar o prazo para conclusão de exercícios e atividades para estudantes neurodiversos. Eles também relataram que têm um monitor para auxiliar o estudante.

Apenas o professor PG1 buscava intervenção constante para o estudante como estratégia, que poderia ser dele mesmo ou do monitor. O que ele chamou de intervenção constante foi a ação de estar sempre perguntando ao estudante se ele precisava de ajuda, se havia entendido, entre outras. Dois dos três professores usaram a estratégia de explicar o mesmo mais de uma vez, buscando a compreensão dos estudantes neurodiversos. Apenas o professor PT1 relatou adaptava seu material de aula, usando mais imagens e menos texto o máximo possível.

### 5.2.3 Discussão

Ao avaliar os dados das entrevistas, obtivemos percepções de cada uma de nossas unidades de análise, guiados pelas questões de pesquisa, discutidas abaixo.

#### **Percepções das dificuldades de estudantes neurodiversos em aulas de programação**

Os estudantes entrevistados relataram dificuldades nas aulas de programação, gerando reflexão sobre o que poderia ser feito para que esses estudantes tenham sucesso no curso. Podemos observar algumas questões-chave abaixo:

- **Materiais e Exercícios** - Os estudantes apontam dificuldades nos materiais de aula e exercícios, gerando reflexão sobre como estes poderiam ser adaptados para um público neurodiverso. Quais seriam os pontos a melhorar?
- **Entendimento da aplicação da Programação** - Foi mencionado pelos estudantes que há dificuldade em entender a aplicação de Algoritmos no mundo real. Essa dificuldade sinaliza a necessidade de maior contextualização entre conceitos de Programação e as necessidades práticas do mercado, por exemplo.
- **Interpretação dos problemas** - Os estudantes alegaram dificuldade em interpretar o que os problemas pedem. Isso ocorre tanto em exercícios quanto em avaliações. É

fato que, em alguns tipos de deficiência cognitiva, temos algumas características que podem impactar, por exemplo, o raciocínio lógico e a abstração [43], o que exigiria uma reformulação na forma como as questões são escritas.

- **Criando Algoritmos sem ajuda** - Outro problema crítico identificado nas respostas foi a dificuldade em saber como iniciar um algoritmo sem ajuda, o que pode sugerir a necessidade de dispositivos que os lembrem da estrutura e dos principais elementos de um algoritmo no escopo do problema que desejam resolver naquele momento.
- **Lembrando o que fazer em exercícios e avaliações** - Foi relatada dificuldade em saber quais elementos de programação devem ser utilizados para cada situação-problema.

Os problemas identificados durante a coleta de dados fornecem fortes evidências de que é necessário reavaliar a abordagem para o ensino de programação em sala de aula. Várias estratégias podem ser consideradas para melhorar a experiência de aprendizagem do estudante, como revisar os textos das declarações de exercícios e materiais para minimizar seu impacto negativo. Essas medidas visam fornecer um ambiente de aprendizagem mais inclusivo e acessível, garantindo que todos os estudantes possam se envolver e aprender a codificar eficazmente.

### **Percepções de professores de programação sobre estudantes neurodiversos**

A percepção dos professores sobre estudantes neurodiversos indica algumas questões que precisam ser exploradas, como pode ser visto abaixo:

- **Falta de treinamento adequado** - Os professores reconhecem que os estudantes neurodiversos precisam de assistência adequada nas aulas de programação e todos estão interessados em atender adequadamente os estudantes com deficiência em suas aulas. Há um problema, mas não há treinamento adequado para esses professores. Buscar maneiras de auxiliar os estudantes com suas necessidades específicas toma tempo e trabalho extras, dificultando encontrar soluções sem a devida orientação.
- **Comunicação insuficiente** - Ambas as instituições de ensino possuem centros de apoio a pessoas com deficiência. Esses centros visam acolher e buscar suporte ade-

quando para todos os estudantes com necessidades específicas no ambiente de ensino. Na prática, esse suporte é mais efetivo para aqueles com deficiências sensoriais ou físicas (contratação de intérpretes de Libras para dar suporte a estudantes surdos, por exemplo). No caso de estudantes neurodivergentes, esse suporte se limita a sinalizar que o estudante tem uma necessidade específica, que precisa de suporte, mas não está claro qual suporte lhe é devido.

Em relação à falta de comunicação por parte da instituição de ensino, observa-se que setores especializados geralmente informam os professores quando há estudantes com deficiência e, dentro do possível, fornecem orientações que podem ser úteis. No entanto, o suporte ainda é insuficiente para estudantes com deficiências que não são “visíveis”, deixando os professores preocupados com o progresso de seus estudantes e sem orientação adequada sobre como agir. Há uma clara necessidade de melhorar o suporte e fornecer orientações mais abrangentes para os professores poderem atender de forma mais eficaz às necessidades desses estudantes.

Diante da demanda por treinamento insuficiente, cada professor buscou soluções alternativas com base em suas percepções. Isso gerou nossa terceira questão de pesquisa, que trata de como os professores buscaram abordar a inclusão de estudantes neurodiversos em suas aulas de programação.

### **Percepções de Professores de Programação Atuando com estudantes Neurodiversos**

Com treinamento inadequado, os professores relataram buscar estratégias para auxiliar os estudantes neurodiversos.

- **Aumento do Tempo** - Houve consenso entre os professores de que esses estudantes precisavam de mais tempo para realizar as atividades do que os demais estudantes da turma. Um dos professores chegou a usar a estratégia de fazer o estudante estudar um conteúdo que duraria apenas um semestre em dois. O tempo extra pode ser interessante, principalmente para estudantes que têm níveis de desatenção e demoram mais para se concentrar.
- **Explicação** Uma estratégia adotada por dois professores foi explicar o assunto mais de uma vez, buscando variar as formas de falar para minimizar as dificuldades dos

estudantes. Dessa forma, estudantes com diferentes níveis de dificuldade podem se beneficiar.

- **Adaptação de Materiais e Atividades** Apenas um professor relatou que tenta adaptar suas aulas. Sua estratégia é aumentar o número de imagens. Usar imagens é insuficiente para uma adaptação adequada, mas é um começo.

Como podemos ver, as ações dos professores em relação aos estudantes neurodiversos ainda carecem de melhores estratégias para ajudar efetivamente esses estudantes nas aulas de programação. Os professores não têm suporte adequado para mostrar a eles o que pode ser feito. Sabemos que as adaptações exigem tempo e conhecimento, mas devemos entender que elas são essenciais para incluir efetivamente esses estudantes nas aulas de programação.

#### 5.2.4 Ameaças a Validade

Qualquer pesquisa apresenta ameaças à validade. Neste caso, identificamos o número de participantes como uma limitação, o que justificamos porque o número de sujeitos elegíveis é pequeno na população. Portanto, não podemos garantir que os achados seriam generalizados para outros participantes com diferentes tipos de comprometimento cognitivo em diferentes níveis.

Outra limitação seriam os fatores humanos. Eles podem afetar o processo de triagem de entrevistas e análise de conteúdo, pois o mesmo pesquisador definiu, realizou e analisou a pesquisa.

### 5.3 Considerações Finais do Capítulo

Este capítulo visou responder às questões de pesquisa “QP1 - Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?” e “QP2 - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória?”

Na Seção 5.1, respondendo a QP1, vimos que algumas características das deficiências cognitivas podem impactar nas habilidades necessárias para aprender programação, porém nem todo impacto seria negativo, e com as devidas adaptações conseguiremos auxiliar esses

estudantes. A área de Educação em Computação ainda é relativamente nova [85], entretanto já existem esforços consideráveis na literatura para discutir sobre as habilidades necessárias para aprender programação. Esta parte da tese buscou trazer uma visão teórica dessas habilidades do ponto de vista da inclusão. Diante disso, gerou discussão de novas perspectivas para ajudar pessoas com deficiência cognitiva. Essas pessoas podem aprender, mas de uma forma diferente e com o devido suporte de professores e instituições.

Observamos nesse estudo que uma existe um grande conjunto de habilidades importantes para aprender a programar. Essas habilidades estão principalmente ligadas à Matemática e à Lógica. Além disso, identificamos que as deficiências cognitivas podem gerar impacto nas habilidades necessárias para programar, porém, nem todo tipo de deficiência impactaria em todas as habilidades.

Na Seção 5.2, respondendo a QP2, vimos as percepções de professores de programação e estudantes neurodiversos no contexto de uma disciplina de programação introdutória. Observamos que os professores carecem de capacitação e informações adequadas para auxiliar esses estudantes e, que os estudantes neurodiversos possuem dificuldades em compreender a aplicação da programação, bem como lembrar sozinhos o que fazer.

Vimos que ainda há muito a ser discutido e explorado e que os professores estão interessados em como proceder, mas precisam de orientação. Com suporte adequado, acreditamos que os professores podem atingir positivamente esse grupo de estudantes, que beneficiará seu aprendizado por meio de condições adequadas às suas necessidades. As instituições de ensino devem investir em suporte especializado para professores, o que pode fazer grande diferença no acolhimento de estudantes com necessidades específicas. Além disso, pode ser decisivo para a permanência desses estudantes no curso.

Dadas as dificuldades relatadas por estudantes e professores, vimos uma demanda aparente para considerar novas abordagens para o ensino de programação para pessoas neurodiversas. É essencial desenvolver estratégias e recursos pedagógicos inclusivos e adaptados às características e habilidades desses estudantes, proporcionando-lhes igualdade de oportunidades. Isso pode envolver a criação de ambientes de aprendizagem acessíveis, usando linguagem e instruções claras, e fornecendo suporte individualizado e estruturas de aprendizagem diferenciadas. Superar esse problema requer um comprometimento coletivo para garantir que todos os estudantes tenham acesso igualitário às oportunidades de aprendiza-

gem de programação, independentemente de suas habilidades cognitivas.

Iniciativas de inclusão beneficiam estudantes que têm limitações permanentes, mas também podem ajudar qualquer estudante que tenha uma limitação temporária ou mesmo dificuldades decorrentes de uma formação educacional precária, pois a programação não é algo que se aprende tradicionalmente ao longo da vida escolar.

Observamos que este campo de pesquisa ainda é pouco explorada e possui lacunas a serem preenchidas. Portanto, no próximo capítulo apresentamos uma iniciativa voltada para esse fim.

## Capítulo 6

# Instrumentos Pedagógicos Adaptados para Pessoas Neurodivergentes

Para responder à questão de pesquisa QP3: “Como tornar o ensino de programação inclusivo para pessoas neurodivergentes?”, construímos o que chamamos de instrumentos pedagógicos para criação de aulas de programação adaptado para estudantes neurodiversos. Estes instrumentos têm a forma de um guia, destinado aos professores de programação, para que esses possam adaptar seus materiais de aula para atender necessidades específicas do público neurodiverso. Dessa forma, o professor pode criar uma aula desde o princípio já pelo guia ou adaptar um material que ele já possua. Este guia foi pensado e construído considerando recomendações de órgãos de saúde mundiais em conjunto com adequações voltadas a realidade vivida nas aulas de programação.

O presente estudo foi dividido em duas fases: (1) elaboração do guia para criação de roteiros de aulas de programação acessíveis e (2) validação do guia por professores de programação e profissionais da área de neuro psicopedagogia, por meio da técnica Delphi.

### **6.1 O guia para aulas de programação inclusiva ao público neurodivergente**

No contexto educacional, observamos a necessidade de materiais que auxiliem professores em suas aulas e sejam inclusivos. Diante disso, concebemos um guia para nortear os profes-

sores a produzirem aulas de programação inclusiva ao público neurodivergente. O guia em sua íntegra se encontra no Apêndice H.

Para poder auxiliar os estudantes, os professores precisam compreender as reais necessidades deles. Como o foco deste trabalho são os estudantes neurodiversos, trabalhamos com as características inerentes a esse público-alvo e elaboramos este guia para apoiar os professores, focando-se em aulas de programação introdutória.

Do ponto de vista das necessidades específicas de estudantes com deficiência cognitiva, buscamos inicialmente mapear as principais características para adequá-las ao contexto desta pesquisa. Buscamos documentos norteadores de órgãos de saúde, como o Manual Diagnóstico de Transtornos Mentais [6]. Este documento foi o ponto de partida, já que aborda as características de cada tipo de deficiência cognitiva, enumerando quais as principais dificuldades enfrentadas por estes.

O conteúdo extraído do levantamento bibliográfico foi analisado e recomendações foram extraídas para serem incorporadas a estrutura do guia para aulas de programação. Vale ressaltar que as recomendações geradas buscam unir as informações dos órgãos competentes de saúde e informações inerentes ao ensino de programação. Desse modo, o guia foi dividido em seções, as quais apresentam um escopo específico e uma explicação para sua existência no documento embasada na literatura.

O guia é dividido em partes, sendo uma parte para orientações para plano de aula, outra para roteiro de aula, e por fim, temos recomendações para adaptação do conteúdo, bem como exercícios e avaliações de programação.

### 6.1.1 Orientações para Plano de Aula

O planejamento de aulas faz parte do trabalho do professor. No nosso guia, temos uma área dedicada a orientações para criação de um plano de aula, que seria alinhado com o roteiro de aula inclusivo.

O plano de aula está dividido em seções, como podemos ver na Tabela 6.1.

Para melhor planejamento das aulas, considerando as especificidades dos estudantes neurodiversos, é importante que o conteúdo da disciplina seja dividido em partes menores, que aqui chamamos de **Unidade de Conteúdo**.

Como vimos no estudo abordado na Seção 5.1, existem **habilidades** necessárias para

Tabela 6.1: Plano de Aula

<b>Seção</b>	<b>Objetivo</b>
Unidade de Conteúdo	Indicar a qual unidade o plano pertence
Habilidades exploradas nesta unidade	Apresentar o conjunto de habilidades envolvidas nesse conteúdo
Roteiro Teórico de Estudo	Breve resumo do que será visto na unidade
Exercícios	Resumo das atividades que serão utilizadas

aprender programação que podem ser impactadas em estudantes com deficiência cognitiva. Diante disso, é importante observar quais seriam as principais habilidades envolvidas em cada unidade de conteúdo, para ter um panorama mais adequado de como adaptar o conteúdo considerando essas habilidades.

Um dos intuitos de adaptar materiais de aula é o de fornecer aos estudantes uma fonte confiável de estudo, que seja adequada as suas necessidades. Portanto, é importante que se especifique no plano de aula qual será o conteúdo **teórico** que será abordado naquela unidade.

No ensino de programação, uma parte muito importante é a resolução de **exercícios** para construção de Algoritmos. Diante disso, é importante que o professor planeje as atividades que serão ofertadas para os estudantes poderem treinar a construção de códigos para aquela unidade de conteúdo.

Portanto, temos que a parte de plano de aula do guia é disponibilizada para o professor planejar sua aula, já refletindo como esta será adaptada conforme apresentamos na próxima parte, sendo a da construção do roteiro de aula.

### 6.1.2 Orientações para Roteiro de Aula

O conteúdo de programação possui variadas fontes de estudo. Por exemplo, temos a disposição livros físicos e digitais, bem como apostilas e tutoriais na Internet. Esta vasta dispo-

nibilidade infelizmente não se aplica a pessoas com deficiência. Pessoas com necessidades específicas podem ter dificuldade em encontrar materiais que sejam adaptados às suas necessidades. Portanto, no guia proposto por esta pesquisa apresentamos um template para que o professor de programação possa gerar roteiros de aula adequados as necessidades de estudantes neurodiversos.

Assim como no template para o plano de aula, temos o template do roteiro, divididos em seções, onde cada seção tem uma finalidade específica. Podemos ver na Tabela 6.2 a disposição das seções do template para roteiro de aula.

Tabela 6.2: Roteiro de Aula

Seção	Objetivo
Qual é nossa unidade de conteúdo?	Indicar a qual unidade o roteiro pertence
O que queremos aprender nesta unidade	Apresentar ao estudante um resumo do que será visto no roteiro
Contextualizando	Fazer ligações do conteúdo com o mundo real e concreto
O que vamos precisar	Deixar claro para o estudante o que ele precisará para trabalhar com esse conteúdo
Roteiro para Estudo	Parte teórica do roteiro, explicando o conteúdo seguindo as orientações disponibilizadas no guia
Resumo do que foi apresentado na unidade	Resumir o que foi visto na unidade
Para saber mais	Disponibilizar fontes alternativas de consulta ao conteúdo do roteiro
Exercícios e Exemplos de Código	Disponibilizar exercícios seguindo as orientações disponibilizadas no guia

O roteiro se inicia mostrando claramente **qual é a unidade de conteúdo** que será abordada. Para o estudante neurodiverso é importante ter essa informação para que ele possa

saber o que esperar do material.

Em seguida, na mesma linha de trabalhar com a expectativa do aluno, temos uma seção onde se mostra **o que ele irá aprender na unidade**, ou seja, quais serão os tópicos de conteúdo que serão vistos.

Pessoas com deficiência cognitiva podem ter dificuldade em entender analogias e abstrações, portanto inserimos no roteiro uma seção de **Contextualização**, para fazer com que o professor aponte pelo menos um exemplo de utilização do conteúdo da unidade, fazendo ligação com o mundo real.

É de suma importância ser claro com alunos neurodiversos. Então especificar **quais materiais ou ferramentas serão necessários** para esta aula é crucial. Além de disponibilizá-los, também é necessário que o professor demonstre como utilizá-los. No caso de alunos neurodivergentes, embora não seja necessário usar tecnologias assistivas como leitores de tela, é importante que os alunos aprendam a utilizar ferramentas como Integrated Development Environment (IDE) e a executar os códigos presentes no roteiro caso este seja feito em um ambiente como o Google Collab<sup>1</sup>.

A parte principal do material é o **Roteiro para Estudo**. É nela onde o professor irá colocar o conhecimento teórico envolvido na unidade de conteúdo, bem como exemplos de aplicação. Para construção desta seção ele irá consultar as orientações para criação de conteúdo disponibilizadas no guia.

Para favorecer o entendimento e uma posterior consulta, é interessante apresentar um **resumo** em tópicos ao final do roteiro. É importante destacar os conceitos-chave aprendidos na unidade nesta seção para o aluno lembrar do que ela se trata. Alunos neurodivergentes podem enfrentar dificuldades em planejamento e organização, o que pode afetar a aprendizagem, mesmo em alunos com inteligência acima da média. Portanto, estruturar os principais conceitos em tópicos no final do roteiro ajuda a organizar mentalmente o conhecimento para esses estudantes.

Além do conteúdo disponibilizado no roteiro, inserimos uma seção **para saber mais**, com o intuito de disponibilizar fontes de estudo adicional, para conferir possibilidade ao estudante de procurar outras fontes de conteúdo confiáveis caso ele tenha interesse. Nesta seção podem ser inseridos links para materiais digitais, indicações de livros, entre outros.

---

<sup>1</sup><https://colab.research.google.com/>

No ensino de programação, a resolução de **exercícios** é uma parte muito importante do processo. Quanto mais treino o estudante tiver, mais facilidade ele terá na resolução de outros problemas. Diante disso, é interessante que se tenha uma lista de atividades que contemplem a unidade de conteúdo e considerem as orientações para minimizar as dificuldades cognitivas. No guia também dispomos de orientações para elaboração de exercícios e avaliações.

### 6.1.3 Orientações para Adaptação de Conteúdo, Exercícios e Avaliações

Alunos com deficiência cognitiva ou neurodiversos podem ter dificuldades em algumas habilidades como abstração e resolução de problemas. Por outro lado, é igualmente importante pensar nas potencialidades que a neurodiversidade pode proporcionar. Por exemplo, estudos apontam que estudantes com dislexia possuem agilidade na resolução de problemas, destacando a programação como um campo que pode potencializar seus pontos fortes e mitigar suas fraquezas [29; 87].

Indivíduos com autismo e TDAH podem desenvolver hiper foco, uma característica na qual concentram intensa atenção em assuntos de interesse. Essa característica pode ser benéfica para o aprendizado da programação, se o aluno manifestar interesse. Então o papel do professor pode ser de estimular essa inclinação por meio de abordagens envolventes, como a integração de jogos, caso o aluno aprecie essa forma de interação [19].

#### Adaptação do Conteúdo

Para adaptarmos um roteiro de aula, utilizamos princípios do Design Universal para Aprendizagem [72]. Além deste, também utilizamos as Recomendações do W3C para acessibilidade Cognitiva [83], bem como também buscamos informações relevantes no Manual Diagnóstico de Transtornos Mentais [6], documento norteador sobre os tipos de condições neurológicas que podem afetar as pessoas bem como suas características.

Focando em alunos neurodivergentes e pensando nas especificidades do ensino de Programação, temos recomendações que mais se encaixam, são algumas delas:

- Oferecer ao estudante **múltiplas formas de apresentação da informação** - Pode-se, por exemplo, oferecer ao aluno o roteiro de estudos na forma de **áudio e vídeo**; criando

o roteiro usando o Google Collab temos a flexibilidade de inserir trechos de código que podem ser testados pelo aluno em tempo real; outra forma de apresentação pode ser a **utilização de imagens ou diagramas**;

- Oferecer ao estudante **esclarecimentos sobre itens específicos de vocabulário e símbolos ou fórmulas matemáticas** - Em Programação temos o uso de muitos termos em inglês, que não é nossa língua nativa, podendo gerar problemas no entendimento. É importante deixar claro o significado dos termos sempre que necessário. Outra estratégia utilizada em Programação são exemplos utilizando fórmulas matemáticas. Diante disso, também é importante explicar ou deixar uma explicação de fácil acesso para o estudante consultar caso necessário.
- Oferecer ao estudante **pontos de conexão com assuntos aprendidos anteriormente** - Alunos neurodivergentes podem possuir dificuldades de memorização e como o conhecimento de Programação é cumulativo. É importante que se lembre sempre que possível o que já foi visto antes e que será necessário para a unidade de conteúdo atual;
- **Facilitar o entendimento de textos**: textos longos e cheios de expressões complicadas podem desestimular alunos neurodivergentes. é importante que essa parte do roteiro de estudo, sendo mais teórica, seja elaborada recorrendo a quebras no texto, como parágrafos mais curtos, uso de figuras, tabelas, destaques como negrito ou outra estratégia que deixe a leitura mais confortável para esse público-alvo;
- Complementar o texto com **outras formas de apresentação**: ao longo da parte teórica, é importante que se acrescente exemplos de código para o aluno ter noção de como aquele conceito seria aplicado na prática no Algoritmo. O uso do Google Collab é interessante, ao permitir que o aluno teste o código em tempo real naquele ponto específico do roteiro. Dessa forma, esse recurso pode possibilitar que o estudante demonstre mais atenção;
- Buscar, ao longo do texto, fazer **ligações com o mundo real e com exemplos mais concretos** de uso daquele conteúdo: os alunos neurodivergentes podem ter dificuldade com abstração. Então, buscar recursos de realizar a ligação entre o abstrato e o concreto, pode auxiliar o aluno a compreender melhor como poderia aplicar os conceitos

de Programação fora da sala de aula;

- Alunos neurodivergentes podem apresentar dificuldades na aprendizagem, como, por exemplo, a dificuldade em entender o que é lido. Muitas vezes em Programação, os professores usam um grande texto para o aluno extrair dali informações para criar um algoritmo que resolva aquele problema. Como esses textos são elaborados pode se tornar um empecilho para que o aluno neurodivergente consiga entender o que se busca. Deve-se sempre prezar pelo **texto simples e literal** dentro do possível, pois uma característica desse público é a dificuldade em entender as coisas no sentido figurado.

### Adaptação de Exercícios

Exercícios são estratégias importantes para praticar os conceitos aprendidos. No ensino de programação, o uso de listas de exercícios é amplamente utilizado já que, quanto mais algoritmos o aluno constrói, mais experiência ele ganha para resolver outros. Pensando nas especificidades de programação, podemos adaptar os exercícios considerando-se as seguintes recomendações:

- **A lista de exercícios deve ter um cabeçalho**, contendo informações importantes sobre a execução das atividades. Podemos informar aqui, por exemplo, qual será a ferramenta a ser utilizada, a linguagem de programação, além de um primeiro **exercício resolvido** para o aluno poder consultar quando precisar e servir de base para os outros. Alunos **autistas** por exemplo, tendem a se sentir mais confortáveis com a previsibilidade e a rotina. Diante disso, um cabeçalho pode ajudar nesse sentido. Já para o aluno com **TDAH**, o cabeçalho pode ser um recurso para não o deixar ansioso sem saber o que está por vir.
- Enunciados objetivos, buscando um **texto mais simples e com instruções claras**. O uso de negrito em palavras-chave pode ser uma boa estratégia para chamar atenção no texto. Para os **autistas**, pode ser mais difícil entender analogias que não estejam literalmente explicadas, então deve-se ter cuidado com isso. Para alunos com **TDAH**, o negrito pode ser uma boa estratégia para chamar atenção para os pontos-chave.
- Cada questão deve ter o **enunciado, exemplos de entrada de dados e exemplos de**

**saída de dados correspondentes.** Isso ajuda o aluno neurodivergente a saber por onde começar.

- **Exercícios em grupo** são interessantes, porém espera-se que grupos de até três componentes deixe o estudante neurodiverso mais confortável. Um grupo muito grande pode deixar o neurodivergente ansioso e disperso.
- Nas disciplinas de programação é comum fazermos atividades do tipo **projetos**, onde existe um problema grande a ser resolvido criando-se um programa. Nesse caso, é importante que a entrega seja **dividida em etapas**, dessa forma o aluno irá ser condicionado a se focar em apenas uma parte por vez, diminuindo a chance de ele buscar fazer várias atividades ao mesmo tempo, característica do neurodivergente que possui hiperatividade.
- Estabelecer **limites de tempo mínimo e máximo** previamente pode ser interessante para que o aluno não desista de resolver as questões na primeira dificuldade.

### **Adaptação de Avaliações**

No caso de atividades avaliativas, as **recomendações para exercícios também se aplicam**, mas além delas temos algumas específicas:

- É importante que todos os **critérios de avaliação e instruções de realização** fiquem claros e sejam colocados no **cabeçalho da folha**, para o aluno poder ler quantas vezes forem necessárias, já que durante a avaliação ele pode esquecer deles.
- **Mencionar claramente a pontuação** de cada questão ajuda o aluno a não ficar ansioso e se organizar melhor.
- Da mesma forma que dividimos a disciplina em unidades de conteúdo, também é interessante **dividir as avaliações** por essas unidades, sendo o aluno avaliado ao final de cada unidade, o que vai ajudá-lo a manter o foco e não se perder no que foi aprendido.
- O professor (ou monitor, ou um profissional leitor) pode **ler as questões para o aluno**, trazendo explicações adicionais oralmente que auxiliem no entendimento.

## 6.2 Validação do guia

Um dos pontos discutidos na EDR é o de envolver os interessados na avaliação da qualidade do design [24]. A partir desse ponto buscamos estratégias para envolver professores de programação e profissionais que apoiam estudantes neurodiversos. Dessa forma, após a criação do guia, submetemos este a uma validação por especialistas utilizando a técnica Delphi.

Para que os instrumentos produzidos nesta pesquisa tenham respaldo científico, precisasse avaliá-lo de alguma forma. Diante da natureza exploratória da pesquisa, foi utilizado o método Delphi [3; 38; 57; 65; 69; 90] para validar o guia produzido e refiná-lo com as considerações obtidas dos avaliadores. Escolheu-se esse método pela necessidade de amparo e consenso fundamentado por especialistas, em relação ao conteúdo e estrutura.

O método Delphi tem sido utilizado cada vez mais em pesquisas na área de Educação, devido a vantagens como que as opiniões de vários especialistas na área que deseja ser pesquisada têm maior validade que opiniões individuais. Além disso, o processo é conduzido de forma anônima e em separado, evitando que os juízes precisem estar fisicamente no mesmo lugar e no mesmo horário [38].

### 6.2.1 Painel de Juízes Especialistas

No método Delphi, é necessário o recrutamento de especialistas na área que se deseja validar. O processo visa que os especialistas cheguem a um consenso em relação ao que é proposto. Existe a recomendação que a quantidade de avaliadores esteja entre 5 e 20 participantes [70]. Nesta pesquisa convidamos dezoito participantes.

Os especialistas, que o método denomina juízes, foram escolhidos em dois perfis: (1) Professores de Disciplinas Iniciais de Programação, que poderiam contribuir com seu conhecimento na elaboração de aulas de Programação e, (2) Profissionais Especializados em Apoio a Pessoas com Deficiência, que dominam técnicas e diretrizes norteadoras para atender necessidades específicas de pessoas com deficiência. Para a avaliação é interessante que se tenha um grupo heterogêneo de especialistas, para observar as várias facetas do domínio do problema [70].

Os juízes foram convidados e alertados que a sua participação seria dada de forma anô-

nima e voluntária. O convite para a pesquisa foi feito por e-mail, sendo enviado o Termo de Consentimento Livre e Esclarecido e, posteriormente, enviados os materiais e formulários para avaliação. A utilização do método Delphi foi executada em três rodadas.

O estudo aconteceu entre os meses de novembro de 2023 e julho de 2024, desde o envio do convite até a finalização da última rodada de avaliações.

Apesar de termos enviado convite para dezoito pessoas, apenas quatorze pessoas aceitaram participar da pesquisa. Entretanto, dessas quatorze apenas treze participaram de fato da primeira rodada.

Do total de treze participantes da primeira rodada, tivemos nove deles professores de programação e quatro profissionais de Pedagogia. Podemos ver nas Tabelas 6.3 e 6.4, o perfil dos juízes especialistas que participaram da primeira rodada de validação.

Dentre os juízes com perfil de professor de programação, temos três com idade na faixa de 35-39 anos, três na faixa de 40-44 anos e três na faixa acima de 45 anos. Sobre a formação, observamos que a maioria tem graduação em Ciência da Computação, com quatro juízes. Além destes, temos três juízes formados em cursos superiores de Tecnologia, sendo um em Análise e Desenvolvimento de Sistemas, um em Telemática e um em Sistemas para Internet. Por fim, temos apenas um juiz formado em Licenciatura da Computação. Sobre a maior titulação temos quatro com Doutorado, quatro com Mestrado e apenas um com Especialização. Já com relação à experiência no ensino de Programação, vemos que temos juízes bastante experientes, sendo a maioria com experiência de 15 anos ou mais, seguida por dois juízes na faixa de 10 a 14 anos e apenas um com experiência de 5 a 9 anos.

Tabela 6.3: Perfil dos Juízes Especialistas - Professores de Programação

<b>Faixa Etária</b>	<b>Graduação</b>	<b>Maior Titulação</b>	<b>Experiência no Ensino de Programação em Anos</b>
40 - 44 anos	Ciência da Computação	Doutorado	15 anos ou mais
35 - 39 anos	Tecnologia em Análise e Desenvolvimento de Sistemas	Mestrado	10 a 14 anos
45 anos ou mais	Ciência da Computação	Mestrado	15 anos ou mais
35 - 39 anos	Licenciatura da Computação	Mestrado	5 a 9 anos
40 - 44 anos	Ciência da Computação	Doutorado	15 anos ou mais
40 - 44 anos	Tecnologia em Telemática	Doutorado	15 anos ou mais
45 anos ou mais	Ciência da Computação	Doutorado	15 anos ou mais
45 anos ou mais	Ciência da Computação	Especialização	15 anos ou mais
35 - 39 anos	Tecnologia em Sistemas para Internet	Mestrado	10 a 14 anos

Já sobre os juízes com perfil de Profissionais, temos que metade está na faixa etária de 40-44 anos e a outra metade na faixa de 25-29 anos. Observamos que todos têm a formação em Pedagogia, sendo a maioria com a maior titulação de Especialização. Sobre a experiência com pessoas neurodivergentes, temos dois juízes com até 4 anos de experiência, um juiz com 5 a 9 anos de experiência e um juiz com 15 anos ou mais.

Tabela 6.4: Perfil dos Juízes Especialistas - Profissionais

Faixa Etária	Graduação	Maior Titulação	Experiência com Pessoas Neurodivergentes
40 - 44 anos	Pedagogia	Especialização	15 anos ou mais
40 - 44 anos	Pedagogia	Especialização	5 a 9 anos
25 - 29 anos	Pedagogia	Graduação	Até 4 anos
25 - 29 anos	Pedagogia	Especialização	Até 4 anos

### 6.2.2 Os formulários de Avaliação

O processo de avaliação foi feito remotamente, utilizando o Google Forms. Cada juiz preencheu um questionário feito em um formulário com questões que remeteram aos itens do guia produzido. Para ser posteriormente calculado o índice de concordância, utilizamos a escala de Likert, que ia de 1 a 5. Onde nota 5 seria a maior concordância com o ponto e a nota 1 a total discordância com o ponto. Essa resposta foi obrigatória para todos os itens avaliados. As orientações sobre a escala de resposta foram as seguintes:

1. **Não adequado:** Nada adequado, não correspondendo em nada ao objetivo que se destina;
2. **Pouco adequado:** 25% adequado, correspondendo muito pouco ao objetivo que se destina;
3. **Moderadamente adequado:** 50% adequado, moderadamente correspondente ao objetivo que se destina;
4. **Muito adequado:** 75% adequado, correspondendo bastante ao objetivo que se destina;
5. **Completamente adequado:** 100% adequado, correspondendo perfeitamente ao objetivo que se destina.

Cada item avaliado correspondia a uma seção do guia. Para avaliação havia uma pergunta referente a cada item, a escala de Likert para assinalar e uma questão aberta não obrigatória

para o juiz que quisesse escrever sobre sua resposta ao item. Em cada pergunta, caso o juiz escolhesse notas de 1 a 3, sugeríamos que fossem feitas as devidas observações sobre o ponto, sugerindo melhorias que o juiz achasse necessário.

### 6.2.3 Análise dos Dados

Para ocorrer a validação dos itens em cada rodada, utilizamos o cálculo do índice de concordância entre os juízes que também pode ser chamado de Índice de Validade de Conteúdo (IVC) [1; 12]. Este cálculo foi realizado segundo a seguinte fórmula:

$$IVC = \frac{TC}{TJ}$$

Onde TC é o total de notas 4 ou 5 e TJ é o número total de juízes que avaliaram o item.

A análise dos itens pela escala de Likert se deu coletando-se as respostas, calculando-se o IVC de cada uma e verificando o valor. Um item foi considerado validado quando o IVC atingiu no mínimo 0.80, sendo considerado um valor de concordância alta [9].

Além do IVC, também foram coletadas as respostas das questões abertas, que foram analisadas e o que foi considerado pertinente foi absorvido no refinamento do guia.

O detalhamento de cada rodada pode ser visto a seguir.

### 6.2.4 A primeira rodada de Delphi

A primeira rodada teve a participação de treze juízes, sendo nove professores de programação e quatro profissionais de apoio a pessoas neurodiversas.

Como cada perfil iria avaliar de acordo com sua competência, optamos por enviar formulários separados, sendo um para professores de programação e outro para os profissionais de apoio.

Enviamos por Email as informações para dar início a primeira rodada. Esse Email continha as explicações necessárias para a avaliação, o prazo para avaliar o guia e os links para o guia e o formulário de avaliação. O prazo inicial dado foi de 15 dias, que precisou ser estendido em alguns casos. Enviamos e-mail de lembretes para auxiliar os avaliadores a não esquecerem de fazer a avaliação.

A rodada foi encerrada, os dados foram tabulados e analisados consoante a sua natureza. Os dados em escala de Likert foram tabulados e analisados para o cálculo do IVC.

Consideramos que os itens que obtivessem IVC maior ou igual a 0.80 foram considerados aceites.

As Tabelas 6.5 e 6.6 apresentam o resultado da avaliação dos juízes para cada item avaliado do guia na primeira rodada. Observamos que após a avaliação na primeira rodada, dentre todos os doze itens avaliados, apenas um deles teve o IVC acima de 0.80 para os juízes professores e seis deles para os juízes profissionais.

Tabela 6.5: Avaliações dos Juízes Professores de Programação na Primeira Rodada de Delphi

Item Analisado	Escala Likert					IVC
	1	2	3	4	5	
Explicação do Guia			4	2	3	0.56
Plano de Aula						
Habilidades usadas nesse conteúdo		1	1	3	4	0.78
Orientações para dificuldades cognitivas		1	4		4	0.44
Roteiro Teórico de Estudo		1	1	2	5	0.78
Exercícios Adaptados						0.56
Roteiro de Aula						
Unidade de conteúdo?				1	8	1.00
O que queremos aprender nesta unidade			2	1	6	0.78
O que vamos precisar			4		5	0.44
Roteiro para Estudo		1	2	1	5	0.67
Resumo do que foi apresentado na unidade			2	3	4	0.78
Para saber mais			2	2	5	0.78
Exercícios para Praticar	1	1	3	1	3	0.44

Tabela 6.6: Avaliações dos Juízes Profissionais na Primeira Rodada de Delphi

Item Analisado	Escala Likert					IVC
	1	2	3	4	5	
Explicação do Guia				1	3	1.00
Plano de Aula						
Habilidades usadas nesse conteúdo				1	3	1.00
Orientações para dificuldades cognitivas				1	3	1.00
Roteiro Teórico de Estudo			1		3	0.75
Exercícios Adaptados			1		3	0.75
Roteiro de Aula						
Unidade de conteúdo?			1	2	1	0.75
O que queremos aprender nesta unidade			1		3	0.75
O que vamos precisar				1	3	1.00
Roteiro para Estudo			1	2	1	0.75
Resumo do que foi apresentado na unidade					4	1.00
Para saber mais					4	1.00
Exercícios para Praticar	1			1	2	0.75

A parte qualitativa aconteceu com a análise das opiniões e sugestões enviadas pelos juízes, sendo realizadas as devidas modificações no guia, quando cabível.

Sobre o item **explicação do guia**, apenas os juízes professores utilizaram a questão aberta para apontar melhorias. A maioria sugeriu mostrar exemplos de uso do guia. Também foi mencionado deixar o texto mais direto, sem tratar extensivamente de leis e explicações sobre as deficiências.

Sobre o item **habilidades usadas nesse conteúdo**, apenas professores deram sugestões. Foi sugerido que se alterasse o nome da seção de “Habilidades Usadas nesse Conteúdo” para “Habilidades Trabalhadas nessa Unidade”. Além disso, foi sugerido que se colocasse mais claramente as habilidades conforme o conteúdo.

Sobre o item **orientações para dificuldades cognitivas**, que ficava no plano de aula, foi apontado que deveriam ter exemplos de aplicação. Foi sugerido também pelos juízes

professores que se colocasse esse item fora do plano de aula.

Sobre o item **roteiro teórico de estudo**, foi sugerido pelos juízes professores que se tivessem orientações mais claras de como construir o roteiro. Já os juízes profissionais sugeriram que se inserisse um modelo de roteiro no próprio guia.

Sobre o item **exercícios adaptados**, os juízes professores sugeriram mostrar exemplos de aplicação e os juízes profissionais também.

Sobre o item **unidade de conteúdo**, foi sugerido pelos juízes profissionais, que fosse acrescentado que as informações devem utilizar uma linguagem simples, curta e direta.

Sobre o item **o que queremos aprender nesta unidade**, foi sugerido pelos juízes professores que se incluísse no guia uma seção de contextualização. Já os juízes profissionais questionaram sobre adaptação do currículo do estudante. Como existe a possibilidade em alguns casos de um estudante com deficiência ter o currículo adaptado, por exemplo, o estudante só estudaria apenas as unidades de conteúdo que ele tivesse condições para isso. Diante disso, esse item precisaria contemplar apenas o que tivesse no currículo também.

Sobre o item **o que vamos precisar**, foi sugerido pelos juízes professores apresentar em forma de lista o que for colocar, bem como links para tutoriais de uso de ferramentas.

Sobre o item **roteiro para estudo**, foi dito pelos juízes profissionais que se entenderam as dicas de adaptação, porém não conseguiam entender o conteúdo. Já os professores mencionaram sentirem falta de contextualização e mais imagens.

Sobre o item **resumo do que foi apresentado na unidade**, foi sugerido pelos professores que poderia ser colocado em estrutura de tópicos e que poderiam ter exemplos de código.

Sobre o item **para saber mais**, foi sugerido pelos professores links de conteúdo que não fossem pagos.

Sobre o item **exercícios para praticar**, foi dito pelos juízes profissionais que não localizaram esse item no guia, e os professores não identificaram adaptações nos exercícios.

Além das questões abertas sobre os itens, ao final do formulário perguntamos aos juízes sugestões de outras seções que poderiam ser interessantes de serem incorporadas ao guia. Os juízes professores sugeriram uma seção de contextualização, bem como seções de orientações gerais acerca das adaptações.

Por fim, também pedimos aos juízes para deixar considerações finais sobre este guia apresentado. Nas Tabelas 6.7 e 6.8 vemos o que foi dito por eles sobre o guia na primeira

rodada. Para identificarmos os juízes, utilizamos a letra J seguida da letra P para professor e A para profissional de apoio, bem como um número de ordem.

Tabela 6.7: Considerações Finais dos Juízes Professores Primeira Rodada de Delphi

<b>ID</b>	<b>Consideração</b>	<b>Perfil</b>
JP1	“No guia está claro. Mas a maior dificuldade que temos é encontrar exemplos de como fazer exercícios adaptados (não acredito que seja falta de material de estudo).”	Professor
JP2	“O guia é teoricamente bem construído, carecendo apenas de um cenário explicativo que evidencie a aplicação de cada etapa para o profissional.”	Professor
JP3	“O guia está bem elaborado e bem explicado.” “Talvez por causa da minha expectativa ao considerar o público-alvo, senti falta de algo mais voltado (Imagens, vídeos, exemplos...) para o estudante com alguma deficiência, principalmente de abstração, compreensão, interpretação e concentração.” “Parabéns pela pesquisa! Posso estar errada, mas acredito que o número de estudantes que possuem alguma deficiência é muito maior do que os números oficiais.”	Professor
JP4	“Considero viável e necessário. Eu apenas gostaria de ver algo que possa servir como exemplo de aplicação no Guia Genérico. Para mim, é difícil saber, por exemplo, quais são as orientações necessárias para adicionar em um roteiro de aula meu para este público.”	Professor
JP5	“Creio que o artefato educacional está bem rico e pode apoiar os professores de programação na adaptação ou criação de materiais mais inclusivos. Parabéns pelo trabalho!”	Professor
JP6	“Falando um pouco do roteiro. Apesar dessa proposta inicial focar em um roteiro teórico, mas acredito que poderia se pensar em algum tipo de interatividade para tentar prender mais a atenção do estudante no roteiro.”	Professor
JP7	“Interessante e com boa perspectiva de uso.”	Professor
JP8	“Material bem elaborado, precisando de ajustes finos apenas.”	Professor
JP9	“Nenhuma, neste momento.”	Professor

Tabela 6.8: Considerações Finais dos Juízes Professores Primeira Rodada de Delphi

<b>ID</b>	<b>Consideração</b>	<b>Perfil</b>
JA1	“O guia será de importante valia para nortear os profissionais sobre como auxiliar na aprendizagem de pessoas neurodivergentes de forma mais efetiva e eficaz. É importante também que para o roteiro ser elaborado o Plano de Ensino Individualizado (PEI) do aluno já esteja em vigor para dar mais objetividade ao que será ensinado para o indivíduo.”	Profissional
JA2	“Parabéns! Excelente trabalho, importantíssimo para que as pessoas neurodivergentes possam ter um melhor desenvolvimento.”	Profissional
JA3	“A proposta é de bastante relevância, pois nós como professores, ao trabalhar com crianças neurodivergentes se sentimos um pouco desorientados sobre a questão das adaptações de materiais, então com um material de apoio o ensino e a aprendizagem dessas crianças passa a ser positiva, com as metas e as habilidades consolidadas.”	Profissional
JA4	“Gostei bastante, pois sou professora e a cada dia chegam mais crianças que necessitam de um guia como esse.”	Profissional

O guia refinado foi enviado novamente para os juízes especialistas na segunda rodada, excluindo-se perguntas sobre as seções aceitas sem modificações segundo o critério de concordância IVC, sendo solicitado a eles novos apontamentos ou confirmação das alterações realizadas.

### **6.2.5 Segunda Rodada de Delphi**

A segunda rodada foi realizada entre fevereiro e março de 2024. Enviamos para os juízes um novo formulário do Google Forms por e-mail, bem como o link para o guia reformulado.

Dos treze participantes que participaram da primeira rodada, onze responderam a se-

gunda rodada. Sendo oito do perfil Professores de Programação e três do perfil Profissionais de Apoio a pessoas neurodiversas.

No formulário da segunda rodada, apresentamos aos juízes um gráfico com o IVC para cada item avaliado na primeira rodada, bem como um compilado com as sugestões de reformulações obtidas na primeira rodada. Nessa fase, os juízes podiam tomar decisões amparados pelas observações uns dos outros, não perdendo o anonimato.

A segunda versão do guia, foi ajustada conforme o sugerido pelos juízes. As sugestões acatadas estão apresentadas nas Tabelas 6.9 e 6.10.

Tabela 6.9: Sugestões dadas pelos Juízes Professores de Programação na Primeira Rodada de Delphi

<b>Item Analisado</b>	<b>Alterações</b>
<b>Explicação do Guia</b>	exemplos, uso de imagens, exemplos de materiais
<b>Plano de Aula</b>	
<b>Habilidades usadas nesse conteúdo</b>	Mudar para “Habilidades trabalhadas nesta unidade”, menos explicações
<b>Orientações para dificuldades cognitivas</b>	exemplo de aplicação, retirar do template para o plano de aula, indicar habilidades afetadas
<b>Roteiro Teórico de Estudo</b>	exemplo de aplicação
<b>Exercícios Adaptados</b>	exemplificar estratégias
<b>Roteiro de Aula</b>	
<b>O que queremos aprender nesta unidade</b>	habilidades que serão trabalhadas
<b>O que vamos precisar</b>	colocar links ou tutoriais
<b>Roteiro para Estudo</b>	contextualizar, exemplo melhor
<b>Resumo do que foi apresentado na unidade</b>	estrutura de tópicos
<b>Para saber mais</b>	link direto para o conteúdo, retirar opções pagas
<b>Exercícios para Praticar</b>	exemplos de código

Tabela 6.10: Sugestões dadas pelos Juízes Profissionais na Primeira Rodada de Delphi

<b>Item Analisado</b>	<b>Alterações</b>
<b>Plano de Aula</b>	
<b>Roteiro Teórico de Estudo</b>	modelo de roteiro de estudos para nortear os professores
<b>Exercícios Adaptados</b>	dicas de adaptação
<b>Roteiro de Aula</b>	
<b>O que queremos aprender nesta unidade</b>	se o currículo do aluno neurodivergente estiver adaptado? esse item será adaptado também?
<b>O que vamos precisar</b>	colocar links ou tutoriais
<b>Roteiro para Estudo</b>	contextualizar, exemplificar melhor
<b>Resumo do que foi apresentado na unidade</b>	estrutura de tópicos

Na segunda rodada, cada item possuía novamente a escala de Likert e espaço para o juiz escrever considerações sobre o item.

Após o recebimento das avaliações desta rodada, os dados foram tabulados e realizado novo cálculo do IVC.

As Tabelas 6.11 e 6.12 apresentam o resultado da avaliação dos juízes para cada item avaliado do guia na segunda rodada.

Tabela 6.11: Avaliações dos Juízes Professores de Programação na Segunda Rodada de Delphi

Item Analisado	Escala Likert					IVC
	1	2	3	4	5	
Explicação do Guia			1	4	3	0.88
Plano de Aula						
Habilidades trabalhadas nessa unidade			3	3	2	0.63
Roteiro Teórico de Estudo				4	4	1.00
Exercícios Adaptados			3	4	1	0.63
Roteiro de Aula						
O que queremos aprender nesta unidade			1	2	5	0.88
O que vamos precisar				3	5	1.00
Roteiro para Estudo	1		3	2	2	0.50
Resumo do que foi apresentado na unidade				4	4	1.00
Para saber mais				2	6	1.00
Exercícios para Praticar		1	1	3	3	0.75
Orientações para dificuldades cognitivas	1			5	2	0.88

Tabela 6.12: Avaliações dos Juízes Profissionais na Segunda Rodada de Delphi

Item Analisado	Escala Likert					IVC
	1	2	3	4	5	
Explicação do Guia				1	2	1.00
Plano de Aula						
Habilidades trabalhadas nessa unidade				1	2	1.00
Roteiro Teórico de Estudo				1	2	1.00
Exercícios Adaptados					3	1.00
Roteiro de Aula						
O que queremos aprender nesta unidade				1	2	1.00
O que vamos precisar					3	1.00
Roteiro para Estudo				1	2	1.00
Resumo do que foi apresentado na unidade				1	2	1.00
Para saber mais				1	2	1.00
Exercícios para Praticar			1		2	0.67
Orientações para dificuldades cognitivas					3	1.00

A parte qualitativa aconteceu com a análise das opiniões e sugestões enviadas pelos juízes, sendo realizadas as devidas modificações no guia, quando cabível.

Sobre o item **explicação do guia**, foi sugerido pelos professores ajustar o título do template para “Template de Roteiro de Aula por Unidade de Conteúdo”.

Sobre o item **habilidades exploradas nessa unidade**, os professores sugeriram que tivessem exemplos práticos. Além disso, questionaram se não seria algo da disciplina e não de cada unidade de conteúdo.

Sobre o item **roteiro teórico de estudo**, os juízes professores elogiaram as orientações dadas para criação do roteiro.

Sobre o item **exercícios adaptados**, os juízes sugeriram ter orientações específicas para os diferentes tipos de neurodiversidade. Também foi sugerido colocar no guia que “caso o estudante não possua um profissional leitor” o professor pode ler as questões para ele. Foi sugerido também retirar a palavra “Adaptados” deixando só exercícios. Foi apontado pelos

juízes profissionais que seria importante o professor saber se o estudante possui a habilidade cognitiva necessária para a aquisição do conteúdo.

Sobre o item **o que queremos aprender nesta unidade**, foi apontado pelos juízes apenas correções no texto, como, por exemplo, um juiz apontou que “não seria trivial deduzir o que é estado do programa”.

Sobre o item **o que vamos precisar**, foi sugerido pelos juízes professores colocar sugestões de ferramentas com menos propagandas, para não contribuir com a dispersão do estudante neurodiversos. Além disso, colocar os hiperlinks para as ferramentas.

Sobre o item **roteiro para estudo**, foi sugerido pelos juízes deixar o roteiro de estudo executável, utilizando, por exemplo, o Google Collab. Na primeira rodada foi enviado o exemplo do roteiro no Collab, mas no intuito de facilitar para os avaliadores, inserimos um exemplo de roteiro no próprio guia.

Sobre o item **resumo do que foi apresentado na unidade**, foi sugerido o uso de mapas mentais quando cabível.

Sobre o item **exercícios para praticar**, foi dito pelos juízes para adequar o título. Assim o título foi alterado no guia, se tornando “Exercícios e Exemplos de Código”. Também foi sugerido colocar links para repositórios de códigos de exemplo.

Sobre o item **Orientações**, seguindo as sugestões da primeira rodada, o texto foi retirado do template do plano de aula para uma seção após os templates. Um dos juízes não havia entendido o porquê de ter saído e por isso deu nota 1.

Além das questões abertas sobre os itens, ao final do formulário novamente perguntamos aos juízes sugestões de outras seções que poderiam ser interessantes de serem incorporadas ao guia. Os juízes professores sugeriram refletir mais sobre exercícios, e sugeriram trazer links para outros ambientes para o aluno executar o código.

Por fim, também pedimos aos juízes para deixar considerações finais sobre o guia apresentado nesta rodada. Nas Tabelas 6.13, 6.13 e 6.15 vemos o que foi dito por eles sobre o guia na segunda rodada.

Tabela 6.13: Considerações Finais dos Juizes Professores JP1 a JP6 da Segunda Rodada de Delphi

<b>ID</b>	<b>Consideração</b>	<b>Perfil</b>
JP1	“As modificações efetuadas trouxeram melhorias, em especial, para os exercícios e avaliações.”	Professor
JP2	“Eu gostei muito do guia, achei objetivo e os exemplos + explicações detalhadas (ao final) nos ajudam a entender todo o contexto para poder realizar o artefato.”	Professor
JP3	“Gostei muito da evolução entre a primeira rodada e a segunda. Apenas considero que talvez a separação de orientações do template possa dificultar o consumo do artefato.”	Professor
JP4	“Não acho interessante colocar o tempo para realizar um exercício, isso pode gerar gatilhos para os estudantes que não atingirem o objetivo. Na seção "Exemplo de código para lembrar", incluir um texto dizendo o que é feito no algoritmo-exemplo: "Lembre-se de como fazer a leitura de dados"”.	Professor
JP5	“O roteiro evoluiu bastante em relação ao anterior. Acredito que está caminhando em direção a um consenso.”	Professor
JP6	<p>“Para facilitar, uma vez que o "Guia Genérico" tem muita informação. Talvez seja mais interessante separar em arquivos, por exemplo:</p> <ul style="list-style-type: none"> <li>- Template para Plano de Aula</li> <li>- Exemplo de Plano de Aula</li> <li>- Template para Roteiro de Estudo</li> <li>- Plano de Aula;</li> <li>- Roteiro para Estrutura Condicional;</li> <li>- Exercício Adaptado;</li> <li>- ... ”</li> </ul> <p>“Parabéns pelo trabalho! Espero ter contribuído.”</p>	Professor

Tabela 6.14: Considerações Finais dos Juízes Professores JP7 e JP8 da Segunda Rodada de Delphi

ID	Consideração	Perfil
JP7	<p>“Gostaria de parabenizar-lhes pela iniciativa de preparar este artefato educacional e dizer que ele está bem interessante, embora possa ser continuamente aprimorado.”</p> <p>“Sugiro que possam rever algumas frases que poderiam ser quebradas, como o trecho: "... podendo gerar problemas no entendimento então... "que pode ficar "... entendimento. Sendo assim, é importante... ". Outros dois trechos cujos ajustes são semelhantes são os seguintes: "... desestimular alunos neurodivergentes, é importante..."; "... ponto específico do roteiro, dessa forma temos... "; "... Recursividade, nesse caso, agregar... ". Ajustar pontuação também, acrescentando vírgula em frases como esta: "Se ele quiser chegar mais rápido, podemos imaginar... ""</p> <p>“Acredito que ter no template do plano de aula o item "Exercícios Adaptados" onde no conteúdo está apenas "Exemplos a seguir" fica um pouco estranho. Acredito que não precisaria vir no template do plano, assim como o item "Roteiro Teórico de Estudo", ou então talvez esses itens precisem ser melhor explicados pensando em quem vai criar o plano de aula e que terá de incluí-los ao seguir as orientações.”</p> <p>“Os exemplos de Plano de Aula e Roteiro estão interessantes, mas acredito que como na parte de "Roteiro para Estudo" ficou mais a parte de "Divisão inteira" e uso do operador %, sem nenhum exemplo da unidade "Estrutura Condicional" em si, ficou um pouco estranho. Sugiro que se mude a unidade para focar em "Divisão inteira e resto da divisão inteira", ou que na parte de Roteiro para Estudo se inclua algo sobre "Estrutura Condicional".”</p>	Professor
JP8	“Muito bem elaborado. Parabéns!”	Professor

Tabela 6.15: Considerações Finais dos Juízes Profissionais na Segunda Rodada de Delphi

<b>ID</b>	<b>Consideração</b>	<b>Perfil</b>
JA1	“É uma ótima proposta, principalmente para ajudar os profissionais da educação.”	Profissional
JA2	“Acredito que o roteiro colaborará e muito com os alunos neurodivergentes, principalmente no que diz respeito a organização e planejamento tanto das ideias, como dos estudos.”	Profissional
JA3	“Parabéns pelo excelente trabalho, tema de extrema importância que está sendo abordado e trabalhado com excelência.”	Profissional

### 6.2.6 Terceira Rodada de Delphi

A terceira rodada foi realizada entre junho e julho de 2024. Enviamos para os juízes um novo formulário do Google Forms por e-mail, bem como o link para o guia reformulado.

Dos onze participantes que participaram da segunda, sete responderam a terceira rodada, sendo todos do perfil Professores de Programação.

No formulário da terceira rodada, apresentamos aos juízes um gráfico com o IVC para cada item avaliado na segunda rodada, bem como um compilado com as sugestões de reformulações obtidas na segunda rodada. Nessa fase os juízes podiam tomar decisões amparados pelas observações uns dos outros, não perdendo o anonimato.

A terceira versão do guia, foi ajustada conforme o sugerido pelos juízes. As sugestões acatadas estão apresentadas na tabela 6.16.

Tabela 6.16: Sugestões dadas pelos Juízes Professores de Programação na Segunda Rodada de Delphi

<b>Item Analisado</b>	<b>Alterações</b>
<b>Plano de Aula</b>	
<b>Habilidades trabalhadas nesta unidade</b>	Mudar para “Habilidades exploradas nesta unidade”; exemplificar sobre como inserir problemas mais simples e ir aumentando a complexidade deles;
<b>Exercícios Adaptados</b>	exemplificar estratégias
<b>Roteiro de Aula</b>	
<b>O que queremos aprender nesta unidade</b>	habilidades que serão trabalhadas
<b>O que vamos precisar</b>	colocar links ou tutoriais
<b>Roteiro para Estudo</b>	contextualizar, exemplo melhor
<b>Resumo do que foi apresentado na unidade</b>	estrutura de tópicos
<b>Para saber mais</b>	link direto para o conteúdo, retirar opções pagas
<b>Exercícios para Praticar</b>	exemplos de código

Na terceira rodada cada item possuía novamente a escala de Likert e espaço para o juiz escrever considerações sobre o item.

Após o recebimento das avaliações desta rodada os dados foram tabulados e feito novo cálculo do IVC.

A Tabela 6.17 apresenta o resultado da avaliação dos juízes para cada item avaliado dos instrumentos na terceira rodada.

Tabela 6.17: Avaliações dos Juízes Professores de Programação na Terceira Rodada de Delphi

Item Analisado	Escala Likert					IVC
	1	2	3	4	5	
Plano de Aula						
Habilidades exploradas nessa unidade			2	2	3	0.71
Exercícios Adaptados			2	3	2	0.71
Roteiro de Aula						
Roteiro para Estudo			1		6	0.86
Exercícios para Praticar			2	3	2	0.71

Sobre **habilidades exploradas nessa unidade**, os juízes sugeriram ajustes no texto para deixar claro que as habilidades seriam as necessárias para especificamente esta unidade de conteúdo.

Sobre **exercícios**, os juízes apontaram ser necessário diferenciar a seção de exercícios do plano de aula da seção do roteiro de aula. Também foi apontado que se colocassem links para as orientações que estão no final do guia.

Sobre **roteiro para estudo**, os juízes apontaram ser importante usar links no documento.

Sobre **exercícios para praticar**, os juízes apontaram ser importante verificar a questão do tempo mínimo e máximo no exemplo. Outro ponto apontado por um juiz foi ele se sentir inseguro em saber quais neurodivergências são contempladas nas atividades, ao ficar na dúvida se teria que criar um exercício diferente para cada neurodivergência.

O método Delphi aponta serem executadas rodadas até que se obtenha o consenso ou até que as avaliações apresentem estabilidade, onde as respostas não gerem mais novidades para ajustes. Diante disso, encerramos a validação na terceira rodada. Considerando que não houve mais sugestões novas que demandassem uma nova rodada, e todos os itens avaliados tiveram um IVC acima de 0.70, que também pode ser considerado um índice aceitável [9].

## 6.3 Estudo com estudantes

Faz parte da Educational Design Research, a intervenção de uso dos instrumentos com estudantes. Desta forma, temos além da visão dos especialistas, a visão dos estudantes. Diante disso, organizamos uma etapa de intervenção através do uso de um roteiro de aula de programação adaptado utilizando os instrumentos propostos nesta tese.

Esta intervenção foi projetada partindo do objetivo **OBJ6 - Avaliar o uso dos instrumentos, na prática.**

### 6.3.1 Participantes

Por termos um público-alvo bastante específico, não é fácil encontrar estudantes que se enquadram no estudo. Diante disso, conseguimos nesse momento que apenas um estudante aceitasse participar.

O estudante é aluno do curso de Ciência da Computação da UFCG, está cursando pela primeira vez a disciplina de Programação I, no semestre 2024.1. Ele se enquadra no perfil, ao possuir diagnóstico de Transtorno do Espectro Autista (TEA) e Transtorno do Déficit de Atenção e Hiperatividade (TDAH).

Conforme informações da coordenação de curso, a disciplina possui cem alunos, destes apenas dois entraram por cotas relacionadas à neurodiversidade. Como a pesquisa é voluntária, apenas um dos dois alunos se dispôs a participar e se declarou neurodiverso. O outro estudante, apesar de ingressar na universidade pela cota de pessoas com deficiência, preferiu não ser identificado como tal. Diante disso, a razão por termos feito o estudo com apenas um estudante, mas que por ser metade do total de estudantes neurodiversos da turma, pode ser considerado representativo.

### 6.3.2 Materiais e Métodos

Este estudo objetivou avaliar o uso dos instrumentos pedagógicos adaptados, na perspectiva do aluno neurodiverso de uma disciplina introdutória de programação. O projeto desta intervenção foi elaborado para acontecer em duas etapas: (1) Disponibilizar um roteiro adaptado utilizando o guia, para alunos neurodiversos no ambiente de uma disciplina de programação introdutória simultaneamente; (2) Avaliar a eficácia do roteiro para esses alunos.

Por ocorrer uma greve, houve um atraso no início das aulas do semestre, prejudicando a execução do experimento. Diante disso, apenas a parte 1 do experimento foi executada e a parte 2 será executada como trabalho futuro a esta tese.

A disciplina de Programação I é a disciplina de programação introdutória, onde o estudante irá ter o primeiro contato do curso com programação.

A disciplina já é normalmente dividida em unidades de conteúdo, onde para cada unidade existe a aula teórica, a prática em laboratório e um miniteste. A avaliação é feita através dos minitests. O aluno realiza o miniteste de uma unidade até que tenha êxito, portanto ele só avança para o próximo quando consegue êxito no miniteste anterior.

### 6.3.3 Protocolo

Para a execução do experimento, foi primeiramente contactado o estudante, que aceitou participar do estudo, assinando o Termo de Consentimento Livre e Esclarecido e sendo explicado todo o processo da pesquisa.

Em seguida, foi disponibilizado para ele um roteiro adaptado, da unidade de conteúdo “Estrutura Condicional”. Esse roteiro foi construído utilizando o guia desenvolvido nessa pesquisa, no Google Collab e está disponível online <sup>2</sup>.

Após o aluno ter acesso ao roteiro, ele respondeu um questionário, onde tivemos em vista saber as impressões do aluno sobre o roteiro.

Como o roteiro é dividido em seções, o questionário tinha uma pergunta sobre cada uma delas, para saber se o aluno entendeu a finalidade da seção, com opção de resposta sim ou não. Além disso, também tinha uma pergunta aberta, para o estudante escrever sua opinião sobre a seção.

Por se tratar da parte do roteiro onde o professor encaixa a parte teórica do conteúdo, a seção “Roteiro de Estudo”, teve uma pergunta onde foi colocada uma lista de afirmativas para o estudante marcar todas as que mais se encaixavam na sua opinião. São elas:

- A forma que o conteúdo está exposto no roteiro é adequada.
- As figuras utilizadas ajudam no entendimento do roteiro.

---

<sup>2</sup>[https://colab.research.google.com/drive/1-vKLY991sGvE9yEonlu\\_yCaX1jl2bKlj?usp=sharing](https://colab.research.google.com/drive/1-vKLY991sGvE9yEonlu_yCaX1jl2bKlj?usp=sharing)

- As tabelas utilizadas ajudam no entendimento do roteiro.
- O tamanho da fonte é adequado.
- Eu consigo localizar partes do conteúdo facilmente.
- O formato do roteiro é interessante para mim.
- Não consegui compreender o conteúdo por esse roteiro.
- Outros.

Outra seção onde foram colocadas afirmativas foi a seção de “Exercícios”. As afirmativas foram:

- Os exercícios foram suficientes para estudar.
- Eu consegui entender os exercícios sugeridos.
- Gostaria que o roteiro tivesse mais exercícios.
- Não consegui entender os exercícios do roteiro.
- Outros.

#### **6.3.4 Análise e Resultados**

A análise foi feita com as respostas do aluno, onde podemos observar que o contato com o roteiro foi considerado por ele positivo.

O aluno respondeu ao questionário, mostrando suas impressões sobre cada parte do roteiro. As respostas abertas do aluno estão apresentadas na Tabela 6.18.

Sobre o roteiro, podemos observar que o aluno ficou satisfeito com o material, apenas sendo reforçado mais de uma vez que o tamanho da letra utilizado não estava adequado. Apontamento esse que será considerado para ajustes do próximo roteiro.

Sobre o conteúdo teórico do roteiro, o estudante assinalou os seguintes itens:

- A forma que o conteúdo está exposto no roteiro é adequada.
- As figuras utilizadas ajudam no entendimento do roteiro.

- As tabelas utilizadas ajudam no entendimento do roteiro.
- Eu consigo localizar partes do conteúdo facilmente.
- O formato do roteiro é interessante para mim.
- Outros (“faltou mais um ou dois exemplos”).

Esse resultado reforça a questão do tamanho da fonte, que ele não achou adequado e mostra a necessidade de se inserir uma maior quantidade de exercícios no roteiro.

Tabela 6.18: Respostas do Aluno ao Questionário sobre o Roteiro de Aula - Estrutura Condicional

Pergunta	Observações
Fale sobre as suas <b>dificuldades no conteúdo dessa unidade</b>	“a sintaxe de if, elif, e else”
Qual a sua opinião sobre a parte <b>“Qual é nossa unidade de conteúdo?”</b>	“muito bom, foi bem claro mas falta ser maior a letra”
Qual a sua opinião sobre a parte <b>“O que queremos aprender nesta unidade”</b>	“muito bo também deixa bem claro as funções e sintaxe das três condicionais”
Qual a sua opinião sobre a parte <b>“Contextualizando”</b>	“muito bom, mas falta ser maior a letra tbm”
Qual a sua opinião sobre a parte <b>“O que vamos precisar”</b>	“ótimo, a mesma coisa da letra mas de resto Ta impecável”
Qual a sua opinião sobre a parte <b>“Resumo do que foi apresentado na unidade”</b>	“mesma coisa do tamanho da letra mas Ta ótimo”
Qual a sua opinião sobre a parte <b>“Exercícios”</b>	“faltou mais um exercícios”

O estudo feito apenas com um estudante e um roteiro não é suficiente para afirmar com certeza a eficácia do material, mas nos dá indícios de que o uso dos instrumentos propostos

nesta tese é promissor e pode ser de grande auxílio para os professores, oferecendo instrumentos para melhorar a qualidade do ensino de programação com foco na neurodiversidade.

### **6.3.5 Ameaças a Validade**

Considerando que toda pesquisa apresenta ameaças à validade, identificamos alguns pontos nesta investigação. O número total de participantes na validação seja uma limitação, embora esteja nos parâmetros exigidos pelo método Delphi. Outra possível limitação é o fato de o roteiro enviado aos alunos ter sido elaborado pela própria pesquisadora, o que se justifica pela falta de estudos prévios com professores criando roteiros a partir do guia. Além disso, a quantidade de alunos que teve acesso ao roteiro foi limitada, impedindo a generalização dos resultados para outros participantes com diferentes tipos e níveis de comprometimento cognitivo. Por fim apontamos os fatores humanos, que podem afetar o processo de triagem de entrevistas e análise de conteúdo, pois o mesmo pesquisador definiu, realizou e analisou a pesquisa.

## **6.4 Considerações Finais do Capítulo**

Este capítulo teve em vista responder à questão de pesquisa QP3: “Como tornar o ensino de programação inclusivo para pessoas neurodivergentes?”. Para isso apresentamos o projeto e construção de instrumentos pedagógicos, na forma de um guia, destinado aos professores de programação, para que esses possam adaptar seus materiais de aula para atender necessidades específicas do público neurodiverso. Na Seção 6.1 foi apresentada a construção dos instrumentos, como o guia foi projetado e construído. Apresentamos as seções do guia, bem como as orientações para os professores poderem utilizá-lo.

Na Seção 6.2 mostramos o processo de validação dos instrumentos utilizando o método Delphi, como foi o recrutamento dos especialistas, bem como a coleta e análise das avaliações. Através dessa validação pudemos fazer o refinamento.

Por fim, a Seção 6.3 apresentamos o primeiro experimento realizado com um estudante da UFCG, tendo contato com um roteiro de aula gerado a partir do guia. Pudemos observar que o impacto para o estudante está sendo positivo.

# Capítulo 7

## Considerações Finais

Neste capítulo, apresentamos as considerações finais desta tese de doutorado. Na Seção 7.1, tratamos das conclusões. Na Seção 7.2, apresentamos as contribuições geradas pela pesquisa e como podem ser aplicadas. Por fim, na Seção 7.3, apresentamos as possibilidades de continuidade da pesquisa.

### 7.1 Conclusões

Esta tese constituiu um esforço de Pesquisa em Educação em Computação buscando discutir e dar suporte a inclusão de pessoas com deficiência cognitiva ou neurodiversas no ensino e aprendizagem de programação. A investigação foi pautada em torno de três questões de pesquisa:

- **QP1** - Quais são os possíveis impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação?
- **QP2** - Quais as percepções das dificuldades enfrentadas por estudantes neurodiversos e professores no contexto da Programação introdutória?
- **QP3** - Como tornar o ensino de programação inclusivo para pessoas neurodivergentes?

Seis objetivos específicos foram definidos com o intuito de responder às questões definidas, a saber:

- **OBJ1** - Catalogar as habilidades que os estudantes precisam desenvolver para programar e como essas habilidades podem ser influenciadas pelas características dos estudantes com deficiência cognitiva;
- **OBJ2** - Investigar como se dá atualmente o processo de ensino e aprendizagem de programação no contexto de estudantes neurodiversos;
- **OBJ3** - Mapear informações sobre abordagens e práticas gerais recomendadas para o público neurodiverso;
- **OBJ4** - Investigar como os materiais de aula de programação poderiam ser adaptados, associando competências às necessidades específicas do comprometimento cognitivo;
- **OBJ5** - Elaborar instrumentos pedagógicos para criação de aulas de programação adaptado para estudantes neurodiversos;
- **OBJ6** - Avaliar o uso dos instrumentos, na prática.

Diante disso, tivemos em vista responder às questões de pesquisa mediante estudos realizados a partir de pesquisas bibliográficas, pesquisa de campo e análise de especialistas.

### 7.1.1 Implicações dos Achados da QP1

Na QP1, compreendemos quais os impactos que as deficiências cognitivas podem ter nas habilidades envolvidas no ensino de programação. Ao longo da minha trajetória como professora de programação, pude ter a experiência de lecionar para estudantes com variados tipos de deficiência. Cada tipo de deficiência trazia consigo uma demanda diferente de atenção e cuidado por parte do professor e da instituição de ensino.

Em se tratando de alunos com deficiência cognitiva, observamos na prática que existem características específicas que podem gerar impactos no processo de ensino e na aprendizagem de programação. As habilidades necessárias para aprender o conteúdo de programação podem ser impactadas em pessoas neurodiversas. Estas habilidades são resolução de problemas, habilidades matemáticas, abstração, pensamento crítico, compreensão de leitura, conhecimento em inglês, atenção aos detalhes, criatividade e gerenciamento de tempo. Porém, isso não significa que alunos neurodiversos não podem aprender a programar. Inclusive,

podemos observar que os impactos não são apenas negativos, tendo também impactos positivos, como, por exemplo, o detalhismo de pessoas com TEA. Podemos citar também a característica do hiper foco, que pode estar presente tanto no TDAH quanto no TEA, e pode ser um fator que auxilie em centralizar atenção para aprender programação, caso seja do interesse do estudante.

É desafiador oferecer igualdade de oportunidades entre estudantes com e sem deficiência, mas precisamos desses esforços, para planejar estratégias de adaptação para conteúdo, prática e exercícios. Não é uma missão simples, mas é necessária. Precisamos identificar os prováveis problemas sobre as dificuldades apresentadas pelos estudantes para cada conteúdo. Além disso, antecipar os desafios com o ensino tradicional e propor mudanças quando necessário, testando seus resultados com o público-alvo. Temos convicção de que os estudantes com deficiência cognitiva poderão ter igualdade de oportunidades para ter acesso ao conhecimento que envolve programação, realizadas as devidas adaptações. Ao considerar a inclusão, daremos a atenção devida aos estudantes que precisam de suporte permanente, mas também todos os estudantes da turma serão beneficiados.

### **7.1.2 Implicações dos Achados da QP2**

Já que a inclusão de estudantes neurodiversos nas instituições de ensino de todos os níveis é uma realidade, buscamos na QP2 compreender como acontece uma situação real onde existam estudantes neurodiversos em disciplinas introdutórias de programação. Diante disso, buscamos em uma instituição de ensino técnico e superior, estudantes desse perfil que estivessem cursando a primeira disciplina de programação. A partir daí, ouvimos os relatos desses alunos, suas dificuldades e percepções acerca do processo. Além dos alunos, ouvimos os respectivos professores da disciplina de programação, para saber como eles, de uma área técnica, estão trabalhando a inclusão dos alunos.

Este estudo retratou as percepções de estudantes neurodiversos e seus professores ao ensinar programação. As principais dificuldades relatadas pelos professores foram a necessidade de treinamento e comunicação mais adequados. Do ponto de vista dos estudantes, nosso estudo destacou as principais dificuldades em entender a aplicação da programação, interpretar problemas, criar algoritmos sem assistência e lembrar o que fazer em exercícios e avaliações.

Diante disso tivemos informações importantes de ambos os perfis, onde pudemos esclarecer e perceber que a inclusão no ensino de programação é uma demanda real e urgente no Ensino de Computação, que existem diversas lacunas que precisam ser preenchidas, beneficiando tanto os alunos quanto os professores.

### 7.1.3 Implicações dos Achados da QP3

Reconhecendo a necessidade de promover a inclusão de pessoas neurodiversas em aulas de programação, abordamos a QP3 investigando maneiras de auxiliar nesse processo. Durante a pesquisa, identificamos que muitos professores de programação não possuem a capacitação adequada para apoiar estudantes neurodiversos em suas aulas. Com isso em mente, desenvolvemos instrumentos pedagógicos que oferece aos professores uma forma de adaptar seus materiais didáticos ou criar novos, incluindo exercícios e avaliações, permitindo favorecer a aprendizagem desses estudantes em suas aulas.

## 7.2 Contribuições

A principal contribuição desta tese é promover a inclusão de pessoas neurodiversas no ensino de programação por meio da criação de **instrumentos pedagógicos para criação de aulas de programação adaptado para estudantes neurodiversos**. Esses instrumentos foram desenvolvidos para ajudar professores a adaptar suas aulas de programação para esse público-alvo. Como parte desse recurso, elaboramos um guia para a criação de aulas de programação inclusivas, contendo templates para planos de aula e roteiros. Além disso, o guia oferece orientações para a adaptação de conteúdo teórico, exercícios e avaliações. Validado por especialistas, esse material tem o potencial de impactar positivamente tanto os professores de programação, que terão à sua disposição um guia para tornar suas aulas mais inclusivas, quanto os alunos neurodiversos, que se beneficiarão de materiais didáticos que atendam às suas necessidades específicas.

Como contribuição complementar, trouxemos a **discussão sobre como as habilidades necessárias para aprender programação são impactadas em pessoas com deficiência cognitiva**, observamos não só os impactos negativos, mas também discutimos os positivos, trazendo para a comunidade a reflexão sobre o assunto.

Outra contribuição desta tese é apresentar as **percepções de alunos neurodiversos e de professores de programação sobre o processo atual de ensino e aprendizagem**, de forma que observamos como os professores entendem que necessitam de capacitação adequada e amparo da instituição de ensino para receber esses alunos em suas aulas. Do ponto de vista dos estudantes, o processo de aprender programação pode ser desafiador sem o devido apoio.

Ao longo do desenvolvimento desta pesquisa, as contribuições apontam resultados significativos para a comunidade acadêmica, divulgados mediante publicações em conferências bem-conceituadas, a saber:

- Um Mapeamento Sistemático sobre o Ensino de Programação para Pessoas com Deficiência no Simpósio Brasileiro de Informática na educação (SBIE) 2020 [20].
- A Systematic Literature Review on Teaching Programming to People with Cognitive Disabilities no IEEE Frontiers in Education Conference (FIE) 2021 [21];
- Identifying Programming Skills Impacted in Students with Cognitive Disabilities no IEEE Frontiers in Education Conference (FIE) 2022 [43];
- Perceptions about Teaching Programming in the Neurodiverse Students' Context no IEEE Frontiers in Education Conference (FIE) 2024 (Aceito, mas só será publicado em outubro de 2024).

Além das publicações, tivemos a honra de ser premiados com o **prêmio Google Awards For Inclusion 2022** através do projeto Including People with Cognitive Disabilities in Programming Teaching <sup>1 2</sup>.

Essas publicações e prêmio destacam a relevância desta pesquisa, bem como sua influência positiva no avanço do conhecimento científico e na inclusão de pessoas com deficiência na área de Computação.

---

<sup>1</sup><https://research.google/programs-and-events/award-for-inclusion-research-program/recipients/?filter=2022>

<sup>2</sup><https://portal.ufcg.edu.br/ultimas-noticias/4059-pesquisa-da-ufcg-ganha-premio-google>  
html

## 7.3 **Trabalhos Futuros**

Esta pesquisa gerou contribuições importantes, e podemos avançar a partir de algumas possibilidades de trabalhos futuros, a saber:

- Continuar o estudo qualitativo com estudantes neurodiversos, para poder avaliar a eficácia de roteiros gerados a partir dos instrumentos, observando seu impacto no desempenho dos alunos na disciplina introdutória de programação;
- Conduzir um estudo qualitativo com professores de programação criando seus materiais de aula inclusivos, a partir dos instrumentos, observando se os roteiros produzidos atenderão as necessidades do público alvo;
- Investigar se o guia consegue atender a todas as limitações ou deficiências em específico;
- Realizações de novos experimentos em outras instituições;
- Observar a viabilidade dos instrumentos na criação de roteiros de aula inclusivos para outras disciplinas da área de Computação;

Diante das considerações, esperamos que este estudo seja base para aumentar o interesse dos pesquisadores pelo tema da inclusão no processo de ensino e aprendizagem de programação, bem como de outras áreas da Computação. As iniciativas voltadas para a diversidade são extremamente importantes, promovendo a todos os tipos de estudantes, igualdade de oportunidades.

# Bibliografia

- [1] Neusa Maria Costa Alexandre and Marina Zambon Orpinelli Coluci. Validade de conteúdo nos processos de construção e adaptação de instrumentos de medidas. *Ciência I& Saúde Coletiva*, 16(7):3061–3068, Jul 2011.
- [2] M.A. et al. Almaiah. Exploring the critical challenges and factors influencing the e-learning system usage during covid-19 pandemic. *Education and Information Technologies*, 25(1):5261–5280, 2018.
- [3] Marcelo Moreira Antunes. Técnica delphi: metodologia para pesquisas em educação no brasil. *Revista de Educação PUC-Campinas*, 19(1):63–71, abr. 2014.
- [4] J. Scott Armstrong. Principles of forecasting: A handbook for researchers and practitioners, 2002.
- [5] Abdu Arslanyilmaz, Margie Briley, Gregory Boerio, Katie Petridis, and Ramlah Ilyas. The effects of an accessible computing curriculum for students with autism spectrum disorders. In *2023 IEEE Frontiers in Education Conference (FIE)*, pages 1–6, 2023.
- [6] American Psychiatric Association. Diagnostic and statistical manual of mental disorders, fifth edition, 2013. (Accessed on 02/03/2022).
- [7] Sasha Barab and Kurt Squire. Design-based research: Putting a stake in the ground. *Journal of the Learning Sciences*, 13(1):1–14, 2004.
- [8] Laurence Bardin. *Análise de Conteúdo*. Edições 70, Janeiro 2015.
- [9] G. Bauer, M. W. I& Gaskell. *Pesquisa qualitativa com texto: imagem e som: um manual prático*. Editora Vozes, Janeiro 2015.

- [10] Samuel Montané Baños, Brenda C. Lara Rubio, Israel Durán Encinas, and Jonathan G. Soto Muñoz. Experience in teaching programming related subjects to university students with asperger syndrome. In *2022 International Conference on Inclusive Technologies and Education (CONTIE)*, pages 1–5, 2022.
- [11] Rachel Bonnette, Samuel Abramovich, Adrienne Decker, and Greg A. Fabiano. Building ecosystems of belonging for neurodiverse students: A discussion of instructor practices and training needs. In *Proceedings of the 53rd ACM Technical Symposium on Computer Science Education V. 2, SIGCSE 2022*, page 1192, New York, NY, USA, 2022. Association for Computing Machinery.
- [12] Lígia Maria Presumido BRACCIALLI, Ana Carla BRACCIALLI, Mauro AUDI, and Marcia SCHERER. Tradução e adaptação cultural de instrumentos para avaliar a predisposição do uso de tecnologia assistiva que constitui o modelo <i>matching, person and technology</i>. *Revista Brasileira de Educação Especial*, 25(2):189–204, Apr 2019.
- [13] Ulla H Graneheim Britt-Marie Lindgren, Berit Lundman. Abstraction and interpretation during the qualitative content analysis process - pubmed, 2020.
- [14] A.C. et al Calderon. A double dissociative study into the effectiveness of computational thinking. *Education and Information Technologies*, 2020.
- [15] Jose Aires Castro Filho, Raquel Santiago Freire, and Dennys Leite Maia. *Estudo de Caso como método de pesquisa em Informática na Educação*, volume 3, chapter 1. SBC, Porto Alegre, 2021.
- [16] S. et al. Collings. Unpacking the complexity of planning with persons with cognitive disability and complex support needs. *Journal of applied research in intellectual disabilities : JARID*, 31(1):142–151, 2018.
- [17] Cristina Conchinha, Patrícia Osório, and João Correia de Freitas. Playful learning: Educational robotics applied to students with learning disabilities. In *2015 International Symposium on Computers in Education (SIIE)*, pages 167–171, 2015.

- 
- [18] T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms, fourth edition*. MIT Press, 2022.
- [19] Felipe Fernandes da Silva, Linnyer Beatrys Ruiz Aylon, and Daniela Eloise Flôr. Teaching computational thinking to a student with attention deficit through programming. In *2020 IEEE Frontiers in Education Conference (FIE)*, pages 1–9, 2020.
- [20] Elaine Cristina Juvino De Araújo and Wilkerson L. Andrade. Um mapeamento sistemático sobre o ensino de programação para pessoas com deficiência. In *Anais do XXXI Simpósio Brasileiro de Informática na Educação*, pages 1713–1722, Porto Alegre, RS, Brasil, 2020. SBC.
- [21] Elaine Cristina Juvino De Araújo and Wilkerson L. Andrade. A systematic literature review on teaching programming to people with cognitive disabilities. In *2021 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, 2021.
- [22] Sociedade Brasileira de Computação. Diretrizes para ensino de computação na educação básica. (Accessed on 09/19/2024).
- [23] Milena de França Monteiro, Maria Verónica Segovia González, Akynara Aglaé Rodrigues Santos da Silva Burlamaqui, and Aquiles Medeiros Filgueira Burlamaqui. Educational robotics as a motivational tool for students with asd: A first impression. In *2020 Latin American Robotics Symposium (LARS), 2020 Brazilian Symposium on Robotics (SBR) and 2020 Workshop on Robotics in Education (WRE)*, pages 1–6, 2020.
- [24] Jan Van den Akker et al. Educational design research, 2006. (Accessado em 06/04/2024).
- [25] Kurt Eiselt and Paul Carter. Integrating social skills practice with computer programming for students on the autism spectrum. In *2018 IEEE Frontiers in Education Conference (FIE)*, pages 1–5, 2018.
- [26] Menna Elshahawy, Khaled Aboelnaga, and Nada Sharaf. Codaroutine: A serious game for introducing sequential programming concepts to children with autism. In *2020 IEEE Global Engineering Education Conference (EDUCON)*, pages 1862–1867, 2020.

- [27] Menna Elshahawy, Mariam Bakhaty, and Nada Sharaf. Developing computational thinking for children with autism using a serious game. In *2020 24th International Conference Information Visualisation (IV)*, pages 761–766, 2020.
- [28] M. Fikret Ercan and Dennis Sale. Teaching programming: An evidence based and reflective approach. In *2020 IEEE REGION 10 CONFERENCE (TENCON)*, pages 997–1001, 2020.
- [29] Norman Powell et al. Dyslexia and learning computer programming. *Innovation in Teaching and Learning in Information and Computer Sciences*, 3(2):1–12, 2004.
- [30] Evelina Fedorenko, Anna Ivanova, Riva Dhamala, and Marina Umaschi Bers. The language of programming: A cognitive perspective. *Trends in Cognitive Sciences*, 23(7):525–528, 2019.
- [31] José Figueiredo and Francisco José García-Peñalvo. Building skills in introductory programming. In *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM’18, page 46–50, New York, NY, USA, 2018. Association for Computing Machinery.
- [32] José Figueiredo and Francisco José García-Peñalvo. Teaching and learning strategies of programming for university courses. In *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, TEEM’19, page 1020–1027, New York, NY, USA, 2019. Association for Computing Machinery.
- [33] Christos Galeos, Kostas Karpouzis, and George Tsatiris. Developing an educational programming game for children with adhd. In *2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA)*, pages 1–6, 2020.
- [34] N.W. Gelbar, I. Smith, and B. Reichow. Systematic review of articles describing experience and supports of individuals with autism enrolled in college and university programs. *J Autism Dev Disord*, 44:2593–2601, 2014.
- [35] Antonio Carlos Gil. *Métodos e técnicas de pesquisa social*. Atlas, 2019.
- [36] Antonio Carlos Gil. *Como Elaborar Projetos de Pesquisa*. Atlas, 2022.

- [37] Claudia Roberta Araújo GOMES, Alex Sandro; GOMES. Classificação dos tipos de pesquisa em informática na educação, 2020.
- [38] Megan M. Grime and George Wright. *Delphi Method*, pages 1–6. John Wiley I& Sons Inc., 2016.
- [39] Zhao Guanghui, Lyu Yanjun, Tian Yixiao, Wang Zhaoxia, and Zou Chengming. Case-based teaching organization for python programming that focuses on skill training. In *2018 13th International Conference on Computer Science I& Education (ICCSE)*, pages 1–5, 2018.
- [40] Murray Turoff Harold A. Linstone. *The delphi method: Techniques and applications*, 2002.
- [41] Carl Haynes-Magyar. Neurodiverse programmers and the accessibility of parsons problems: An exploratory multiple-case study. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2024, page 491–497, New York, NY, USA, 2024. Association for Computing Machinery.
- [42] Maya Israel, Gakyung Jeong, Meg Ray, and Todd Lash. Teaching elementary computer science through universal design for learning. In *Proc. of SIGCSE 20*, pages 1220–1226, 2020.
- [43] Elaine Cristina Juvino de Araújo, Wilkerson L. Andrade, and Ana Liz Souto Oliveira. Identifying programming skills impacted in students with cognitive disabilities. In *2022 IEEE Frontiers in Education Conference (FIE)*, pages 1–8, 2022.
- [44] Reyes Juárez-Ramírez, Samantha Jiménez, and Verónica Tapia-Ibarra. What is programming? putting all together, part ii –the cognitive skills associated. In *2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pages 55–64, 2019.
- [45] Reyes Juárez-Ramírez, Christian X. Navarro, Verónica Tapia-Ibarra, Ricardo Macías-Olvera, and César Guerra-García. What is programming? putting all together - a set of skills required. In *2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT)*, pages 11–20, 2018.

- [46] Charles Adams; Alastair Campbell; Rachael Bradley Montgomery; Michael Cooper; Andrew Kirkpatrick. Web content accessibility guidelines (wcag) 2.2, Agosto 2020.
- [47] Barbara Kitchenham and Stuart Charters. Guidelines for performing systematic literature reviews in software engineering, July 2007.
- [48] V.F. et al. Knight. Teaching digital, block-based coding of robots to high school students with autism spectrum disorder and challenging behavior. *Journal of Autism and Developmental Disorders*, 49(3):3113–3126, 2019.
- [49] Varsha Koushik. Making block-based programming accessible to people with cognitive disabilities. *SIGACCESS Access. Comput.*, 1(126), mar 2020.
- [50] Varsha Koushik and Shaun K. Kane. “*It Broadens My Mind*”: *Empowering People with Cognitive Disabilities through Computing Education*, page 1–12. Association for Computing Machinery, New York, NY, USA, 2019.
- [51] Richard E. Ladner and Maya Israel. "for all" in "computer science for all". *Commun. ACM*, 59(9):26–28, aug 2016.
- [52] I-Jui Lee. Applying the game mode and teaching strategies of computational thinking to the improvement of social skills training for children with autism spectrum disorders. In Margherita Antona and Constantine Stephanidis, editors, *Universal Access in Human-Computer Interaction. Applications and Practice*, pages 38–47, Cham, 2020. Springer International Publishing.
- [53] Carla Leitão. *A entrevista como instrumento de pesquisa científica em Informática na Educação: planejamento, execução e análise*, volume 3, chapter 7. SBC, Porto Alegre, 2021.
- [54] Adam Wade Lewis, Vanessa Miller, Ilish Kane, and Matthew Marrazzo. A systematic literature review of cognitive models of the behavior of novice software developers. *J. Comput. Sci. Coll.*, 34(2):208–214, dec 2018.

- [55] Qisheng Li, Josephine Lee, Christina Zhang, and Katharina Reinecke. *How Online Tests Contribute to the Support System for People With Cognitive and Mental Disabilities*, chapter 10, pages 1–15. Association for Computing Machinery, New York, NY, USA, 2021.
- [56] Linda Mannila, Fredrik Heintz, Susanne Kjällander, and Anna Åkerfeldt. Programming in primary education: towards a research based assessment framework. In *Proceedings of the 15th Workshop on Primary and Secondary Computing Education, WiPSCE '20*, New York, NY, USA, 2020. Association for Computing Machinery.
- [57] Joana Brás Varanda Marques and Denise de Freitas. Método delphi: caracterização e potencialidades na pesquisa em educação. *Pro-Posições*, 29(2):389–415, May 2018.
- [58] S. McKenney and T.C. Reeves. *Conducting Educational Design Research*. Taylor I& Francis, 2013.
- [59] Susan McKenney and Thomas C. Reeves. Educational design research: Portraying, conducting, and enhancing productive scholarship. *Medical Education*, 55(1):82–92, 2021.
- [60] Rodrigo Pessoa Medeiros, Geber Lisboa Ramalho, and Taciana Pontual Falcão. A systematic literature review on teaching and learning introductory programming in higher education. *IEEE Transactions on Education*, 62(2):77–90, 2019.
- [61] Erum Mehmood, Adnan Abid, Muhammad Shoaib Farooq, and Naeem A. Nawaz. Curriculum, teaching and learning, and assessments for introductory programming course. *IEEE Access*, 8:125961–125981, 2020.
- [62] Stephen J. Molitor, Joshua M. Langberg, and Steven W. Evans. The written expression abilities of adolescents with attention-deficit/hyperactivity disorder. *Research in Developmental Disabilities*, 51-52:49–59, 2016.
- [63] Makayla Moster, Ella Kokinda, Matthew Re, James Dominic, Jason Lehmann, Andrew Begel, and Paige Rodeghero. "can you help me?"an experience report of teamwork in a game coding camp for autistic high school students. In *2022 IEEE/ACM 44th*

*International Conference on Software Engineering: Software Engineering Education and Training (ICSE-SEET)*, pages 50–61, 2022.

- [64] Anna Nystedt. Designing for cognitive disability how flexichat affects the accessibility of instant messaging. Master’s thesis, UmeåUniversity, Department of Computing Science, 2019.
- [65] Jonathan Osborne, Sue Collins, Mary Ratcliffe, Robin Millar, and Rick Duschl. What “ideas-about-science” should be taught in school science? a delphi study of the expert community. *Journal of Research in Science Teaching*, 40(7):692–720, 2003.
- [66] Mariano et al PIMENTEL. Design science research: fazendo pesquisas científicas rigorosas atreladas ao desenvolvimento de artefatos computacionais projetados para a educação, 2020.
- [67] T. et al Plomp. *Educational Design Research: An Introduction*. Netherlands Institute for Curriculum Development, 2013.
- [68] Natalia Mabel Quintero and Aino Ahtinen. University language instructors programming robotic learning applications: Design and implementation of encouraging programming workshop experiences. In *Proceedings of the 11th International Conference on Human-Agent Interaction*, HAI ’23, page 171–179, New York, NY, USA, 2023. Association for Computing Machinery.
- [69] Luciano Coelho Romera, Eliane Maria Carrit Delgado-Pinheiro, Flávia Rodrigues dos Santos, and Osni Lázaro Pinheiro. Elaboração de um material educativo para subsidiar a prática de professores de educação física no trabalho com alunos com perda auditiva. *Educação*, 46(1):e8/ 1–27, jan. 2021.
- [70] Gene Rowe and George Wright. *Expert Opinions in Forecasting: The Role of the Delphi Technique*, pages 125–144. Springer US, Boston, MA, 2001.
- [71] Johnny Saldana. *The Coding Manual for Qualitative Researchers*. SAGE Publications, 2021.
- [72] Eladio Sebastián-Heredero. Diretrizes para o desenho universal para a aprendizagem (dua). *Revista Brasileira de Educação Especial*, 26(4):733–768, Oct 2020.

- [73] Misbahu Sharfuddeen Zubair, David Brown, Matthew Bates, and Thomas Hughes-roberts. Are visual programming tools for children designed with accessibility in mind? In *Proceedings of the 12th International Conference on Education Technology and Computers, ICETC '20*, page 37–40, New York, NY, USA, 2021. Association for Computing Machinery.
- [74] Yaqing Shi, Song Huang, and Changyou Zheng. Research on programming courses teaching based on blended learning. In *Proceedings of the 5th International Conference on Frontiers of Educational Technologies, ICFET 2019*, page 30–34, New York, NY, USA, 2019. Association for Computing Machinery.
- [75] J. Singer. *Neurodiversity: The birth of an idea*. Judy Singer; 2nd edition (July 3, 2016), 2017.
- [76] David Statter and Michal Armoni. Teaching abstraction in computer science to 7th grade students. *ACM Trans. Comput. Educ.*, 20(1), jan 2020.
- [77] Andreas Stefik, Willlliam Allee, Gabriel Contreras, Timothy Kluthe, Alex Hoffman, Brianna Blaser, and Richard Ladner. Accessible to whom? bringing accessibility to blocks. In *Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. I, SIGCSE 2024*, page 1286–1292, New York, NY, USA, 2024. Association for Computing Machinery.
- [78] Silke Stienen-Durand and Jennifer George. Supporting dyslexia in the programming classroom. *Procedia Computer Science*, 27:419–430, 2014. 5th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion, DSAI 2013.
- [79] Sylvia Stuurman, Harrie J.M. Passier, Frédérique Geven, and Erik Barendsen. Autism: Implications for inclusive education with respect to software engineering. In *Proceedings of the 8th Computer Science Education Research Conference, CSERC '19*, page 15–25, New York, NY, USA, 2019. Association for Computing Machinery.
- [80] Matthew S. Taylor, Eleazar Vasquez, and Claire Donehower. Computer programming

- with early elementary students with down syndrome. *Journal of Special Education Technology*, 32(3):149–159, 2017.
- [81] Rob Thompson. Teaching coding to learning-disabled children with kokopelli’s world. In *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 258–259, 2016.
- [82] Simon Thompson. Where do i begin? a problem solving approach in teaching functional programming | springerlink, 1997. (Accessed on 02/03/2022).
- [83] W3C. Making content usable for people with cognitive and learning disabilities, Julho 2019.
- [84] Jane Waite and Sue Sentance. Teaching programming in schools: A review of approaches and strategies. (Accessed on 12/14/2021).
- [85] Jane Waite and Sue Sentance. Teaching-programming-in-schools-pedagogy-review-raspberry-pi-foundation.pdf. (Accessed on 07/29/2024).
- [86] Wei Wang, Kathy B. Ewoldt, Mimi Xie, Alberto M. Mestas-Nuñez, Sean Soderman, and Jeffrey Wang. Virtual summer camp for high school students with disabilities - an experience report. In *Proceedings of the 54th ACM Technical Symposium on Computer Science Education V. 1*, SIGCSE 2023, page 458–464, New York, NY, USA, 2023. Association for Computing Machinery.
- [87] S. Wille, J. Century, and M. Pike. Computer science principles (csp) and students with learning differences: Expanding opportunities for a hidden underrepresented group. In *In Proceedings of the RESPECT’16*, pages 1–8, 2016.
- [88] S. Wille, J. Century, and M. Pike. Exploratory research to expand opportunities in computer science for students with learning differences. *Computing in Science Engineering*, 19(3):40–50, 2017.
- [89] Robert K. Yin. *Case Study Research: Design and Methods*. Sage Publications, 2013.
- [90] Andrea Cristina Mariano Yoshinaga, Beatriz Oliveira Pereira, Wanderlei Abadio de Oliveira, Isis Pires Goncalves, Miyeko Hayashida, and Marta Angelica Iossi Silva.

Intervenção antibullying proposta por enfermeiros: elaboração e validação pelo método Delphi. *Aletheia*, 51:8 – 20, 12 2018.

## **Apêndice A**

# **Um Mapeamento Sistemático sobre o Ensino de Programação para Pessoas com Deficiência**

## Um Mapeamento Sistemático sobre o Ensino de Programação para Pessoas com Deficiência

Elaine Cristina Juvino de Araújo<sup>1,2</sup>, Wilkerson L. Andrade<sup>2</sup>

<sup>1</sup>Instituto Federal de Educação, Ciência e Tecnologia da Paraíba (IFPB)  
Campina Grande - Paraíba - Brazil

<sup>2</sup>Universidade Federal de Campina Grande (UFCG)  
Campina Grande - Paraíba - Brazil

elainec@ifpb.edu.br, wilkerson@computacao.ufcg.edu.br

**Abstract.** *There are many challenges faced by people with disabilities, among them are the challenges in learning programming. As there is a need to address deficiencies in a specific way in teaching and learning, a systematic mapping was carried out in order to provide an overview of the area. Initially, 1844 works were found, but only 29 met the inclusion and exclusion criteria of articles defined in this work. These works were classified according to their approach. As a result, a research map in the area was created and it was also identified that the greatest efforts have been made for people with visual impairments and that there is no standardized methodology for each type of disability.*

**Resumo.** *Muitos são os desafios enfrentados pelas pessoas com deficiência, entre esses destacam-se os desafios na aprendizagem de programação. Como existe a necessidade de se tratar as deficiências de maneira específica no ensino-aprendizagem, foi realizado um mapeamento sistemático a fim de fornecer uma visão geral da área. Inicialmente foram encontrados 1844 trabalhos, porém apenas 29 satisfizeram os critérios de inclusão e exclusão de artigos definidos. Estes trabalhos foram classificados de acordo com sua abordagem. Como resultado, um mapa de pesquisas na área foi criado e também foi identificado que os maiores esforços têm sido para pessoas com deficiência visual e que não existe uma padronização de metodologia para cada tipo de deficiência.*

### 1. Introdução

Pode-se definir como deficiência qualquer condição incapacitante que acomete as pessoas, seja do tipo permanente como uma síndrome ou temporária como um braço imobilizado por uma fratura. Martin [Martin 2009] destaca que existem vários tipos de deficiência como: déficit intelectual, físico, visual, auditivo ou múltiplo (quando o indivíduo possui mais de um tipo de deficiência), as síndromes (como a de Down) e os transtornos (como o Autismo). No Brasil, segundo o último censo, 23,9% da população declarou possuir algum tipo de deficiência. Essa significativa parcela da população apresenta inúmeros desafios, principalmente no que diz respeito à sua capacitação [IBGE 2010].

Atualmente, um dos maiores desafios na educação, não apenas do Brasil, é fazer com que estudantes com deficiência tenham acesso a ensino adequado (hoje eles são incluídos na escola e excluídos do conhecimento). O material didático geralmente

não foi projetado para esta inclusão, sendo um motivo relevante para o desenvolvimento de ferramentas (digitais) e metodologias que considerem as diferentes necessidades de cada deficiência, mas que seja aplicável dentro de um contexto comum de sala de aula [Martin 2009].

Pessoas com deficiência podem entrar em qualquer curso técnico ou superior que desejar, inclusive cursos da área de Computação. Ao iniciar um curso dessa área, elas irão cursar disciplinas de programação, que compreendem o processo de concepção, criação e teste de algoritmos. Essa área já é considerada complexa e com altos índices de evasão, mesmo para alunos que não possuem deficiência alguma [Souza et al. 2016]. A abordagem utilizada nos cursos de programação envolve, muitas vezes, a exposição de situações-problema nas quais o aluno deverá ser capaz de criar um algoritmo que chegue à solução esperada. Quando o aluno não consegue entender o problema que precisa resolver, dificilmente chegará a acertar o exercício. Gomes e Mendes [Gomes and Mendes 2015] expõem que os métodos de ensino mais tradicionais não são os mais adequados para apoiar a aprendizagem de programação, nas quais se apresentam turmas com número elevado de estudantes e pouca produtividade das aulas expositivas no campo da programação. Neste contexto, pode-se supor que a criação de conteúdos motivadores para a aprendizagem e a utilização de métodos de ensino adequados podem ser importantes para um suporte adequado aos estudantes, em particular aqueles com deficiência.

A área de ensino de programação para pessoas com deficiência ainda necessita de mais iniciativas que auxiliem docentes e discentes no processo de ensino-aprendizagem. Este trabalho mostra os resultados de um mapeamento sistemático focado em descobrir no que e como os pesquisadores estão trabalhando na área de ensino de programação voltada para pessoas com deficiência, o protocolo de pesquisa criado e a forma como foi realizado. O restante do trabalho está organizado da seguinte forma: a Seção 2 apresenta trabalhos relacionados, a Seção 3 descreve a metodologia para realização do mapeamento sistemático, a Seção 4 apresenta os resultados do estudo, na Seção 5 discute-se o que foi observado na pesquisa, incluindo prováveis ameaças à validade. Por fim, as considerações finais estão na Seção 6.

## 2. Trabalhos Relacionados

Al-Ratta e Al-Khalifa [Al-Ratta and Al-Khalifa 2013] executaram uma revisão sistemática de literatura para analisar e classificar trabalhos relacionados ao ensino de programação para pessoas cegas. Os resultados apontaram que a maioria das pesquisas focaram em soluções para permitir que pessoas cegas usassem tecnologias, porém poucas foram as pesquisas que tratavam sobre o ensino de programação para esse público. O mapeamento aqui apresentado se difere desse por ser mais abrangente, visto que o propósito é de identificar o que está sendo feito na área de ensino de programação sem focar em apenas uma deficiência, buscando catalogar as estratégias de ensino-aprendizagem atualmente utilizadas, além de selecionar trabalhos mais recentes entre 2010 e 2020.

Silva *et al.* [Silva et al. 2015] realizaram uma revisão sistemática de literatura sobre ensino-aprendizagem de programação em eventos nacionais, verificando que há uma prevalência de trabalhos voltados apenas para o ensino superior e não abordaram a temática para estudantes com deficiência. O trabalho aqui exposto realizou buscas de artigos apenas em Inglês, nacionais ou não, e não se limitou a um nível de ensino específico.

### 3. Procedimentos Metodológicos

A metodologia escolhida para identificar trabalhos relacionados ao tema foi o mapeamento sistemático de literatura (MSL). Esse método é recomendado para áreas de pesquisa em que existem poucos estudos primários relevantes e de qualidade, identificando lacunas para possíveis trabalhos futuros e problemas em aberto para serem investigados.

Para a execução do mapeamento, foi utilizada a abordagem definida por Petersen *et al.* [Petersen et al. 2008], que consiste em um conjunto de etapas bem definidas bastante difundido em pesquisas na área de Computação. A seguir, são descritas as etapas do processo escolhido e como foram realizadas.

#### 3.1. Definição das Questões de Pesquisa (QP) do Mapeamento Sistemático

A primeira etapa do MSL consiste em identificar as prováveis questões de pesquisa a serem respondidas. A ideia nesse momento inicial é de definir questões que remetem ao problema de pesquisa de uma forma mais geral. Desejou-se com esse MSL identificar trabalhos nos seguintes aspectos: (i) abordagens associadas aos tipos específicos de deficiências; (ii) tecnologias que estão sendo utilizadas na área; (iii) limitações identificadas nas pesquisas recentes; (iv) problemas em aberto na área de adaptação de exercícios de programação para pessoas com deficiência; e (v) como estão sendo feitas as avaliações das pessoas com deficiência em disciplinas de programação.

Para esse trabalho foram identificadas as seguintes QP:

- QP1 - Quais as principais abordagens pedagógicas investigadas no domínio da Acessibilidade em Ensino de Programação?
- QP2 - Em quais níveis educacionais tem sido mais investigada a área de Acessibilidade em Ensino de Programação?
- QP3 - Quais as tecnologias utilizadas na Acessibilidade em Ensino de Programação?
- QP4 - Como os alunos deficientes são avaliados no Ensino de Programação?
- QP5 - Quais os problemas em aberto na Acessibilidade em Ensino de Programação?

#### 3.2. Escopo da Pesquisa

Seguindo os passos definidos por Petersen [Petersen et al. 2008], após a identificação das QP, define-se o escopo da pesquisa. Nesta fase, são definidas a chave de busca e as bases nas quais os trabalhos serão pesquisados.

A primeira definição nesta etapa foi de que a busca seria feita apenas em Inglês e artigos publicados nos últimos 10 anos, de forma a identificar trabalhos mais relevantes e recentes na área. O passo seguinte foi a escolha das bases nas quais seriam localizados os trabalhos. Foram selecionadas as seguintes bases de busca: IEEE, ACM e Springer Link.

A definição da chave de busca é muito importante, pois é através dela que os trabalhos relevantes na área são localizados e catalogados. Foi definido que a pesquisa nas bases se daria apenas no título, resumo e palavras chave. Para esse projeto foi definida a seguinte chave de busca inicial: (“programming” OR “computer science” OR “cs”) AND (“teaching” OR “exercise” OR “students”) AND (“disabilities” OR “autism” OR “blind” OR “deaf” OR “accessibility” OR “for all”). Em uma primeira busca, foram encontrados 444 trabalhos na ACM, 678 na Springer e 722 no IEEE. A etapa seguinte foi a triagem dos artigos que efetivamente seriam relevantes para a pesquisa.

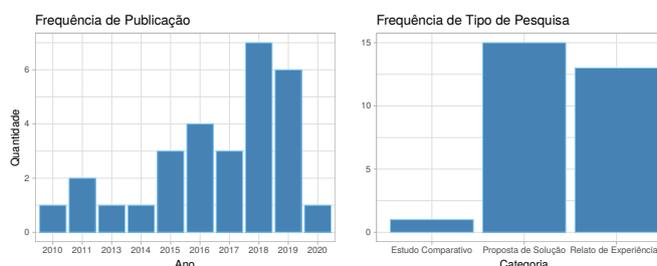


Figura 1. Frequência de (a) publicação e (b) de tipo de pesquisa

### 3.3. Triagem dos Artigos

De forma a buscar os trabalhos que são importantes para as questões de pesquisa, foram definidos alguns critérios de inclusão e exclusão. Como critérios de inclusão temos: CI1) Artigos em Inglês; CI2) Os trabalhos devem conter os termos da chave de busca presentes no título e/ou no resumo e/ou nas palavras-chave do artigo selecionado; CI3) Artigos completos que apresentem método, proposta ou prática de ensino de programação para pessoas com algum tipo de deficiência. Os critérios de exclusão definidos foram: CE1) Artigos que não descrevem um método, proposta ou prática de ensino de programação para pessoas com algum tipo de deficiência; CE2) Artigos com abordagens ou modelos provenientes de uma mesma pesquisa; CE3) Artigos de revisão de literatura.

Feitas as consultas às bases de dados, os trabalhos retornados foram refiltrados após a leitura do título, palavras chave e resumo de cada trabalhos, de forma a observar quais trabalhos atendiam aos critérios de inclusão e exclusão. Dos 1844 trabalhos, 425 foram considerados candidatos por atenderem ao CI1, após uma nova triagem foi feita a leitura dos resumos e reaplicação dos critérios de inclusão e exclusão, sendo selecionados 29 trabalhos ao final, cujos detalhes estão disponíveis online<sup>1</sup>.

## 4. Resultados

Esta pesquisa foi realizada no mês de Junho de 2020. Assim, esta seção apresenta os resultados que foram obtidos até essa data. Mostramos os dados de forma geral e em seguida, respondemos cada questão de pesquisa.

### 4.1. Resultados Gerais

O resultado do mapeamento sistemático realizado apontou que existe interesse em ensino de programação para pessoas com deficiência, conforme mostra a Figura 1(a). O ano com maior número de publicações foi 2018, com a publicação de 7 artigos. Com relação ao tipo de pesquisa realizada, os trabalhos foram catalogados segundo a classificação de Petersen [Petersen et al. 2008], que estabelece os seguintes tipos: relato de experiência, pesquisa de avaliação, pesquisa de validação, proposta de solução, artigo de opinião e artigo filosófico. Como pode-se ver na Figura 1(b), foram encontrados trabalhos em apenas 3 tipos, sendo proposta de solução o tipo mais frequente, seguido por relato de experiências.

<sup>1</sup><https://tinyurl.com/y6sr8sjk>

#### **4.2. QP1 - Quais as principais abordagens pedagógicas investigadas no domínio da Acessibilidade em Ensino de Programação?**

As abordagens identificadas nos artigos catalogados neste mapeamento sistemático foram: Programação Adaptada (16/29 artigos), Programação em Blocos (4/29 artigos), Robótica (5/29 artigos), Programação Tátil ou Desplugada (2/29 artigos), Programação Colaborativa (1/29 artigo) e Jogos (1/29 artigo). Como pode-se ver, a abordagem mais utilizada foi o que chamamos de programação adaptada. Nessa abordagem é utilizada algum tipo de adaptação para o ensino de programação tradicional, por exemplo o uso de um leitor de telas para pessoas cegas ou a tradução de conteúdo para língua de sinais para pessoas surdas.

#### **4.3. QP2 - Em quais níveis educacionais tem sido mais investigada a área de Acessibilidade em Ensino de Programação?**

Na investigação realizada neste trabalho, foi identificado que existem pesquisas no ensino fundamental, médio e superior. Os níveis mais investigados são fundamental (14/29 artigos) e médio (11/29 artigos). Esse resultado pode ser explicado pela maioria dos trabalhos serem originados dos Estados Unidos onde existem iniciativas consistentes de inserção de aprendizagem de programação para alunos antes de entrarem no ensino superior.

Quando consideramos o contexto do nosso país vemos que não é comum ter contato com conceitos de programação antes de entrar em um curso de nível superior, o que coloca diversos alunos, inclusive com deficiência, em um curso específico da área mas com quase ou nenhuma base. Dessa forma, investigar as experiências de alunos com deficiência no ensino superior é uma área que pode ser abordada, já que foi percebida a pouca quantidade de trabalhos relacionados a esse nível de ensino.

#### **4.4. QP3 - Quais as tecnologias utilizadas na Acessibilidade em Ensino de Programação?**

O apoio de tecnologias é bastante utilizado no ensino de programação, porém foi observado no MSL que não existe um padrão quando se trata de pessoas com deficiência. A maioria das tecnologias abordadas foi utilizada por no máximo 2 estudos, o que pode ser visto na Tabela 1. Apenas uma tecnologia fugiu a esse comportamento, que foi o Lego Mindstorms utilizado em 4 trabalhos.

#### **4.5. QP4 - Como os alunos deficientes são avaliados no Ensino de Programação?**

Uma questão delicada no ensino é o processo de avaliação dos alunos, que precisa ser bem planejado para que possa efetivamente mostrar ao professor se de fato o conteúdo foi assimilado. Neste mapeamento, identificamos que ainda não existem métodos de avaliação claros para pessoas com deficiência em ensino de programação. A maioria dos trabalhos utiliza como parâmetro as impressões pessoais dos alunos para identificar se a abordagem utilizada foi bem aceita, o que foi feito através de questionários, entrevistas e observação.

Apenas um trabalho catalogado mostrou uma forma de avaliação diferente, Stefik *et al.* [Stefik et al. 2011] adaptaram a escala inicialmente proposta por Askar e Davenport [Askar and Davenport 2009], que busca medir a autoeficácia em programação Java. No caso do trabalho de Stefik *et al.* [Stefik et al. 2011], a escala foi adaptada de forma trivial para a linguagem Hop, que foi a utilizada no trabalho.

Tabela 1. Tecnologias utilizadas

Tecnologia(s)	#	Referências
ABTB	1	[Konecki et al. 2016]
Adobe Connect, Google Docs	1	[Chiou and Young 2017]
BJC Unit, Code.org	1	[Wille et al. 2017]
Bonk	1	[Kane et al. 2018]
C++	1	[Connelly 2010]
Code.org	2	[Stefik et al. 2019], [Wille et al. 2016]
HTML,CSS e JavaScript	1	[Kearney-Volpe et al. 2019]
JAWS	1	[Pereira et al. 2018], [Connelly 2010]
Pascalzin	1	[Pereira et al. 2018]
LIBRAS	2	[Oliveira et al. 2015], [Granada et al. 2018]
Ruby	1	[Kane and Bigham 2014]
Não definida	1	[Capovilla et al. 2015]
UDL	3	[Israel et al. 2020], [Koushik and Kane 2019], [Ray et al. 2018]
SodBeans+Hop(Quorum)	1	[Ladner and Stefik 2017], [Stefik et al. 2011]
Torino	2	[India et al. 2019], [McMillan and Rodda-Tyler 2016]
Scratch	2	[Koushik and Kane 2019], [Munoz et al. 2018]
StoryBlocks	1	[Koushik et al. 2019]
SuperLOGO	1	[Granada et al. 2018]
Lego Mindstorms	4	[Conchinha et al. 2015], [Ludi and Reichlmayr 2011], [Ludi et al. 2018], [Granada et al. 2018]
Phogo, Phytton	1	[Gonzalez-Sacristan et al. 2016]
Legos	1	[Capovilla et al. 2013]
PAC Mate	1	[Connelly 2010]
Java	2	[Eiselt and Carter 2018], [Ludi and Reichlmayr 2011]

#### 4.6. QP5 - Quais os problemas em aberto na Acessibilidade em Ensino de Programação?

O mapeamento realizado comprovou que existem estudos relevantes na área, porém ainda existe um vasto campo de pesquisa a ser explorado. Alguns problemas em aberto foram identificados: (i) Como avaliar a aprendizagem em programação de um aluno com deficiência? (ii) Como adaptar exercícios de programação para alunos com deficiência? (iii) Qual metodologia de ensino é adequada para esse público?

Todos esses problemas em aberto precisam levar em consideração que cada tipo de deficiência possui suas especificidades, o que leva a crer que, quando da realização de pesquisas na área, esse é um fator de extrema prioridade.

#### 5. Discussões

Identificamos com este MSL que existem diversas pesquisas buscando auxiliar pessoas com deficiência a aprender programação. Como cada tipo de deficiência exige um estudo específico, poucos foram os trabalhos que não restringiram claramente suas pesquisas para um tipo específico de deficiência.

A maioria dos trabalhos utilizou como estudo de caso deficientes visuais e, mesmo para esse público alvo, ainda há muito a ser explorado. Foi observado que para esse tipo de aluno é muito importante a presença de um tutor, de forma a auxiliá-lo nas atividades [Connelly 2010]. No caso de alunos surdos, foi visto que estes possuem mais dificuldade em raciocínio numérico e no conceito de variáveis, sendo um questionamento pertinente analisar quais as melhores formas de abordagem desse tipo de conceito para esses alunos [Granada et al. 2018].

Alunos autistas foram estudados em alguns trabalhos, sendo analisado que associar o ensino de programação ao treinamento de habilidades sociais pode ser benéfico

para eles [Eiselt and Carter 2018].

### 5.1. Abordagens e Adaptações

A maioria dos trabalhos encontrados no mapeamento utilizou o que chamamos de Programação Adaptada, que seria utilizar uma mediação específica para que o aluno com deficiência conseguisse acessar as informações e produzir seus códigos de programação. Por exemplo, para alunos cegos, a ideia da programação auditiva pode auxiliar na construção dos programas [Stefik et al. 2011]. Já para alunos surdos, o uso de adaptação por linguagens de sinais se torna interessante, já que para boa parte das pessoas surdas a língua de sinais é sua primeira língua, mas para isso é necessária a criação de glossários de termos técnicos sinalizados, ou seja, assim como na tradução em qualquer linguagem, deve-se ter o cuidado de traduzir determinado termo associado ao seu contexto. Entretanto, mesmo a língua de sinais pode sofrer com regionalismos e isto deve ser levado em conta quando da criação dos sinais técnicos. Existem iniciativas para criação de sinais internacionais, como o criado colaborativamente no ASL-STEM Forum<sup>2</sup> [Oliveira et al. 2015, Granada et al. 2018].

Alunos com diferenças de aprendizagem foram também analisados e concluiu-se que a barreira mais comum para a aprendizagem de programação foi relacionada à linguagem utilizada [Wille et al. 2017], sendo importante investigar quais adaptações na linguagem técnica poderiam melhorar a assimilação por parte desses alunos. Alunos autistas por exemplo, possuem melhor assimilação com o concreto e o explícito, então Stuurman *et al.* [Stuurman et al. 2019] afirmam que deve-se abordar os conteúdos exigindo-se o mínimo de contexto possível para entender o que se quer dizer.

### 5.2. Replicabilidade de Experimentos

Lung [Lung et al. 2008] afirma que a replicação é um importante fator para verificação de dados empíricos, desempenhando um papel essencial no avanço do conhecimento científico. Entretanto, ele também concluiu que é difícil e em alguns casos até inconveniente executar replicações literais envolvendo seres humanos na área de computação. Observando os trabalhos encontrados nesse MSL identificamos que as características dos estudantes, os diferentes graus de deficiência e o conhecimento dos pesquisadores poderiam influenciar substancialmente os resultados encontrados.

### 5.3. Ameaças à Validade

Esta pesquisa possui ameaças à validade. O conjunto de artigos retornados podem ser afetados pelas bases e pela chave de busca, bem como por fatores humanos na catalogação dos dados. Outro fator de ameaça à validade foi não selecionar artigos de outras línguas. Ao calibrar a chave de busca, percebeu-se que alguns artigos relevantes não utilizavam termos mais genéricos relacionados ao tema, como “*disabilities*”, em seus títulos, palavras-chave ou resumos. Portanto, foram incluídos termos específicos de algumas deficiências que podem causar algum impacto nos resultados.

## 6. Considerações Finais

Aprender programação já é considerado complexo por diversos fatores e quando se trata de alunos com deficiência a preocupação com o processo de ensino-aprendizagem se torna

<sup>2</sup><https://aslstem.cs.washington.edu/>

maior. Neste contexto, este trabalho realizou um mapeamento sistemático de literatura com o objetivo de obter uma visão geral de como os pesquisadores e professores estão trabalhando o ensino de programação para esses alunos.

A maior parte dos trabalhos relatou experiências com ensino de programação para pessoas com deficiência visual, do ensino fundamental e médio e adaptando o uso de materiais e tecnologias com auxílio auditivo de um leitor de tela. Os trabalhos em sua maioria utilizavam abordagens e tecnologias diferentes, o que mostra uma ausência de padronização mesmo se tratando de um mesmo tipo de deficiência. O que vimos de forma recorrente nos artigos catalogados é a real necessidade de se adaptar os métodos de ensino de programação tradicionais para atender os diferentes tipos de alunos.

Como oportunidades para trabalhos futuros identificamos algumas vertentes interessantes como a definição de metodologias de ensino adequadas aos diferentes tipos de deficiência, levando em consideração como promover autonomia desses estudantes, respeitadas às devidas limitações. Outra possibilidade é estudar formas efetivas e sistemáticas de avaliação, de forma a saber de fato quem está aprendendo.

Por último, fazemos os seguintes questionamentos: seria possível ensinar programação para pessoas com deficiência de forma que estes alunos saiam de seus cursos devidamente capacitados para, por exemplo, adentrar o mercado de trabalho? É possível padronizar metodologias de ensino-aprendizagem, tecnologias e avaliações levando em consideração as especificidades de cada tipo de deficiência?

### Referências

- Al-Ratta, N. M. and Al-Khalifa, H. S. (2013). Teaching programming for blinds: A review. In *Fourth ICTA'2013*, pages 1–5.
- Askar, P. and Davenport, D. (2009). An investigation of factors related to self-efficacy for java programming among engineering students. *The Turkish J. of Educ. Tech.*, 8.
- Capovilla, D., Krugel, J., and Hubwieser, P. (2013). Teaching algorithmic thinking using haptic models for visually impaired students. In *Proc. of the LaTiCE'13*, pages 167–171.
- Capovilla, D., Mühlhng, A., and Hubwieser, P. (2015). How learning styles in cs can foster inclusion of visually impaired students. In *Proc. of the LaTiCE'15*, pages 187–192.
- Chiou, P. T. and Young, G. S. (2017). Implementing recommendations of accessibility technology guidelines — the quantitative effects and benefits it offers to non-disabled students. In *Proceedings of the CSCI'17*, pages 1137–1142.
- Conchinha, C., Osório, P., and de Freitas, J. C. (2015). Playful learning: Educational robotics applied to students with learning disabilities. In *Proceedings of the SIE'15*, pages 167–171.
- Connelly, R. (2010). Lessons and tools from teaching a blind student. *J. Comput. Sci. Coll.*, 25(6):34–39.
- Eiselt, K. and Carter, P. (2018). Integrating social skills practice with computer programming for students on the autism spectrum. In *Proc. of the FIE'18*, pages 1–5.
- Gomes, A. and Mendes, A. (2015). À procura de um contexto para apoiar a aprendizagem inicial de programação. *Educação, Formação Tecnologias - ISSN 1646-933X*, 8(1).

- Gonzalez-Sacristan, C., Garcia-Saura, C., and Molins-Ruano, P. (2016). Phogo: A low cost, engaging and modern proposal to learn how to program. In *Proceedings of the IV TEEM '16*, page 67–71.
- Granada, R. P., Barwaldt, R., and Espíndola, D. B. (2018). Glossary of computational terms as a stimulus to programming logic: a case study with deaf students. In *Proceedings of the FIE'18*, pages 1–7.
- IBGE (2010). IBGE | Censo 2010. [Online; accessed 3. Jul. 2020].
- India, G., Ramakrishna, G., Bisht, J., and Swaminathan, M. (2019). Computational thinking as play: Experiences of children who are blind or low vision in india. In *Proceedings of the 21st ASSETS '19*, page 519–522.
- Israel, M., Jeong, G., Ray, M., and Lash, T. (2020). Teaching elementary computer science through universal design for learning. In *Proc. of SIGCSE '20*, page 1220–1226.
- Kane, S. K. and Bigham, J. P. (2014). Tracking @stemxcomet: Teaching programming to blind students via 3d printing, crisis management, and twitter. In *Proceedings of the 45th SIGCSE '14*, page 247–252.
- Kane, S. K., Koushik, V., and Muehlbradt, A. (2018). Bonk: Accessible programming for accessible audio games. In *Proceedings of the 17th IDC '18*, page 132–142.
- Kearney-Volpe, C., Hurst, A., and Fitzgerald, S. (2019). Blind web development training at oysters and pearls technology camp in uganda. In *Proceedings of the 16th W4A '19*.
- Konecki, M., Ivković, N., and Kaniški, M. (2016). Making programming education more accessible for visually impaired. In *Proc. of the 39th MIPRO'16*, pages 887–890.
- Koushik, V., Guinness, D., and Kane, S. K. (2019). Storyblocks: A tangible programming game to create accessible audio stories. In *Proceedings of the 2019 CHI '19*, page 1–12.
- Koushik, V. and Kane, S. K. (2019). It broadens my mind: Empowering people with cognitive disabilities through computing education. In *Proc. of CHI '19*, page 1–12.
- Ladner, R. E. and Stefik, A. (2017). Accessforall: Making computer science accessible to k-12 students in the united states. *SIGACCESS Access. Comput.*, (118):3–8.
- Ludi, S., Bernstein, D., and Mutch-Jones, K. (2018). Enhanced robotics! improving building and programming learning experiences for students with visual impairments. In *Proceedings of the 49th SIGCSE '18*, page 372–377.
- Ludi, S. and Reichlmayr, T. (2011). The use of robotics to promote computing to pre-college students with visual impairments. *ACM Trans. Comput. Educ.*, 11(3).
- Lung, J., Aranda, J., Easterbrook, S., and Wilson, G. (2008). On the difficulty of replicating human subjects studies in software engineering. In *In Proceedings of the 30th ICSE'08*, pages 191–200.
- Martin, C. S. (2009). Os fundamentos das deficiências e síndromes - Aprender e superar. [Online; accessed 3. Jul. 2020].
- McMillan, C. and Rodda-Tyler, A. (2016). Collaborative software engineering education between college seniors and blind high school students. In *Proceedings of the 38th ICSE '16*, page 360–363.

- 
- Munoz, R., Villarroel, R., Barcelos, T. S., Riquelme, F., Quezada, A., and Bustos-Valenzuela, P. (2018). Developing computational thinking skills in adolescents with autism spectrum disorder through digital game programming. *IEEE Access*, 6:63880–63889.
- Oliveira, F. C. d. M., Gomes, G. N., de Freitas, A. T., de Oliveira, A. C., Silva, L. C., and Queiroz, B. (2015). A comparative study of the acceptability of signs for the brazilian sign language created in person and remotely. In *Proc. of SIGCSE '15*, page 207–211.
- Pereira, R. M., da Silva, F. F., and Silla, C. N. (2018). Teaching algorithms for visually impaired and blind students using physical flowcharts and screen readers. In *In Proceedings of the FIE'18*, pages 1–9.
- Petersen, K., Feldt, R., Mujtaba, S., and Mattsson, M. (2008). Systematic mapping studies in software engineering. In *Proceedings of the 12th EASE'08*, page 68–77.
- Ray, M. J., Israel, M., Lee, C. e., and Do, V. (2018). A cross-case analysis of instructional strategies to support participation of k-8 students with disabilities in cs for all. In *Proceedings of the 49th SIGCSE '18*, page 900–905.
- Silva, T., Medeiros, T., Medeiros, H., Lopes, R., and Aranha, E. (2015). Ensino-aprendizagem de programação: uma revisão sistemática da literatura. *Revista Brasileira de Informática na Educação*, 23(01):182.
- Souza, D., Batista, M., and Barbosa, E. (2016). Problemas e dificuldades no ensino de programação: Um mapeamento sistemático. *Revista Brasileira de Informática na Educação*, 24(1):39.
- Stefik, A., Ladner, R. E., Allee, W., and Mealin, S. (2019). Computer science principles for teachers of blind and visually impaired students. In *Proceedings of the 50th SIGCSE '19*, page 766–772.
- Stefik, A. M., Hundhausen, C., and Smith, D. (2011). On the design of an educational infrastructure for the blind and visually impaired in computer science. In *Proceedings of the 42nd SIGCSE '11*, page 571–576.
- Stuurman, S., Passier, H. J., Geven, F., and Barendsen, E. (2019). Autism: Implications for inclusive education with respect to software engineering. In *Proceedings of the 8th CSERC '19*, page 15–25.
- Wille, S., Century, J., and Pike, M. (2016). Computer science principles (csp) and students with learning differences: Expanding opportunities for a hidden underrepresented group. In *In Proceedings of the RESPECT'16*, pages 1–8.
- Wille, S., Century, J., and Pike, M. (2017). Exploratory research to expand opportunities in computer science for students with learning differences. *Computing in Science Engineering*, 19(3):40–50.

## **Apêndice B**

# **A Systematic Literature Review on Teaching Programming to People with Cognitive Disabilities**

# A Systematic Literature Review on Teaching Programming to People with Cognitive Disabilities

Elaine Cristina Juvino de Araújo  
Federal Institute of Paraíba  
Federal University of Campina Grande  
Campina Grande, PB, Brazil  
elainec@ifpb.edu.br

Wilkerson L. Andrade  
Federal University of Campina Grande  
Campina Grande, PB, Brazil  
wilkerson@computacao.ufcg.edu.br

**Abstract**—This Research Full Paper presents a Systematic Literature Review (SLR) that analyzes scientific studies based on teaching programming to people with cognitive disabilities. Teaching programming is naturally challenging, especially when faced with students with different learning levels, including cognitive impairment. Due to the lack of a panorama that contemplates the association between teaching programming and people with cognitive disabilities, we present this SLR to identify which initiatives are available to teach people with cognitive disabilities to programming. This SLR followed the protocol model of a systematic review adapted to cover specific requirements for planning, executing, and presenting results. There are some initiatives to include students with some cognitive impairment found in the literature. However, there are still gaps to explore. The main question answered by this research is to identify what are the efforts and contributions to teaching people with cognitive disabilities to program, looking for related papers in the Springerlink, ACM, and IEEE databases.

**Index Terms**—teaching programming, cognitive disabilities, systematic review

## I. INTRODUCTION

Knowledge in Computer Science (CS) has become increasingly necessary, and most individuals who are trained in this area have access to a reasonable variety of job opportunities [5]. For example, in the United States, there are several efforts to include CS in both elementary and high school curricula to allow students to get in touch with CS concepts at an earlier age [10]. These initiatives are essential to promote the inclusion of students with disabilities, allowing equal access to academic and industry careers [10].

In practice, CS earlier education includes CS concepts, Computational Thinking (CT), programming, and robotics [10]. CT involves solving problems through fundamental concepts of CS [29]. There is evidence that students with cognitive disabilities can learn CT with proper support [17]. Programming crosses all areas of knowledge and can help develop several cognitive skills in the student, to mention: creativity, logical reasoning, critical problem-solving, reflection, interpretation, concentration, among others. Skills developed from the study of programming can cause a significant impact in shaping critical and conscious individuals who can contribute significantly to transforming society [23].

Since elementary school, many efforts have been made to establish programming earlier and earlier in school curric-

ula. Although some initiatives might have been successful in popularizing programming with younger audiences, most forget to consider students who have learning disabilities or attention deficit disorders. Making learning accessible to all students is a significant challenge because teachers need to be appropriately trained to deal with students who learn differently. These students can obtain knowledge by meeting their specific learning needs. Schools seek to improve the educational process by searching for new methodologies, training teachers, and obtaining new technologies [6], [7]. In Brazil, for example, there is a growing increase in enrollments of people with disabilities [9], which leads us to question how to teach programming to students with different ways of learning.

There are several types of cognitive impairments cataloged, such as Autism Spectrum Disorder (ASD) and Attention Deficit Hyperactivity Disorder (ADHD), to name a few. ASD characterizes a deficit in communication and social interaction, behavior patterns, interest, and restricted and repetitive activities. ADHD characterizes a neurodevelopmental disorder that can present harmful levels of inattention, disorganization, and hyperactivity-impulsivity. Inattention and disorganization can involve inability to stay on task, difficulty in paying attention, and missing or misplacing objects at levels that are inconsistent with age or level of development. Hyperactivity-impulsivity can come as excessive activity, restlessness, inability to remain seated, interference and activities of others, and inability to wait - characteristics that may come in excess for the age or the level of development [24].

Game development can be especially beneficial for people with ASD, which is one type of cognitive impairment. Assistive technologies can provide different ways of accessing information, thus guaranteeing a better environment for students with disabilities, including those with learning difficulties [1], [2].

This work aims to determine what and how researchers work in teaching programming for people with cognitive disabilities. To this end, we conducted a Systematic Literature Review (RSL) to answer the question: “what are the existing efforts and contributions to teach people with cognitive disabilities how to code”?

This paper is divided as follows: In Section II, we present

the related work. In Section III, we describe the methodology following the SLR process. We present the results answering the research questions in Section IV. We discuss the results observed during the SLR in Section V. We present the threats to validity in Section VI. Finally, we present the conclusions and pointers for further work in Section VII.

## II. RELATED WORK

There are reviews aimed at teaching programming, but no reviews focused on students with cognitive disabilities. Among the related works, we sought to provide an overview of introductory programming, organizing its results according to the approach: in students, teaching, the curriculum, and assessment. The student approach mentioned a subgroup of underrepresented students, mainly women and non-disabled students. [31].

Another review showed published studies on robotics in teaching programming but focused only on the visually impaired. This work signaled the teaching methodologies and difficulties faced in this scope, but which would not apply to students with cognitive disabilities [32].

Some research was developed to include people with learning disabilities in programming or elementary computing. One paper proposes integrating social skills with the teaching of programming for students with ASD based on the idea that programming requires collaborative efforts. The focus of the course was programming, but there was an interest in taking advantage of the moment so that students could practice applied social skills, such as working in groups, receiving and giving feedback on activities, and knowing how and when to ask for help. The work suggests benefits in this practice but needs more research [25].

Studies believe that block programming can be more interesting for people with learning difficulties, applying in classes with children in elementary school [26], [30]. However, the papers suggest that more research is needed to generate greater student engagement and teacher support.

A study made the cross between the cognitive style of people with autism and the necessary cognitive skills in Software Engineering to generate guidelines for inclusive education. Unfortunately, the generated guidelines are purely speculative, requiring further studies to prove whether they are viable [28].

One paper reports an exploratory study that sought to expand the opportunities for K-12 students who learn differently in teaching computing principles. This research resulted in guidelines for adapting the materials used, being evaluated by feedback from students and teachers involved [6].

## III. METHODOLOGY

An SLR [3] was carried out, seeking to investigate how the inclusion of programming for people with cognitive disabilities is being thought about and carried out. A few steps are required to perform the review: planning the review, conducting the review, and reporting the review.

In planning the review, we need to identify the need for it, define the research questions, and use the protocol to carry

it out. The protocol begins with selecting primary studies, extraction, and synthesis of data in conducting the review. We define the mechanisms for generation and disclosing the information found in the review of the report.

### A. Review Protocol

The RSL protocol defines that a critical question to be answered must be created, and the critical question of this work is to identify “what are the existing efforts and contributions to teach people with cognitive disabilities how to code”? From this question, we derived three research questions (RQ) for identifying the state of the art in this context:

- RQ1: How did teachers approach the content for students with cognitive disabilities?

*The purpose of RQ1 is to verify which approaches at the pedagogical level are being adopted in teaching programming to people with cognitive disabilities.*

- RQ2: What technologies are used to support teaching?

*The objective of RQ2 is to identify which technologies support these approaches within this context.*

- RQ3: How are students evaluated?

*The RQ3 aims to see what is being used to assess the learning of students with cognitive disabilities in teaching programming.*

### B. Conducted Search

Initially, we carry out tests to identify the best terms and bases to be used, which could be within the context of students with some cognitive impairment and programming teaching, arriving at the combination of terms presented in Table I.

TABLE I  
SYNONYMS OR RELATED WORDS

Terms	Synonyms or Related Words for Key-words
Programming-related	“computer science”, “cs”, “programming”
Teaching-related	“teaching”, “learning”, “students”
Terms related to cognitive impairment	“autism”, “cognitive”, “intellectual”, “learning”, “neurodiversity”
Accessibility-related	“disabilities”, “accessibility”

TABLE II  
SUMMARY OF THE STUDIES RETURNED IN EACH DIGITAL LIBRARY

Digital Library	Search Results
ACM	63
IEEE	280
Springer	369
<b>Total</b>	<b>712</b>

The search string chosen was:

**(“computer science” OR “cs” OR “programming”) AND (“teaching” OR “learning” OR “students”) AND (“autism” OR “cognitive” OR “intellectual” OR “learning” OR “neurodiversity”) AND (“disabilities” OR “accessibility”)**

Table II shows the number of studies returned in each digital library. We found seven hundred and twelve screening papers using inclusion (CI) and exclusion (EC) criteria. The databases used were ACM, IEEE, and Springer, and we adapted the search string for each of these databases. The inclusion criteria defined for the selection were:

- (IC1) - Papers in English;
- (IC2) - Papers published in the last 10 years.
- (IC3) - Papers must contain the keywords used in the key; search, in the title and / or in the abstract and / or in the keywords of the selected paper;
- (IC4) - Complete papers that present methodologies, suggestions or practices for teaching programming to people with some type of cognitive impairment.

And the exclusion:

- (EC1) - Repeat papers;
- (EC2) - Review papers.

#### C. Screening of papers for inclusion and exclusion

The screening process is presented in Table III, where we applied the inclusion and exclusion criteria. We read the titles, keywords, and abstracts to select the papers, choosing only those that met the search scope.

The first step was to visualize the number of papers found in the search bases. Even applying a search string, databases often return papers that are not related to the search criteria. Therefore, after applying the inclusion criteria IC1 and IC2, some papers not written in English or not published in the last ten years were discarded.

The second step was to verify which papers had the terms defined in the search string in their title, abstract, or keywords, seeking to remove a good part of the returned papers that were not in the research context.

The third step was to read the titles of the papers, removing those that were not related to the review.

The fourth step was to read the abstracts of the papers, seeking to refine the list of selected ones further.

Finally, the fifth step of the screening process was the complete reading of the twelve papers resulting from the previous steps, of which only eight were considered relevant to the research.

#### D. Snowballing Search

Complementing the automatic review process in the databases, we use snowballing, a search process that involves looking for relevant studies in an paper's reference list or citations, seeking more relevant papers that the automatic search may have excluded. It is challenging to formulate a good search string, as there is often no standard in the terminology used, and the more comprehensive the search terms are, the greater the chances of finding irrelevant papers [12]. Combining search strategies can be an excellent alternative to obtain better results, using snowballing as a primary or secondary search strategy [22].

According to the strategy presented by [12], initially you must select the set of papers to be used as a basis. The basis

TABLE III  
SCREENING OF PAPERS

Steps	Papers
Step 1: Identify and organize the papers found in the search bases by the inclusion criteria IC1 and IC2;	712 papers
Step 2: Refine by inclusion criterion CI3;	241 papers
Step 3: Review of titles removing papers that do not adhere to the theme and / or fit the exclusion criteria;	40 papers
Step 4: Review of abstracts, removing papers that do not adhere to the theme;	12 papers
Step 5: Review of the complete papers removing the papers that do not adhere to the theme;	8 papers

here was the eight papers selected at the end of the screening done in the papers obtained in the automated search.

The snowballing methodology was applied backward from the initial set, identifying the studies that the papers in the set refer to within the investigated theme. At the end of the process, we selected six papers, which were later read and classified, totaling 14 papers for analysis in the review, as can be seen in Table IV.

TABLE IV  
FINAL SELECTION OF PAPERS

Search	Papers
Digital Libraries;	8 papers
Snowballing Search	6 papers
Total:	14 papers

## IV. RESULTS

This section presents the results of this SLR. The survey took place between August 2020 and January 2021. Here we present the results by year, by country, by types, and vehicles published. Finally, the following subsections detail the answer to each of the research questions.

#### A. General Results

While screening the papers included in this research, we found different approaches and technologies used to teaching programming to students with disabilities. Thus, we sought to answer the research questions and the answers are presented in this chapter. The results of the SLR shows that there are efforts to include people with cognitive disabilities in programming teaching, but there is still much to be explored.

1) *Publication year:* We distribute publications by year as shown in 1. We see that the peak of publications occurs in 2016 and 2018, corresponding to 21.42 % of publications. The years 2015 and 2020 account for 14.29 % each and 2011, 2013, 2017, and 2019 for 7.14 % each. Figure 1 shows that the scientific community is interested in the subject, and most years between 2010 and 2020, they published papers in this research context.

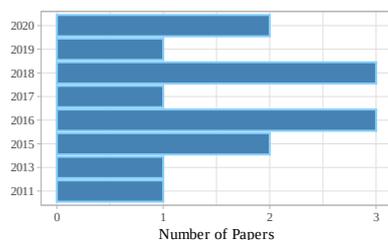


Fig. 1. Frequency by year

2) *Countries distribution*: Figure 2 shows that more than half of the papers published in the area come from the United States (US), totaling nine papers. Among the other papers there were two from the United Kingdom, one from Portugal / Brazil, one from Greece, one from Brazil, and one from Spain. The fact is that in the US there are more consistent initiatives to offer computer education and programming at an earlier age, which ends up reflecting on the most significant number of researches.

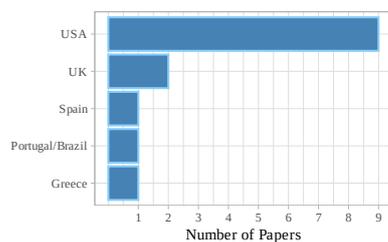


Fig. 2. Frequency by country

3) *Publication vehicles*: Most of the studies were published at conferences (six), as can be seen in Figure 3. The other papers appear in journals (four), symposiums (three), and workshops (one).

4) *Types of studies*: We separated the studies by type of research performed (Figure 4). Most were case studies (nine), followed by experience reports (four) and an exploratory study (one).

**B. RQ1: How did teachers approach the content for students with cognitive disabilities?**

It is understood that traditional teaching methods end up excluding students with cognitive disabilities or who learn differently. Therefore, the studies found reinforce this idea, experimenting with different approaches to help these students

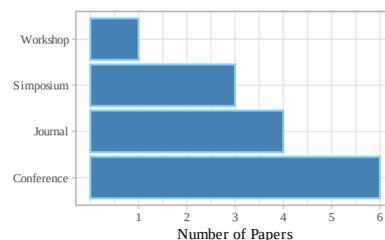


Fig. 3. Frequency by vehicle

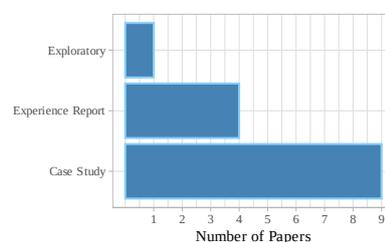


Fig. 4. Frequency by type

learn programming, as shown in Table V. Block programming and robotics were the most cited approaches in the studies.

Teachers who teach computing and CT should pursue integrated and straightforward approaches for instruction and tailor them to the content and learning objectives. There are numerous resources for teachers related to teaching programming and computing based on block programming, which makes it a good strategy for the initial concepts [5], [17], [19]–[21].

Robotics is very popular, mainly in kindergarten, being a strategy that uses fun to represent the most varied programming concepts. Robotics can be a factor of inclusion and equality, in addition to providing students with new knowledge through its attractive challenges [11], [13], [14], [16], [19]. On the other hand, one of the papers points out that students have more engagement when they use pre-built robots, but when they have to build the robots, they become less involved and more frustrated with more difficulty focusing on activities [13].

Other approaches have also proved popular, such as Universal Design for Learning (UDL) and Unplugged Programming (UP). Most initiatives are mainly focused on involving students, making them want to learn, and flexibly providing content, so that each student can choose the best way to receive them. UDL seeks to adapt traditional content and activities to reach different students with their different levels

of learning. There was no standardization among teachers for using this approach, and the main emphasis was on applying the principles of engagement and multiple forms of content presentation. We also realized that using this approach in computing was not significantly different from applying it in other areas. The key to using the UDL would be the teacher's view of their learning goals, noting the students' strengths, and seeking to reduce barriers to learning. [7], [10], [11], [20].

Teaching programming with non-electronic alternatives has become an exciting alternative. Physical toolkits can make abstract computing concepts easier to understand. Papers suggest that this approach is beneficial for teaching people with disabilities, but few studies still point in this direction. The use of physical tools associated with collaborative learning proved promising, with good interaction and enthusiasm among students, helping to keep them focused on activities for longer periods. On the other hand, it is not yet possible to guarantee that the activities with these devices are, in fact, inclusive for all students [11], [18]–[21].

The use of games was mentioned in fewer papers, focusing on the relationship between play and learning. They validate the idea that games leverage skills which could help individuals in the future labor market, such as teamwork and problem solving. This idea would also be interesting for students with ASD, as it stimulates some social skills. Robotics concepts are often used in association with games, but students with a higher functioning spectrum were more interested in game design and real-world storytelling with the characters they created [13], [15].

A single work selected in this review presented accessibility guidelines focused on students with dyslexia, seeking to help these students to overcome their difficulties and stand out in programming classes. A set of recommendations associated with the skills needed for programming was proposed, showing what difficulties the dyslexic programmer may face when trying to gain these skills and the reasonable adjustments that could also be applied to students who learn differently [27].

TABLE V  
APPROACH THE CONTENT

Approach	Papers
Block Programming	[5], [7], [13], [17], [19]–[21]
Robotics	[11], [13], [14], [16], [19]
Unplugged Programming	[11], [18]–[21]
UDL	[7], [10], [11], [20]
Gamming	[13], [15]
Own Guidelines	[27]

#### C. RQ2: What technologies are used to support teaching?

The use of specific technology aimed at teaching how to code is a popular approach in many studies, since there are countless options to be explored to find the best way to serve students. However, there is no standard for teaching people with disabilities, as we can see in Table VI. Two studies used most technologies, except Scratch, Code.org, and Dash Robot, covered by 6, 4, and 3 studies each, respectively.

Another interesting fact is that almost all technologies referenced by papers use blocking or robotics, with Magic Cubes Toolkit being the only technology used for UP.

In addition to serving as an accessible entry path for learning how to code, more playful technologies can promote coding as a fun and stimulating activity.

TABLE VI  
TECHNOLOGIES

Technologies	Papers
Lego Mindstorms	[13], [14]
Unit 3D	[15]
Scratch	[5], [11], [13], [14], [17], [20]
Code.org	[7], [11], [17], [20]
Khan Academy	[20]
Hello Ruby	[11]
Dash Robot	[11], [19], [21]
Blockly	[19], [21]
Alice	[13], [20]
Phogo	[11]
Magic Cubes Toolkit	[18]

#### D. RQ3: How are students evaluated?

Assessing student learning is often challenging due to the individual nature of each student's background. Good planning is essential for the teacher to identify whether the student learned that content. This review made it evident that the evaluation of students with disabilities is still something that needs to be studied and discussed, especially in the context of learning how to code.

The papers found in this review use mainly extracurricular courses, where the focus of the evaluation is the courses and not as a way to measure student's performance formally. Also, the evaluations are based on the student's perceptions, opinions, and personal data obtained through observations and interviews as can be seen in Table VII. Only two studies presented a formal strategy to measure student performance through a frequency checklist, analyzing the number of mistakes and successes of each study participant [19], [21]. There is a strong suggestion that formal evaluation of programming performance is an open field of study.

TABLE VII  
EVALUATION

Evaluation	Papers
Observation	[5], [10], [11], [13]–[18]
Interview	[11], [17]
Focus Group	[11], [13], [14], [16]
Student or Teacher Feedback	[13], [15]
Class Records	[7], [10], [11]
Photo or Video Analysis	[14], [18]
Checklist	[19], [21]

## V. DISCUSSION

Through this SLR, it was possible to notice several types of research involving the teaching of programming and people with cognitive disabilities. People who learn differently was

also a term used, seeking to emphasize that anyone can learn, as long as they have the correct support [7].

#### A. Can anyone learn programming?

One of the significant challenges in teaching programming to people with cognitive disabilities is how these students receive and absorb the content. The technical language seems to be a barrier, which leads us to question the best strategies to adapt the language to facilitate understanding [7].

Studies have shown that close support and mentoring for the individual student can facilitate the involvement of students with cognitive disabilities in the CT tasks assigned to their class [17].

The results found in this research show strong evidence that students with cognitive disabilities can develop the skills needed to learn programming provided they have adequate support, which implies training for teachers, adapted materials, and, on some occasions, collaborative work.

In line with content adaptations, the UDL methodology appears as a popular approach, which seeks to meet the needs of students in a different way by making content available in multiple formats, so that each student chooses the best way to obtain knowledge.

#### B. Programming as a profession

An important issue raised by this research is that the vast majority of papers in the area focus on children, as shown in Figure 5, where 12 of the 14 selected papers cover up to k-12. Programming is then presented playfully as another interesting knowledge for anyone to learn in these cases.

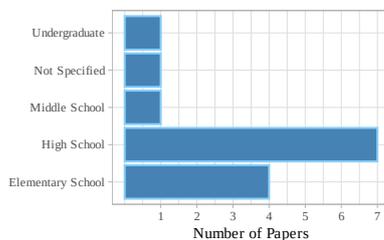


Fig. 5. Frequency by level

We cannot gauge how many of these students will pursue careers in computing. Hence, a possible area to be explored is the teaching of programming for people with disabilities in technical and higher education courses. Students with specific needs are often entering these courses without having prior knowledge on the field.

Teaching programming to students who have cognitive disabilities is already a reality with inclusive education policies. Teachers need to adapt their way of exposing knowledge and evaluating learning for each student, with and without

disabilities, as diversity in learning is already common in the classroom. Teachers face a significant challenge at any level of education: welcoming students who learn in different ways, looking for different methodologies to teach them according to their specific needs. In the context of higher education, however, the teacher's responsibility becomes significantly bigger, as these students are preparing for a professional career most of the time.

#### C. Types of cognitive impairment

The focus of this review was on papers involving research on teaching programming to people with some cognitive impairment. In the scope of cognitive disabilities, we can have several types, and the diversity of terms used in the papers reflects this, as we can see in Figure 6. Most of the terms were used in only one work, except for "Learning Difficulties" used in three papers and "Students with Disabilities" used in two papers. Most papers generically deal with cognitive impairment, leaving the question of whether the strategies used can be generalized or not since each disability has its characteristics and needs.

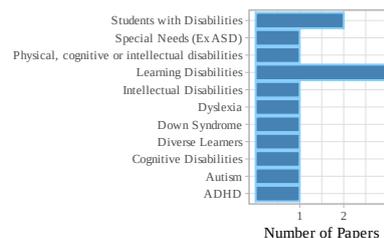


Fig. 6. Frequency by disability

## VI. THREATS TO VALIDITY

Even having followed the SLR protocol, we can identify threats to the research's validity. We designed the search key to cover the maximum number of terms related to the topic, but the lack of standard terminology may result in a smaller set of potential papers in the results. Each search base has specific features that may influence the results obtained and may not have returned all relevant studies for this work. The screening work is manual and human factors can influence data extraction and analysis when reading papers.

## VII. CONCLUSION AND FUTURE WORKS

Learning how to code has been considered an important skill with several initiatives for its popularization, starting as early as in elementary school or even before that. In classes with students of varying profiles, including people with cognitive disabilities, it can be difficult for teachers to teach the many concepts of Computer Science, often the abstract ones. With this SLR, it was possible to understand that several initiatives

seek the inclusion of people with cognitive disabilities in programming classes. We answered three research questions relating to programming and people with cognitive disabilities. Initially, we saw that the most used approaches were Programming in Blocks, Unplugged Programming, and Robotics, possibly as a way to bring a more playful approach within the learning context. Then it was identified that there is a variety of learning technologies explored, Scratch being the most mentioned in the papers, followed by the Code.org platform. It was also observed that there is no well-established learning assessment process, the observation and opinion of students and teachers through interviews is still the most adopted strategy to validate the proposals.

As future research topics, we suggest studies dedicated to helping students with cognitive disabilities in the initial programming disciplines of a technical or higher education course in the field of computing. These students can enter courses without prior knowledge in technology, with a deficit in logic and mathematics, needing additional help to assimilate content and grow to pursue a career in this area. Another critical open question is how to assess these students to see what skills they acquired during the course, a practice that becomes even more critical when these students are enrolled in CS courses since the programming disciplines of later stages require the knowledge gained in the initial classes. The introductory topics presented in those classes are crucial for students to progress in their studies and build a successful career.

#### REFERENCES

- [1] R. Munoz, R. Villarreal, T. S. Barcelos, F. Riquelme, Á. Quezada and P. Bustos-Valenzuela, "Developing Computational Thinking Skills in Adolescents With Autism Spectrum Disorder Through Digital Game Programming," in *IEEE Access*, vol. 6, pp. 63880-63889, 2018, doi: 10.1109/ACCESS.2018.2877417.
- [2] El Kah, A., Lakhouaja, A. Developing effective educative games for Arabic children primarily dyslexics. *Educ Inf Technol* 23, 2911-2930 (2018), doi: <https://doi.org/ez291.periodicos.capes.gov.br/10.1007/s10639-018-9750-2>
- [3] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. In Technical report, Ver. 2.3 EBSE Technical Report. EBSE, sn, 2007.
- [4] T. Zaki et al., "Towards developing a learning tool for children with autism," 2017 6th International Conference on Informatics, Electronics and Vision & 2017 7th International Symposium on Computational Medical and Health Technology (ICIEV-ISCMHT), Himeji, 2017, pp. 1-6, doi: 10.1109/ICIEV.2017.8338597.
- [5] Varsha Koushik and Shaun K. Kane. 2019. "It Broadens My Mind": Empowering People with Cognitive Disabilities through Computing Education. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 514, 1-12. DOI:<https://doi.org/10.1145/3290605.3300744>
- [6] S. Wille, J. Century and M. Pike, "Computer Science Principles (CSP) and students with learning differences: Expanding opportunities for a hidden underrepresented group," 2016 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), Atlanta, GA, 2016, pp. 1-8, doi: 10.1109/RESPECT.2016.7836160.
- [7] S. Wille, J. Century and M. Pike, "Exploratory Research to Expand Opportunities in Computer Science for Students with Learning Differences", in *Computing in Science & Engineering*, vol. 19, no. 3, pp. 40-50, May-June 2017, doi: 10.1109/MCSE.2017.43.
- [8] P. da Silva Neves Lima, L. das Almas Silva, J. L. dos Santos Oliveira, A. A. Franco Brandão and L. de Oliveira Brandão, "Computational games in STEM courses: a systematic review of the literature", 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1-8, doi: 10.1109/FIE44824.2020.9274071.
- [9] Brasil. Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (Inep). Censo da Educação Básica 2019: resumo Técnico. Brasília, 2020.
- [10] Meg J. Ray, Maya Israel, Chung eun Lee, and Virginie Do. 2018. A Cross-Case Analysis of Instructional Strategies to Support Participation of K-8 Students with Disabilities in CS for All. In Proceedings of the 49th ACM Technical Symposium on Computer Science Education (SIGCSE '18). Association for Computing Machinery, New York, NY, USA, 900-905. DOI:<https://doi.org/ez291.periodicos.capes.gov.br/10.1145/3159450.3159482>
- [11] Maya Israel, Gakyung Jeong, Meg Ray, and Todd Lash. 2020. Teaching Elementary Computer Science through Universal Design for Learning. In Proceedings of the 51st ACM Technical Symposium on Computer Science Education (SIGCSE '20). Association for Computing Machinery, New York, NY, USA, 1220-1226. DOI:<https://doi.org/ez291.periodicos.capes.gov.br/10.1145/3328778.3366823>
- [12] Claes Wohlin. 2014. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering (EASE '14). Association for Computing Machinery, New York, NY, USA, Article 38, 1-10. DOI:<https://doi.org/10.1145/2601248.2601268>
- [13] R. Dorsey and A. M. Howard. "Examining the Effects of Technology-Based Learning on Children with Autism: A Case Study", 2011 IEEE 11th International Conference on Advanced Learning Technologies, Athens, GA, USA, 2011, pp. 260-261, doi: 10.1109/ICALT.2011.81.
- [14] C. Conchinha, P. Osório and J. C. de Freitas, "Playful learning: Educational robotics applied to students with learning disabilities", 2015 International Symposium on Computers in Education (SIE), Setubal, Portugal, 2015, pp. 167-171, doi: 10.1109/SIE.2015.7451669.
- [15] C. Galeos, K. Karpouzis and G. Tsatiris, "Developing an educational programming game for children with ADHD." 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA, Zakynthos, Greece, 2020, pp. 1-6, doi: 10.1109/SMAP49528.2020.9248458.
- [16] Carlos Gonzalez-Sacristan, Carlos Garcia-Saura, and Pablo Molins-Ruano. 2016. Phogo: a low cost, engaging and modern proposal to learn how to program. In Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality (TEEM '16). Association for Computing Machinery, New York, NY, USA, 67-71. DOI:<https://doi.org/10.1145/3012430.3012498>
- [17] Snodgrass, Melinda. R., Israel, Maya., & Reese, George. C. (2016). Instructional supports for students with disabilities in K-5 computing: Findings from a cross-case analysis. *Computers & Education*, 100, 1-17.
- [18] Zuzanna Lechelt, Yvonne Rogers, Nicola Yuill, Lena Nagl, Grazia Ragone, and Nicolai Marquardt. 2018. Inclusive Computing in Special Needs Classrooms: Designing for All. In Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems. ACM, 517
- [19] Taylor, Matthew S. "Computer Programming With Pre-K Through First-Grade Students With Intellectual Disabilities". *The Journal of Special Education*, vol. 52, no. 2, Aug. 2018, pp. 78-88, doi:10.1177/0022466918761120.
- [20] Israel, M., Wherfel, Q. M., Pearson, J., Shehab, S., & Tapia, T. (2015). Empowering K-12 Students With Disabilities to Learn Computational Thinking and Computer Programming. *TEACHING Exceptional Children*, 48(1), 45-53. <https://doi.org/10.1177/0040059915594790>
- [21] M. S Taylor, E. Vasquez, and C. Donehower. 2017. Computer Programming with Early Elementary Students with Down Syndrome. *Journal of Special Education Technology* 32, 3 (2017), 149-159. <https://doi.org/10.1177/0162643417704439>
- [22] Kai Petersen, Sairam Vakkalanka, Ludwik Kuzniarz. Guidelines for conducting systematic mapping studies in software engineering: An update. *Information and Software Technology*, Volume 64, 2015, Pages 1-18, ISSN 0950-5849. <https://doi.org/10.1016/j.infsof.2015.03.007>
- [23] Cristina Santos e Denilson Silva e Maria Silveira e Giana Ferreira. 2017. Desafio de Programação para Meninas do Ensino Médio: Um Relato de Experiência. *Anais do Workshop de Informática na Escola* 23, 1 (2017), 137. <https://doi.org/10.5753/cbie.wie.2017.137>

- [24] American Psychiatric Association. 2013. Diagnostic and Statistical Manual of Mental Disorders. American Psychiatric Association. <https://doi.org/10.1176/appi.books.9780890425596>
- [25] K. Eiselt and P. Carter, "Integrating Social Skills Practice with Computer Programming for Students on the Autism Spectrum", 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 2018, pp. 1-5, doi: 10.1109/FIE.2018.8659252.
- [26] Maya Israel, Jamie N. Pearson, Tanya Tapia, Quentin M. Wherfel, George Reese. Supporting all learners in school-wide computational thinking: A cross-case qualitative analysis, *Computers & Education*, Volume 82, 2015, Pages 263-279, ISSN 0360-1315, <https://doi.org/10.1016/j.compedu.2014.11.022>. (<https://www.sciencedirect.com/science/article/pii/S0360131514002759>)
- [27] Silke Stienen-Durand, Jennifer George. Supporting Dyslexia in the Programming Classroom. *Procedia Computer Science*. Volume 27. 2014. Pages 419-430. ISSN 1877-0509. <https://doi.org/10.1016/j.procs.2014.02.046>.
- [28] Sylvia Stuurman, Harrie J.M. Passier, Frédérique Geven, and Erik Barendsen. 2019. Autism: Implications for Inclusive Education with respect to Software Engineering. In Proceedings of the 8th Computer Science Education Research Conference (CSERC '19), Association for Computing Machinery, New York, NY, USA, 15–25. DOI: <https://doi.org/10.1145/3375258.3375261>
- [29] Jeannette M. Wing. 2006. Computational thinking. *Commun. ACM* 49, 3 (March 2006), 33–35. DOI: <https://doi.org/10.1145/1118178.1118215>
- [30] R. Thompson, "Teaching coding to learning-disabled children with Kokopelli's World," 2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC), Cambridge, UK, 2016, pp. 258-259, doi: 10.1109/VLHCC.2016.7739705.
- [31] Andrew Luxton-Reilly, Simon, Ibrahim Albluwi, Brett A. Becker, Michail Giannakos, Amruth N. Kumar, Linda Ott, James Patterson, Michael James Scott, Judy Sheard, and Claudia Szabo. 2018. Introductory programming: a systematic literature review. In Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITICSE 2018 Companion), Association for Computing Machinery, New York, NY, USA, 55–106. DOI: <https://doi.org/ez291.periodicos.capes.gov.br/10.1145/3293881.3295779>
- [32] Damasio, J., Jurak, D., Bittencourt, R., Campos, M., & Amory, A. (2019). Programming teaching with robotic support for people who are visually impaired: a systematic review. *Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação - SBIE)*, 30(1), 1231. doi: <http://dx.doi.org/10.5753/cbie.sbie.2019.1231>

## **Apêndice C**

### **Identifying Programming Skills**

### **Impacted in Students with Cognitive**

### **Disabilities**

# Identifying Programming Skills Impacted in Students with Cognitive Disabilities

Elaine Cristina Juvino de Araújo  
Federal Institute of Paraíba  
Federal University of Campina Grande  
Campina Grande, PB, Brazil  
elainec@ifpb.edu.br

Wilkerson L. Andrade  
Federal University of Campina Grande  
Campina Grande, PB, Brazil  
wilkerson@computacao.ufcg.edu.br

Ana Liz Souto Oliveira  
Federal University of Paraíba  
Rio Tinto, PB, Brazil  
analiz@dcx.ufpb.br

**Abstract**—This Full Research Paper aims to identify and compare the skills involved in learning programming and the difficulties that students with cognitive disabilities may face, discussing the impacts generated by these skills for these students. Several studies talk about what skills a person needs to learn to program. However, we do not have enough to consider the needs of students with disabilities, especially cognitive impairments. Did we conduct a study to answer the following research questions: RQ1 - What skills are involved in programming teaching? RQ2 - Can cognitive impairments affect the skills needed to program? We lift the part the skills involved in learning programming and the cognitive skills impediments in the literature. Finally, we made a critical analysis impact on that audience. By cross-referencing this information, we can highlight that the skills needed to program impact students with cognitive impairments such as math, logic, and problem-solving. This impact does not necessarily mean that these students cannot learn to program. We believe adapting materials, exercises and assessments can foster learning for this audience.

**Index Terms**—teaching programming, cognitive disabilities, programming skills

## I. INTRODUCTION

Inclusive education can be considered enriching for teachers, promoting values such as empathy, respect, and equity, and can also have a beneficial effect on other students within the university [24]. Providing inclusive access to Computer Science education is a significant challenge for educators as they need to address the diverse needs of diverse students in the classroom, including students with learning disabilities such as reading, math, neurodevelopmental disorders, and others [1].

Learning programming is considered an exciting competency for individuals to have a promising professional career in Computer Science and other fields. Programming is challenging knowledge, and teaching people to program is not a simple task [5]. Even though it is a career that attracts many people, Computing is still considered excluding for people with disabilities, especially those with cognitive disabilities [11].

The first programming discipline is essential because it is the students' first contact with the fundamentals and programming languages. Typically, they arrive at university without programming classes in their previous school life. It is a discipline that traditionally has significant rates of failure and

dropout, which has generated research to investigate ways to keep students in the course. There are many efforts to solve these problems, investigating new curricular adaptations and the search for the insertion of this knowledge at earlier ages, including childhood [20].

Several studies talk about what skills a person needs to learn to program. Understanding how to solve a problem with programming requires different skills from the student, such as thinking logically and understanding some mathematical concepts. Students need to know the skills involved in creating computer programs that solve real-world problems, which are specific and usually acquired in a short time, over [7]. Another critical issue is the difficulty in identifying programming skill demands, as different programmers may use different ways to solve a problem. We may find a programmer who has issues with specific skills, like designing solutions, but is good at debugging programs [21].

Despite existing efforts, we still do not have enough to consider the needs of students with disabilities, especially cognitive disabilities. Cognitive impairment affects the individual's cognition in some way, including neurodevelopmental disorders such as autism spectrum disorder (ASD) and attention deficit hyperactivity disorder (ADHD), but not limited to these [12], [18]. Understanding the prerequisites to direct how to deal with these students, especially those with more significant difficulties, is essential. The cognitive aspects of computer programming are an area of research where there is much to be explored [15]. We still need more research to improve programming teaching materials, methodologies, and how students absorb it.

Compared to other disabilities such as visual impairment, few studies investigate cognitive impairments in Computer Education [11]. We conducted this research to explore what might impact the learning of programming students with intellectual disabilities. Several works discuss the skills needed to program, but we did not find studies linking these skills to people with cognitive disabilities with specific limitations and needs. This article presents research on skills for learning to program and aims to analyze the likely impacts of these skills on the learning of students with cognitive disabilities. It seeks to answer the following research questions: RQ1 - What skills are involved in programming teaching? and RQ2 - Can

cognitive impairments affect the skills needed to program?

This paper is structured as follows: Section II contains related works. In Section III, we present the methodology, and in Section IV, we have the results answering the research questions. We discuss the results found in Section V. We show threats to validity in Section VI and conclude and point to future work in Section VII.

## II. RELATED WORKS

Several jobs deal with programming skills in general. It is pertinent to understand if anyone can learn to program as long as they have or somehow acquire specific skills. Given this, it is still not explored how people with disabilities can be impacted when learning programming, which is a vast field of research, as each category of disability may have different characteristics and needs.

A previous systematic review [19] identified some efforts to teach programming to people with cognitive disabilities. She made a collection of articles that addressed programming and people with some cognitive disabilities. He identified that much of the work is focused on contact with programming at an early age, still in childhood, in extracurricular courses. Therefore, there is a large paper space focusing on teaching programming and cognitive impairment in higher education, where we can deal with people seeking programming as a potential career. The results indicated that students with cognitive disabilities could develop the skills necessary to learn programming provided they have adequate support, both in access to teachers prepared to receive them and in adapted materials and collaborative work. In this previous work, the need arose to observe in more detail the relationship between the difficulties inherent to cognitive deficiencies and the teaching of programming. Then, this research inspires us to design the current research seeking answers to this goal.

Medeiros et al. brought interesting results identifying student skills for programming [32]. The review gave an overview of the teaching and learning of programming in higher education, revealing the challenges and problems. One of the research questions was to identify the skills and prior knowledge that the student would need to learn to program. The other issue was related to the challenges of students and teachers. As part of the results, he categorized skills into programming-related (such as problem solving and mathematics) and education-related in general (such as knowledge of English and creativity). This paper seeks the skills needed to program but is not concerned with students who may learn differently from the traditional way.

Stuurman et al. aimed to create a set of recommendations for the inclusive education of autistics related to Software Engineering [22]. For this, it compared the cognitive style of autistic students with Computational Thinking. The work resulted in guidelines for the education of students with autism that are: (1) Explicit Context, (2) Explicit Guidelines, (3) Exercises, and (4) Consequences for Education. Although these recommendations are based only on the student's opinions, there is a conviction that they are valid but that the area needs

further exploration. The authors point out some suggestions for future work to support the autistic cognitive style, such as listening to students' opinions, a collaboration between autistic and neurotypical students, and explicit recommendations for everything done. However, this research speculatively created guidelines without empirical evidence of its efficiency.

Israel et al. present a study that researched the academic engagement of students with autism in the elementary teaching of Computer Science [36]. A case study was carried out using collaborative work with three elementary school students with ASD in an extra course. One of the limitations of the work was the number of students and their selection criteria. As a result, they observed that engagement was varied, often low, with little problem-solving. They also saw that the experience could have been more positive if they had had individualized support. In future work, they suggest deepening the students' experiences with deficiencies to improve their performance.

Eiselt et al. presented a work in progress that sought to integrate social skills with computer programming for students with autism [40]. An extracurricular course analyzed the results for nine students between 9 and 16 years. Research suggests that combining social skills with computer education benefits autistics, but more studies are needed to prove it.

Silva et al. sought to show the experience of teaching Computational Thinking through the teaching of Programming to a high school student with ADHD [13]. An extracurricular course was conducted for 34 weeks, which resulted in solid evidence of student learning. It was divided into three phases: (1) Discovery of the Computer; (2) Introduction to Computational Thinking through Programming; and (3) Learning by doing, with problem-solving, collaborative work, and challenges. The researchers used the following steps to validate the method: (1) Student Profile Analysis, (2) Student Assessment and (3) Student Performance Analysis with the results of the Assessment. To consider that the student was successful in learning programming, the authors expected him to reach at least 80% of correct answers in the evaluation. The student obtained 84% of correct answers, suggesting the experiment's effectiveness. For future work, recommend carrying out courses with more students to prove the method's success. The experience presented is engaging but carried out in an extracurricular class. In addition, we have no evidence of positive results for students of Computing courses with ADHD.

With these related works, we did not answer our research question as they work on programming skills without taking into account students with disabilities [32] or use extracurricular courses to evaluate their theories [13], [22], [36], [40], which leaves us with gaps to explore, such as knowing what skills are needed for programming that can be impacted by cognitive impairments, to which we are dedicated in this research. We want to research ways to encourage students of Computer Science courses to learn to program so that they can progress in their courses and, who knows, get a profession.

### III. METHODOLOGY

This section presents the research questions and the methodology for answering each one.

#### A. Research Questions

The main objective of this work is to identify the primary skills involved in the programming learning process that could impact students with cognitive disabilities. Thus, we derive the following research questions:

- (RQ1) What are the skills involved in teaching programming?

The RQ1 aims to identify the skills needed to learn to program, carrying out a bibliographic search related to the skills involved in programming learning.

- (RQ2) Can cognitive impairments affect the skills needed to program?

In RQ2, we analyze the impact on students with cognitive disabilities based on the skills needed to program. We looked for the potential cognitive impediments that would impact these skills.

#### B. A Systematic Review Protocol

We adopted a systematic review protocol based on Kitchenham and Charters [25] to answer RQ1. We present the steps as follows:

1) *Conducted Search*: To initially identify the skills needed to program, we focused our bibliographic research on the IEEE Xplore and ACM digital libraries, performing a literature review necessary for our analysis.

The search string chosen to obtain skills related to teaching programming was:

**(“programming” AND “teaching” AND “skills”)**

We searched for the terms chosen in the articles’ titles, abstracts, and keywords. At the end of the search, we obtained several works, as shown in Table I.

TABLE I  
SUMMARY OF THE STUDIES RETURNED IN EACH DIGITAL LIBRARY

Digital Library	Search Results
ACM	374
IEEE	500
<b>Total</b>	<b>874</b>

When searching the databases with this search key, we obtained 874 papers. We choose the inclusion and exclusion criteria. The inclusion criteria are:

- (IC1) - Papers in English;
- (IC2) - Papers published between 2018 and 2022 (until January);
- (IC3) - Papers that contain the terms in the search string at least in the title, abstract, or keywords;
- (IC4) - Papers dealing with skills involved in teaching programming.

The exclusion criteria are:

- (EC1) - Repeated papers;
- (EC2) - Papers that do not adhere to the theme;
- (EC3) - Studies that do not address skills in the context of programming.

2) *Screening of papers*: We present the screening process results in Table II, applying exclusion and inclusion criteria. Later we read the titles, abstracts, and keywords because the database can return articles unrelated to what we were looking for, even applying the chosen search key. Some of the articles returned in the search were discarded because they did not meet the criteria.

TABLE II  
SCREENING OF PAPERS

Steps	Papers
Step 1: Identify and organize the papers found in the search bases by the inclusion criteria IC1 and IC2;	874 papers
Step 2: Refine by inclusion criteria IC3;	325 papers
Step 3: Review of titles removing papers that fit the exclusion criteria;	69 papers
Step 4: Review of abstracts, removing papers that fit the exclusion criteria;	25 papers
Step 5: Review of the complete papers that fit the exclusion criteria;	8 papers

We check the number of articles returned in the search for the first step by refining the result. We applied the IC1 and IC2 inclusion criteria, thus removing papers not in English or had not been published from 2018 to 2022 (until January).

In the second step, we selected only the articles with the keyword search terms in the title, abstract, or keywords. We eliminated several articles that passed the automated search but were not related to what we were researching, and we had 325 papers as a result. In the third stage, we analyzed the titles of the articles, refining them once more and removing what was not related, obtaining 69 papers. We read the abstracts in the fourth stage, and then we had the number of articles even further reduced, with a total of 25 papers for analysis. Finally, the last step of the screening process was the complete reading of the articles resulting from the previous steps, of which only eight were considered relevant for the research and on which we focused our work.

#### C. Theoretical Analysis

To answer RQ2, we used a literature search based on papers obtained in the previous review [19] and an official document from health agencies [18] that crossed with our findings from RQ1. In this search, we found some of the main characteristics of each disability and reports of how people who have one of these can react to situations that involve teaching programming. Based on this investigation, we proposed a cross between this information and the skills needed for the program we saw in RQ1, seeking to find evidence of how to help these students better.

#### IV. RESULTS

This section presents the search results between October 2021 and January 2022. Here we offer the general results supporting the answer to each research question.

As this work focuses on the intersection of programming skills with cognitive disabilities, we seek to identify how the specificities of students who fit this diagnosis impact in this case through a bibliographic search.

To achieve our research objective, we started a literature review to find out what is necessary for anyone to learn to program so that we can later analyze the impact on students with cognitive disabilities using our findings from the previous review [19]. We selected papers that listed likely skills for learning programming and compiled the results to answer the first research question in the next section.

First, we present the general results, such as studies distribution by year, country, and publication vehicle (journal or conference).

We distribute articles by year, as shown in Figure 1. We can see that there is interest in the topic, but even searching papers from 2018 to 2022 (until January), we found papers only in 2018, 2019, and 2020 using our criteria.

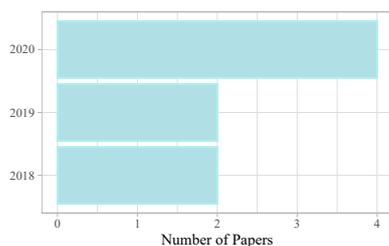


Fig. 1. Frequency of publication of papers distributed per year

Observing Figure 2, we can see the papers from different countries, with only Mexico responsible for more than one paper.

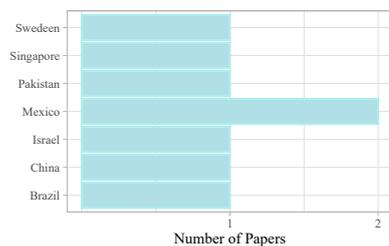


Fig. 2. Frequency of publication of papers distributed per country

As for the publication vehicle, most of them (five papers) are conference papers, and the rest (three articles) are journal papers, as shown in Figure 3.

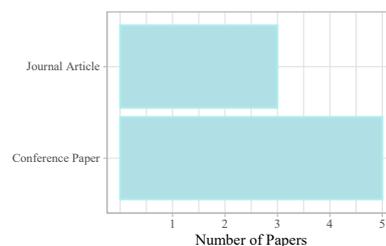


Fig. 3. Frequency of publication of papers distributed per vehicle

##### A. RQ1: What skills are involved in teaching programming?

To answer RQ1, we searched the literature for what skills would be necessary for those who want to learn to program. From the search focused on the IEEE and ACM, we selected eight papers that dealt with the subject [28]–[35]. By synthesizing the selected ones, we could enumerate a set of skills we will use as a basis for this research. These skills are what we will discuss in detail as follow.

To learn to program, the student needs some skills that they can acquire throughout his life; they are:

1) *Problem Solving*: Problem-solving is part of the concept of Computational Thinking (CT) [27], where programming is also inserted. Part of the learning process is planning the resolution of a given problem, breaking it down into smaller steps, and making its solution more effective.

One of the biggest challenges for Computer Science students is learning programming concepts, including problem-solving skills. Many students have difficulties understanding the problem and converting it into a program [28].

Considering that a computer program is also a solution, we observe that the programmer needs to know how to solve problems. Problem-solving is a core skill in computer programming. Students need to learn how to solve a problem, understanding that there is no one-size-fits-all solution, and it's a need to create another solution for each issue [29]. Assuming that there is no single solution to a problem, we can say that students with different learning profiles can reach other solutions for the same problem.

Another critical issue is that in addition to having the ability to solve problems, it would be interesting for the programmer to have the ability to solve problems collaboratively since, in employment, they will often face the creation of solutions with a team of workers [34].

2) *Mathematical Skills*: We can say that programming and mathematics go together. Observing traditional programming exercises, we see several of these using mathematical formulas

as a starting point. The student then needs to interpret the problem and identify what would be the mathematical logic that would fit there to arrive at its solution.

The logical reasoning part of mathematics is about understanding the cause-and-effect relationship of things. Following a precondition, a logical conclusion is inferred from it. It also comprises sequencing steps to solve a given problem, sometimes using mathematical procedures and formulas [28].

3) *Abstraction*: Abstraction is a concept that can help the student create a structure to solve a given problem. Usually, in this phase, at the end of the solution, he will refine the steps into your algorithm [23]. Abstract thinking consists of thinking about a problem on several levels simultaneously, seeking to detail each level [28].

4) *Critical Thinking*: Critical thinking consists of conceptualizing something, analyzing and synthesizing it to obtain a conclusion or answer to a problem [28].

Critical thinking is often difficult for people. Understanding what it means and inferring a conclusion is considered a significant challenge when having information about something.

5) *Reading Comprehension*: Text comprehension is about finding the main idea of the text, its implications, and the possibility of relating it to the knowledge the reader already has, which can help in its understanding [28].

The first step of programming would be to understand the problem to solve a problem. To understand a problem, the programmer must have mastery of reading and effectively understand what he is reading.

Most approaches to teaching programming revolve around problem-solving skills, which can leave out other essential skills such as language skills [14].

6) *Knowledge in English*: When we talk about computer programming, we can see that the mastery of the English language can be a facilitator, as programming languages use this language in their syntax [32], so understanding English helps to decipher what each command does for the meaning of the term.

7) *Attention to Details*: It seems evident that paying attention to learning something is necessary, but this skill is not so simple. In a dynamic world full of stimuli, paying attention to something with so many others becomes difficult.

From a programming point of view, paying attention requires considerable concentration from the observer to analyze, for example, a problem or a source code. For programming, it is essential to think when creating, testing, or correcting a solution, thinking about both the happy flow and the failures that can be estimated [28].

8) *Creativity*: Creativity is a general skill where the individual can think to have ideas, which can be beneficial when creating an algorithm since there is no single solution to solve a problem. Computer programming helps to develop skills like creativity [7]. Being creative can help the programmer.

9) *Time Management*: Managing time well is considered an essential skill for programming [32]. When developing a project, we need to know how to divide the time for the execution of the resolution steps properly.

### B. RQ2 - Can cognitive impairments impact the skills needed to program?

We use the term cognitive impairment to cover various impairments that somehow limit the individual's mental functions [16]. To answer RQ2, we first worked with some of the characteristics associated with cognitive impairments extracted from the literature and health agencies [18] and related to programming learning. We sought to observe the set of characteristics of each disability, which crossed with the list of skills we have.

Several types of disability can affect individuals' cognition. In this work, we focus on some of the most known; they are:

- Autism - Autism Spectrum Disorder is characterized by deficits in communication, social interaction, and repetitive behavior patterns.
- ADHD - Attention Deficit Hyperactivity Disorder is characterized by inattention, impulsivity, or both that persist over time.
- Dyslexia - In Dyslexia, we have a pattern of learning difficulties, mainly in word recognition and understanding.
- Intellectual Disability - People with intellectual disabilities usually have deficits in mental functions such as reasoning, problem-solving, and abstract thinking, among others.

Cognitive impairments can have levels of severity: mild, moderate, severe, and profound. Levels may vary from person to person, and we may have cases of individuals with more than one disability simultaneously, for example, Autism and Intellectual Disability [18].

We used a theoretical search based on the literature to cross skills and characteristics of disabilities, and, as a result, we found the relationships explained in Table III.

Crossing skills and different types of cognitive disabilities, we have in Table III an outline of which skills impact each disability to have a starting point for future interventions.

TABLE III  
SKILLS VERSUS COGNITIVE DISABILITIES

Skill	Disability
Problem Solving	Intellectual, Dyslexia
Mathematical Skills	Intellectual,
Abstraction	Intellectual, ADHD
Critical Thinking	Autism
Reading Comprehension	Intellectual, Dyslexia, ADHD
Knowledge in English	Cognitive Disabilities
Attention to Details	ADHD
Creativity	ADHD
Time Management	Intellectual, Autism

1) *Problem Solving*: Studies show that there is still little research linking students with Intellectual Disabilities and programming skills. These students have more difficulty in adaptive functions directly related to skills such as problem-solving, acquired more slowly than a neurotypical student [37].

2) *Mathematical Skills*: Analyzing the skills found in RQ1, we can see that students with Intellectual Disabilities will have

difficulties in skills mainly linked to concepts of logic and mathematics [37].

3) *Abstraction*: Students with Intellectual Disabilities have an easier time understanding concrete concepts rather than abstract concepts [37].

4) *Critical Thinking*: Critical thinking can be complex, for example, for those who have Autism and challenging behaviors, so programming activities that incorporate this skill should be designed to favor this skill [39]. In ADHD, critical thinking can also be impaired [18].

5) *Reading Comprehension*: Autistic individuals may have delays in language and understanding, a lack of speech, and impairments in social communication. People with moderate Intellectual Disabilities may make slow progress in reading and writing. [18].

6) *Knowledge in English*: The programming languages are English-based, so anyone unfamiliar with the idiom may have difficulties. People with cognitive disabilities who are not native speakers of English [17] may need additional support. Several factors can affect performance for programmers [15], especially beginners. Language is one of those factors which can influence the programmer having a significant impact on performance and programming learning.

7) *Attention to Details*: Attention is an essential skill for programming. We need to be attentive to details when we are creating code. In addition to creating the solution, we must also be careful to obey the language's syntax. Those with ADHD may have difficulty with attention, hyperactivity, or both [2].

8) *Creativity*: Skills that involve greater concentration, such as creativity, would be more impacted in students with ADHD [2].

9) *Time Management*: People with an Intellectual Disability may have trouble understanding time. Autistic people have difficulty switching activities and organizing and planning their time [18]. This condition can cause damage to meeting deadlines, for example.

Studies suggest that learning skills in computer programming can lead to skills in sequencing, investigation, or problem solving for students with intellectual disabilities [37]. On the other hand, studies have shown that students with dyslexia can solve problems quickly, and programming is an area that can enhance their strengths and overcome their weaknesses [2], [38]. When interviewed, some professional developers with dyslexia stated their condition would not pose significant obstacles in their careers [4]. People with Autism and ADHD can develop hyperfocus, a trait where the person devotes a great deal of concentration to something that interests them. This characteristic can facilitate if the student is interested in learning programming. The teacher can look for ways to bring this interest through the use of situations that hold his attention, such as the use of games if he likes to play [13]. We can not look at cognitive deficiencies only about their difficulties but their potential, verifying the characteristics of each deficiency that could help in programming learning.

## V. DISCUSSION

This study made it possible to realize that the skills involved in teaching programming can impact students with some cognitive disabilities. As this condition can vary in level and affect individuals differently, we can infer that each student may need specific support. In this way, the teacher's watchful eye will be essential to understand which skills each student is having difficulties using the appropriate recommendations and observe their potential to help them in their studies. The teacher alone may not be successful, so how could he know how to identify these difficulties and skills? What approach could he use: tools, games, specific tests? It's an important question.

We need to analyze what it will take to provide this support for future research. Therefore, the teacher's perspective may be an essential issue. Technologies are constantly evolving, but the school resists traditionally approaching the curriculum, leading to students' lack of interest [10]. Improving didactics for programming courses has always been a challenge for teachers [9]. Teachers often do not provide adequate support due to a lack of knowledge about practical [3] approaches. Teachers must give a constant and personalized support to each student, allowing them to acquire and improve their specific programming skills [8].

Another issue would be the teaching perspective. Programming teaching can revolve around different teaching theories, such as constructivism [9]. These widely used theories can facilitate or hinder adequate support for students with cognitive disabilities. It is necessary to deepen the studies with what we already have as an alternative and propose other possibilities of knowing what works for this audience.

The challenges are not limited to these. We can also add the evaluative question, which is very important and needs to be fair within the limitations and potential of the student.

The Universal Design for Learning [3] was presented as a strategy for teaching people with disabilities. It is based on three basic principles: (1) Multiple means of engagement, (2) Multiple means of representation, and (3) Multiple means of action and expression. Its essence of offering the student multiple ways to learn seems to be a promising approach. Still, it was not created explicitly to meet the specifics of teaching programming, nor for a specific type of disability. We need more research on it, especially for people with cognitive disabilities.

The W3C [26] currently has a working group dedicated to creating guidelines for web content for people with disabilities. These guidelines are supplemental to the existing and consolidated Web Accessibility recommendations [42]. We have a lot of content and materials about programming on the web, especially after the COVID pandemic, which increased the need to explore new teaching opportunities, especially at a distance [41]. These notes are interesting because some can be applied to content creation, for example, for programming classes.

For the following steps, the question remains, if people with cognitive disabilities have difficulties in concepts such

as abstraction, logic, and mathematics, how can they learn programming that is content that requires these same concepts? The key to the answer is the constant search for relevant information that brings a sensitive look to the cause of these students. Difficulties exist, and students may take longer to learn, but evidence suggests they can learn. We can not only look at the challenges presented but also explore students' potential, seeking the best in them to favor their learning. Constant support is the key to the success of people with cognitive disabilities in achieving their goals [17].

#### VI. THREATS TO VALIDITY

Like any research, this one has threats to its validity. We conducted a bibliographic study through a systematic review of the skills literature and subsequent analysis. There were limitations due to the terms and characteristics of each search base. As the screening and analysis work is manual, human factors can affect the extraction and analysis of data and the interpretation of articles. Still, we tried to minimize them with more than one check.

#### VII. CONCLUSION AND FUTURE WORKS

Computer Education is still a new field of research [6], but there are already considerable efforts in the literature to discuss what skills are needed for programming. This work focused on a new look at these skills, a more inclusive look, seeking new perspectives to help people with cognitive disabilities who can learn, but in a different way and with the proper support of teachers and institutions.

We observed through RQ1 that a wide range of skills is necessary for those who want to learn to program. These skills are mainly linked to Mathematics and Logic. With RQ2, we found that the various types of cognitive impairment can impact the skills needed to program, and every kind of disability would not necessarily impact all skills.

It is a great challenge to offer equal opportunities to students with and without disabilities, and we need to plan adaptation strategies for content, practice, and exercises. It does not look like an easy but necessary task. We must research the likely problems about the difficulties presented by the students for each content. Try to foresee many challenges with traditional teaching and propose changes, testing your results with the target audience. We are confident that students with cognitive disabilities will have equal opportunities to access knowledge that involves programming with appropriate adaptations.

For future work, we suggest the creation of proposals for curricular adaptations, materials, contents, exercises, and assessments. For this, they must consider the skills impacted by each disability as a starting point. We can take these proposals to practical application in the classroom and follow the evolution of students in the first programming discipline, providing training and support to teachers.

When considering inclusion, we will favor students who need permanent support, and all students in the class will benefit.

#### REFERENCES

- [1] S. Wille, J. Century, and M. Pike. "Exploratory Research to Expand Opportunities in Computer Science for Students with Learning Differences", *Computing in Science Engineering*, vol. 19, no. 3, pp. 40–50, 2017.
- [2] S. Wille, J. Century, and M. Pike. "Computer Science Principles (CSP) and students with learning differences: Expanding opportunities for a hidden underrepresented group", in *In Proceedings of the RESPECT'16*, 2016, pp. 1–8.
- [3] M. Israel, G. Jeong, M. Ray, and T. Lash. "Teaching Elementary Computer Science through Universal Design for Learning", in *Proc. of SIGCSE 20*, 2020, pp. 1220–1226. doi: 10.1145/3328778.3366823.
- [4] R. Thompson. "Teaching coding to learning-disabled children with Kokopelli's World", in *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, 2016, pp. 258–259. doi: 10.1109/VLHCC.2016.7739705
- [5] A. Luxton-Reilly et al., "Introductory Programming: A Systematic Literature Review", in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, 2018, pp. 55–106. doi: 10.1145/3293881.3295779.
- [6] J. Waite and S. Sentance. "Teaching programming in schools: A review of approaches and strategies". <https://www.raspberrypi.org/app/uploads/2021/11/Teaching-programming-in-schools-pedagogy-review-Raspberry-Pi-Foundation.pdf>. [Online]. Available: <https://www.raspberrypi.org/app/uploads/2021/11/Teaching-programming-in-schools-pedagogy-review-Raspberry-Pi-Foundation.pdf>
- [7] J. Figueiredo and F. J. García-Peñalvo, and Learning Strategies of Programming for University Courses," in *Proceedings of the Seventh International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2019, pp. 1020–1027. doi: 10.1145/3362789.3362926.
- [8] J. Figueiredo and F. J. García-Peñalvo, "Building Skills in Introductory Programming," in *Proceedings of the Sixth International Conference on Technological Ecosystems for Enhancing Multiculturality*, 2018, pp. 46–50. doi: 10.1145/3284179.3284190.
- [9] Y. Shi, S. Huang, and C. Zheng. "Research on Programming Courses Teaching Based on Blended Learning," in *Proceedings of the 5th International Conference on Frontiers of Educational Technologies*, 2019, pp. 30–34. doi: 10.1145/3338188.3338198.
- [10] C. Conchinha, P. Osório, and J. C. de Freitas, "Playful learning: Educational robotics applied to students with learning disabilities," in *2015 International Symposium on Computers in Education (SIEE)*, 2015, pp. 167–171. doi: 10.1109/SIEE.2015.7451669.
- [11] V. Koushik and S. K. Kane. "It Broadens My Mind": Empowering People with Cognitive Disabilities through Computing Education," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, New York, NY, USA: Association for Computing Machinery, 2019, pp. 1–12. [Online]. Available: <https://doi.org/10.1145/3290605.3300744>
- [12] Q. Li, J. Lee, C. Zhang, and K. Reinecke. "How Online Tests Contribute to the Support System for People With Cognitive and Mental Disabilities," in *The 23rd International ACM SIGACCESS Conference on Computers and Accessibility*, New York, NY, USA: Association for Computing Machinery, 2021. [Online]. Available: <https://doi.org/ez291.periodicos.capes.gov.br/10.1145/3441852.3471229>
- [13] F. F. da Silva, L. Beatrys Ruiz Aylon, and D. E. Flôr, "Teaching Computational Thinking to a Student with Attention Deficit Through Programming," in *2020 IEEE Frontiers in Education Conference (FIE)*, 2020, pp. 1–9. doi: 10.1109/FIE44824.2020.9273915.
- [14] E. Fedorenko, A. Ivanova, R. Dhamala, and M. U. Bers. "The Language of Programming: A Cognitive Perspective," *Trends in Cognitive Sciences*, vol. 23, no. 7, pp. 525–528, 2019. doi: <https://doi.org/10.1016/j.tics.2019.04.010>.
- [15] A. W. Lewis, V. Miller, I. Kane, and M. Marrazzo. "A Systematic Literature Review of Cognitive Models of the Behavior of Novice Software Developers," *J. Comput. Sci. Coll.*, vol. 34, no. 2, pp. 208–214, Dec. 2018.
- [16] A. Nystedt. "Designing for Cognitive Disability How FlexiChat Affects the Accessibility of Instant Messaging," *UmeåUniversity, Department of Computing Science*, 2019.
- [17] S. Collings, A. Dew, and L. Dowse. "Unpacking the complexity of planning with persons with cognitive disability and complex support

- needs," *Journal of Applied Research in Intellectual Disabilities*, vol. 31, no. 1, pp. 142–151, 2017, doi: <https://doi.org/10.1111/jar.12381>.
- [18] American Psychiatric Association: *Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition*. Arlington, VA: American Psychiatric Association, 2013
- [19] E. C. J. De Araújo and W. L. Andrade, "A Systematic Literature Review on Teaching Programming to People with Cognitive Disabilities," 2021 IEEE Frontiers in Education Conference (FIE), 2021, pp. 1–8, doi: [10.1109/FIE49875.2021.9637361](https://doi.org/10.1109/FIE49875.2021.9637361).
- [20] Sobral, S.R. Teaching and Learning to Program: Umbrella Review of Introductory Programming in Higher Education. *Mathematics* 2021, 9, 1737. <https://doi.org/10.3390/math9151737>
- [21] Kurland, D. M., Clément, C. A., Mawby, R., & Pea, R. D. (1987). Mapping the cognitive demands of learning to program. In R. D. Pea & K. Sheingold (Eds.), *Mirrors of minds: Patterns of experience in educational computing* (pp. 103–127). Ablex Publishing.
- [22] Sylvia Stuurman, Harrie J.M. Passier, Frédérique Geven, and Erik Barendsen. 2019. Autism: Implications for Inclusive Education with respect to Software Engineering. In *Proceedings of the 8th Computer Science Education Research Conference (CSERC '19)*. Association for Computing Machinery, New York, NY, USA, 15–25. DOI:<https://doi.org/10.1145/3375258.3375261>
- [23] Calderon, A.C., Skillicorn, D., Watt, A. et al. A double dissociative study into the effectiveness of computational thinking. *Educ Inf Technol* 25, 1181–1192 (2020). <https://doi.org/10.1007/s10639-019-09991-3>
- [24] Rafael Carballo, Victor M. Molina, María-Dolores Cortes-Vega & Ruth Cabeza-Ruiz (2022): Students with disabilities at university: benefits and challenges from the best faculty members' experiences, *European Journal of Special Needs Education*, DOI: 10.1080/08856257.2022.2031104
- [25] B. Kitchenham and S. Charters. Guidelines for performing systematic literature reviews in software engineering. In *Technical report*, Ver. 2.3 EBSE Technical Report. EBSE. sn, 2007.
- [26] Making Content Usable for People with Cognitive and Learning Disabilities. (2019, July 16). W3.org. <https://www.w3.org/TR/coga-usable/>.
- [27] M. Elshahawy, K. Aboelnaga and N. Sharaf, "CodaRoutine: A Serious Game for Introducing Sequential Programming Concepts to Children with Autism," 2020 IEEE Global Engineering Education Conference (EDUCON), 2020, pp. 1862–1867, doi: [10.1109/EDUCON45650.2020.9125196](https://doi.org/10.1109/EDUCON45650.2020.9125196).
- [28] R. Juárez-Ramírez, C. X. Navarro, V. Tapia-Ibarra, R. Macías-Olvera and C. Guerra-García, "What is Programming? Putting all Together - A Set of Skills Required," 2018 6th International Conference in Software Engineering Research and Innovation (CONISOFT), 2018, pp. 11–20, doi: [10.1109/CONISOFT.2018.8645956](https://doi.org/10.1109/CONISOFT.2018.8645956).
- [29] M. F. Ercan and D. Sale, "Teaching programming: An evidence based and reflective approach," 2020 IEEE REGION 10 CONFERENCE (TENCON), 2020, pp. 997–1001, doi: [10.1109/TENCON50793.2020.9293812](https://doi.org/10.1109/TENCON50793.2020.9293812).
- [30] Z. Guanghui, L. Yanjun, T. Yixiao, W. Zhaoxia and Z. Chengming, "Case-Based Teaching Organization for Python Programming that Focuses on Skill Training," 2018 13th International Conference on Computer Science & Education (ICCSE), 2018, pp. 1–5, doi: [10.1109/ICCSE.2018.8468860](https://doi.org/10.1109/ICCSE.2018.8468860).
- [31] R. Juárez-Ramírez, S. Jiménez and V. Tapia-Ibarra, "What is Programming? Putting all together, Part II –The Cognitive Skills associated," 2019 7th International Conference in Software Engineering Research and Innovation (CONISOFT), 2019, pp. 55–64, doi: [10.1109/CONISOFT.2019.00018](https://doi.org/10.1109/CONISOFT.2019.00018).
- [32] R. P. Medeiros, G. L. Ramalho and T. P. Falcão, "A Systematic Literature Review on Teaching and Learning Introductory Programming in Higher Education," in *IEEE Transactions on Education*, vol. 62, no. 2, pp. 77–90, May 2019, doi: [10.1109/TE.2018.2864133](https://doi.org/10.1109/TE.2018.2864133).
- [33] E. Mehmood, A. Abid, M. S. Farooq and N. A. Nawaz, "Curriculum, Teaching and Learning, and Assessments for Introductory Programming Course," in *IEEE Access*, vol. 8, pp. 125961–125981, 2020, doi: [10.1109/ACCESS.2020.3008321](https://doi.org/10.1109/ACCESS.2020.3008321).
- [34] L. Mannila, F. Heintz, S. Kjällander, and A. Åkerfeldt. 2020. "Programming in primary education: towards a research based assessment framework". In *Proceedings of the 15th papershop on Primary and Secondary Computing Education (WIPSC'E 20)*. Association for Computing Machinery, New York, NY, USA, Article 10, 1–10. DOI:<https://doi.org/10.1145/3421590.3421598>
- [35] D. Statter and M. Armoni. 2020. "Teaching Abstraction in Computer Science to 7th Grade Students". *ACM Trans. Comput. Educ.* 20, 1, Article 8 (March 2020), 37 pages. DOI:<https://doi.org/10.1145/3372143>
- [36] Maya Israel , Moon Y. Chung , Quentin M. Werfel & Saddeddine Shehab (2020): A descriptive analysis of academic engagement and collaboration of students with autism during elementary computer science, *Computer Science Education*, DOI: [10.1080/08993408.2020.1779521](https://doi.org/10.1080/08993408.2020.1779521)
- [37] M. S Taylor, E. Vasquez, and C. Donehower. 2017. Computer Programming with Early Elementary Students with Down Syndrome. *Journal of Special Education Technology* 32, 3 (2017), 149–159. <https://doi.org/10.1177/0162643417704439>
- [38] Norman Powell, David Moore, John Gray, Janet Finlay & John Reaney (2004) Dyslexia and learning computer programming, *Innovation in Teaching and Learning in Information and Computer Sciences*, 3:2, 1–12, DOI: [10.11120/ital.2004.03020005](https://doi.org/10.11120/ital.2004.03020005)
- [39] Knight, V.F., Wright, J., Wilson, K. et al. Teaching Digital, Block-Based Coding of Robots to High School Students with Autism Spectrum Disorder and Challenging Behavior. *J Autism Dev Disord* 49, 3113–3126 (2019). <https://doi.org/10.1007/s10803-019-04033-w>
- [40] K. Eiselt and P. Carter, "Integrating Social Skills Practice with Computer Programming for Students on the Autism Spectrum", 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 2018, pp. 1–5, doi: [10.1109/FIE.2018.8659252](https://doi.org/10.1109/FIE.2018.8659252).
- [41] Almaiah, M.A., Al-Khasawneh, A. & Althunibat, A. Exploring the critical challenges and factors influencing the E-learning system usage during COVID-19 pandemic. *Educ Inf Technol* 25, 5261–5280 (2020). <https://doi.org/10.1007/s10639-020-10219-y>
- [42] Web Content Accessibility Guidelines (WCAG) 2.2. Charles Adams; Alastair Campbell; Rachael Bradley Montgomery; Michael Cooper; Andrew Kirkpatrick. W3C. (2020, August 11). W3C Working Draft. URL: <https://www.w3.org/TR/WCAG22/>

## **Apêndice D**

### **Parecer Comitê de Ética**

UFCG - HOSPITAL  
UNIVERSITÁRIO ALCIDES  
CARNEIRO DA UNIVERSIDADE  
FEDERAL DE CAMPINA  
GRANDE / HUAC - UFCG



### PARECER CONSUBSTANCIADO DO CEP

#### DADOS DO PROJETO DE PESQUISA

**Título da Pesquisa:** ENSINO DE PROGRAMAÇÃO PARA PESSOAS COM DEFICIÊNCIA

**Pesquisador:** ELAINE CRISTINA JUVINO DE ARAUJO

**Área Temática:**

**Versão:** 2

**CAAE:** 46773221.9.0000.5182

**Instituição Proponente:** Universidade Federal de Campina Grande

**Patrocinador Principal:** Financiamento Próprio

#### DADOS DO PARECER

**Número do Parecer:** 4.974.319

#### Apresentação do Projeto:

De acordo com o pesquisador o projeto descreve que o ensino de programação envolve diversas habilidades e pode ser bastante desafiador por não ser um conhecimento aprendido tradicionalmente ao longo da vida escolar. Pesquisas apontam que a inclusão de alunos atípicos em sala de aula é crescente, o que pode ser difícil para o professor gerenciar. Assim, este projeto de pesquisa propõe realizar um estudo longitudinal em estudantes com deficiência de cursos da área de computação do IFPB, da UFCG e UFPB com o propósito de analisar as habilidades cognitivas, bem como o desempenho de alunos com deficiência em disciplinas iniciais de programação. Os participantes são os estudantes com deficiência dos cursos da área de Computação do IFPB, UFCG e UFPB. O experimento consiste em acompanhar esses alunos ao longo do curso, aplicando testes de resolução de problemas de programação com e sem adaptações. Os dados obtidos serão analisados e relacionados com o desempenho dos alunos nas respectivas disciplinas. Esperamos como resultado: (i) identificar as habilidades relacionadas que os alunos já possuem antes de cursarem a disciplina inicial de programação; (ii) identificar as principais dificuldades enfrentadas pelos estudantes com deficiência nas disciplinas iniciais de programação; e (iii) analisar o impacto que adaptações direcionadas às deficiências podem trazer para o desempenho dos estudantes.

**Endereço:** Rua: Dr. Carlos Chagas, s/ n

**Bairro:** São José

**CEP:** 58.107-670

**UF:** PB

**Município:** CAMPINA GRANDE

**Telefone:** (83)2101-5545

**Fax:** (83)2101-5523

**E-mail:** cep@huac.ufcg.edu.br

UFCG - HOSPITAL  
UNIVERSITÁRIO ALCIDES  
CARNEIRO DA UNIVERSIDADE  
FEDERAL DE CAMPINA  
GRANDE / HUAC - UFCG



Continuação do Parecer: 4.974.319

**Objetivo da Pesquisa:**

O pesquisador refere como objetivo:

Objetivo Primário: Investigar os efeitos do uso de adaptação de conteúdo de programação direcionada à alunos com deficiência, no contexto de uma disciplina introdutória de programação em cursos de nível técnico ou superior da área de Computação.

**Avaliação dos Riscos e Benefícios:**

Foi elencado pela pesquisadora como riscos e benefícios:

Riscos:

Os participantes da pesquisa podem se sentir de alguma forma intimidados ou desconfortáveis em realizar os testes. Caso isso aconteça, esses podem desistir de participar da pesquisa a qualquer momento e receberão todo apoio necessário às suas dificuldades.

Benefícios:

Os benefícios e resultados esperados neste projeto seguem duas linhas. A primeira diz respeito a entender quais as dificuldades enfrentadas por alunos com deficiência ao entrar em um curso da área de Computação e cursar uma disciplina da área de programação, conhecendo os problemas de forma clara, podemos buscar formas de minimizá-los. A segunda linha diz respeito a o que fazer para que estudantes com deficiência consigam aprender a programar e prosseguir no curso, almejando ao fim conseguir uma profissão.

**Comentários e Considerações sobre a Pesquisa:**

A pesquisa denota relevância científica e social por propor melhorar o processo de ensino e aprendizagem de programação para alunos com deficiência em cursos de nível técnico e superior da área de Computação.

**Considerações sobre os Termos de apresentação obrigatória:**

Foram anexados ao sistema:

- Projeto completo
- Folha de rosto
- Orçamento
- Cronograma
- Roteiro de coleta de dados

**Endereço:** Rua: Dr. Carlos Chagas, s/ n  
**Bairro:** São José **CEP:** 58.107-670  
**UF:** PB **Município:** CAMPINA GRANDE  
**Telefone:** (83)2101-5545 **Fax:** (83)2101-5523 **E-mail:** cep@huac.ufcg.edu.br

UFCG - HOSPITAL  
UNIVERSITÁRIO ALCIDES  
CARNEIRO DA UNIVERSIDADE  
FEDERAL DE CAMPINA  
GRANDE / HUAC - UFCG



Continuação do Parecer: 4.974.319

- Termo de Assentimento Livre e Esclarecido
- Termos de Consentimento Livre e Esclarecido
- Termo de Compromisso dos Pesquisadores
- Termos de Anuências

**Conclusões ou Pendências e Lista de Inadequações:**

Não existem inadequações éticas para o início da pesquisa.

**Considerações Finais a critério do CEP:**

**Este parecer foi elaborado baseado nos documentos abaixo relacionados:**

Tipo Documento	Arquivo	Postagem	Autor	Situação
Informações Básicas do Projeto	PB_INFORMAÇÕES_BÁSICAS_DO_PROJETO_1705178.pdf	05/08/2021 15:24:02		Aceito
Projeto Detalhado / Brochura Investigador	ProjetoComiteEticaElaine.pdf	05/08/2021 15:22:18	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	TCLE.pdf	05/08/2021 15:22:01	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
TCLE / Termos de Assentimento / Justificativa de Ausência	TCLE_Responsavel.pdf	05/08/2021 15:21:55	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
Outros	Anuencia_IFPB.pdf	06/05/2021 15:51:01	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
Outros	TERMODECOMPROMISSOELAINE.pdf	06/05/2021 15:48:43	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
Outros	Anuencia_UFPB.pdf	06/05/2021 15:01:11	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
Outros	autorizacaodosifpb.pdf	08/04/2021 06:34:06	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
Declaração de concordância	Anuencia_UFCG_ElaineWilkerson_assinado.pdf	16/03/2021 14:49:47	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito
Folha de Rosto	FolhadeRosto_ElaineWilkerson.pdf	16/03/2021 14:47:01	ELAINE CRISTINA JUVINO DE ARAUJO	Aceito

**Situação do Parecer:**

**Endereço:** Rua: Dr. Carlos Chagas, s/ n  
**Bairro:** São José **CEP:** 58.107-670  
**UF:** PB **Município:** CAMPINA GRANDE  
**Telefone:** (83)2101-5545 **Fax:** (83)2101-5523 **E-mail:** cep@huac.ufcg.edu.br

UFCG - HOSPITAL  
UNIVERSITÁRIO ALCIDES  
CARNEIRO DA UNIVERSIDADE  
FEDERAL DE CAMPINA  
GRANDE / HUAC - UFCG



Continuação do Parecer: 4.974.319

Aprovado

**Necessita Apreciação da CONEP:**

Não

CAMPINA GRANDE, 14 de Setembro de 2021

---

**Assinado por:**

**Andréia Oliveira Barros Sousa**  
(Coordenador(a))

**Endereço:** Rua: Dr. Carlos Chagas, s/ n

**Bairro:** São José

**CEP:** 58.107-670

**UF:** PB

**Município:** CAMPINA GRANDE

**Telefone:** (83)2101-5545

**Fax:** (83)2101-5523

**E-mail:** cep@huac.ufcg.edu.br

## **Apêndice E**

# **Termo de Consentimento Livre e Esclarecido**



COMITÊ DE ÉTICA EM PESQUISA EM SERES HUMANOS - CEP  
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG  
HOSPITAL UNIVERSITÁRIO ALCIDES CARNEIRO – HUAC



## TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

ESTUDO: Ensino de Programação para Pessoas com Deficiência

*Você está sendo convidado (a) a participar do projeto de pesquisa acima citado. O documento abaixo contém todas as informações necessárias sobre a pesquisa que estamos fazendo. Sua colaboração neste estudo será de muita importância para nós, mas se desistir a qualquer momento, isso não causará nenhum prejuízo a você.*

Eu, (inserir o nome, profissão, residente e domiciliado na ....., portador da Cédula de identidade, RG ....., e inscrito no CPF/MF..... nascido(a) em \_\_\_\_ / \_\_\_\_ /\_\_\_\_, abaixo assinado(a), concordo de livre e espontânea vontade em participar como voluntário(a) do estudo “**Ensino de Programação para Pessoas com Deficiência**” e que está sob a responsabilidade da aluna de Doutorado do Programa de Pós-Graduação em Computação da UFCG Elaine Cristina Juvino de Araújo ([elaine@copin.ufcg.edu.br](mailto:elaine@copin.ufcg.edu.br)) e que está sob orientação do professor Wilkerson L. Andrade ([wilkerson@computacao.ufcg.edu.br](mailto:wilkerson@computacao.ufcg.edu.br)).. Declaro que obtive todas as informações necessárias, bem como todos os eventuais esclarecimentos quanto às dúvidas por mim apresentadas.

Estou ciente que:

Esse projeto tem como objetivo investigar os efeitos do uso de adaptações de conteúdo no contexto de uma disciplina de programação introdutória. A pesquisa está sendo realizada no contexto das disciplinas iniciais de programação na UFCG, IFPB e UFPB. Os resultados deste trabalho servirão como base para melhorar as práticas de ensino de programação para pessoas com deficiência. A pesquisa apresenta benefícios diretos para os participantes, já que para a coleta de dados, serão realizados estudos que lhes farão refletir sobre o processo de aprendizagem de programação, algo que poderá auxiliar no entendimento dos conteúdos.

A coleta de dados poderá ocorrer através do uso de dados históricos na disciplina e através de exercícios com duração de até 2 horas, de forma presencial ou online. Os dados serão coletados através da análise de código gerado e da observação e aplicação de questionários e entrevistas.



COMITÊ DE ÉTICA EM PESQUISA EM SERES HUMANOS - CEP  
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG  
HOSPITAL UNIVERSITÁRIO ALCIDES CARNEIRO – HUAC



Caso este termo seja entregue de forma impressa e presencial, e contenha informações incompreensíveis, as dúvidas podem ser sanadas com a entrevistadora e apenas ao final, quando todos os esclarecimentos forem fornecidos, caso concorde com a participação do estudo, irei rubricar as folhas e assinar ao final deste documento, que está em duas vias. Uma via me será entregue e a outra ficará com a pesquisadora responsável. Se a minha participação for online, uma via deste termo poderá ser enviada para o meu email e a entrevistadora estará disponível no endereço acima citado para sanar quaisquer dúvidas.

Caso não concorde, não haverá penalização, bem como será possível retirar o consentimento a qualquer momento, sem qualquer penalidade. Caso aceite participar, receberá todo apoio necessário para sanar suas dúvidas e auxiliar em suas dificuldades. Se se sentir de alguma forma intimidado ou desconfortável em realizar os testes você poderá desistir de participar da pesquisa a qualquer momento.

Atestado de interesse pelo conhecimento dos resultados da pesquisa

- ( ) Desejo conhecer os resultados desta pesquisa  
( ) Não desejo conhecer os resultados desta pesquisa.

Caso me sinta prejudicado (a) por participar desta pesquisa, poderei recorrer ao Comitê de Ética em Pesquisas com Seres Humanos – CEP, do Hospital Universitário Alcides Carneiro - HUAC, situado a Rua: Dr. Carlos Chagas, s/ n, São José, CEP: 58401 – 490, Campina Grande-PB, Tel: 2101 – 5545, E-mail: cep@huac.ufcg.edu.br; Conselho Regional de Medicina da Paraíba e a Delegacia Regional de Campina Grande.

Campina Grande - PB, \_\_\_\_ de \_\_\_\_\_ de 2021.

( ) Participante/( ) Responsável .....

Testemunha 1 : \_\_\_\_\_  
Nome / RG / Telefone

Testemunha 2 : \_\_\_\_\_  
Nome / RG / Telefone



COMITÊ DE ÉTICA EM PESQUISA EM SERES HUMANOS - CEP  
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG  
HOSPITAL UNIVERSITÁRIO ALCIDES CARNEIRO – HUAC



Responsável pelo Projeto:

**Elaine Cristina Juvino de Araújo** (83) 99624-7025/  
elaine@copin.ufcg.edu.br

## **Apêndice F**

# **Termo de Consentimento Livre e Esclarecido Responsável Menor**



COMITÊ DE ÉTICA EM PESQUISA EM SERES HUMANOS - CEP  
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG  
HOSPITAL UNIVERSITÁRIO ALCIDES CARNEIRO – HUAC



**TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO**  
(Responsável menor)

**ESTUDO: Ensino de Programação para Pessoas com Deficiência**

*Você está sendo convidado (a) a autorizar a participação do menor sob sua responsabilidade do projeto de pesquisa acima citado. O documento abaixo contém todas as informações necessárias sobre a pesquisa que estamos fazendo. A colaboração neste estudo será de muita importância para nós, mas se desistir a qualquer momento, isso não causará nenhum prejuízo a vocês.*

Eu, (inserir o nome, profissão, residente e domiciliado na ....., portador da Cédula de identidade, RG ....., e inscrito no CPF/MF..... nascido(a) em \_\_\_\_ / \_\_\_\_ / \_\_\_\_\_, abaixo assinado(a), autorizo o menor sob minha responsabilidade, de livre e espontânea vontade, em participar como voluntário(a) do estudo “**Ensino de Programação para Pessoas com Deficiência**” e que está sob a responsabilidade da aluna de Doutorado do Programa de Pós-Graduação em Computação da UFCG Elaine Cristina Juvino de Araújo ([elaine@copin.ufcg.edu.br](mailto:elaine@copin.ufcg.edu.br)) e que está sob orientação do professor Wilkerson L. Andrade ([wilkerson@computacao.ufcg.edu.br](mailto:wilkerson@computacao.ufcg.edu.br)).. Declaro que obtive todas as informações necessárias, bem como todos os eventuais esclarecimentos quanto às dúvidas por mim apresentadas.

Estou ciente que:

Esse projeto tem como objetivo investigar os efeitos do uso de adaptações de conteúdo no contexto de uma disciplina de programação introdutória. A pesquisa está sendo realizada no contexto das disciplinas iniciais de programação na UFCG, IFPB e UFPB. Os resultados deste trabalho servirão como base para melhorar as práticas de ensino de programação para pessoas com deficiência. A pesquisa apresenta benefícios diretos para os participantes, já que para a coleta de dados, serão realizados estudos que lhes farão refletir sobre o processo de aprendizagem de programação, algo que poderá auxiliar no entendimento dos conteúdos.

A coleta de dados poderá ocorrer através do uso de dados históricos na disciplina e através da aplicação de exercícios com duração de até 2 horas, de forma presencial ou online. Os dados serão coletados através da análise de código gerado e da aplicação de questionários e entrevistas.



COMITÊ DE ÉTICA EM PESQUISA EM SERES HUMANOS - CEP  
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFPG  
HOSPITAL UNIVERSITÁRIO ALCIDES CARNEIRO – HUAC



Caso este termo seja entregue de forma impressa e presencial, e contenha informações incompreensíveis, as dúvidas podem ser sanadas com a entrevistadora e apenas ao final, quando todos os esclarecimentos forem fornecidos, caso concorde com a participação do estudo, irei rubricar as folhas e assinar ao final deste documento, que está em duas vias. Uma via me será entregue e a outra ficará com a pesquisadora responsável. Se a participação for online, uma via deste termo poderá ser enviada para o meu email e a entrevistadora estará disponível no endereço acima citado para sanar quaisquer dúvidas.

Caso não concorde, não haverá penalização, bem como será possível retirar o consentimento a qualquer momento, sem qualquer penalidade. Caso aceite permitir a participação do menor, receberá todo apoio necessário para sanar suas dúvidas e auxiliar em suas dificuldades. Se você ou o menor se sentir de alguma forma intimidado ou desconfortável em realizar os testes poderá desistir de participar da pesquisa a qualquer momento.

Atestado de interesse pelo conhecimento dos resultados da pesquisa

- ( ) Desejo conhecer os resultados desta pesquisa  
( ) Não desejo conhecer os resultados desta pesquisa.

Caso me sinta prejudicado (a) por participar desta pesquisa, poderei recorrer ao Comitê de Ética em Pesquisas com Seres Humanos – CEP, do Hospital Universitário Alcides Carneiro - HUAC, situado a Rua: Dr. Carlos Chagas, s/ n, São José, CEP: 58401 – 490, Campina Grande-PB, Tel: 2101 – 5545, E-mail: cep@huac.ufcg.edu.br; Conselho Regional de Medicina da Paraíba e a Delegacia Regional de Campina Grande.

Campina Grande - PB, \_\_\_\_\_ de \_\_\_\_\_ de 2021.

( \_\_\_\_\_ ) Participante / ( \_\_\_\_\_ ) Responsável  
.....

Testemunha 1 : \_\_\_\_\_  
Nome / RG / Telefone



COMITÊ DE ÉTICA EM PESQUISA EM SERES HUMANOS - CEP  
UNIVERSIDADE FEDERAL DE CAMPINA GRANDE - UFCG  
HOSPITAL UNIVERSITÁRIO ALCIDES CARNEIRO – HUAC



Testemunha 2 : \_\_\_\_\_  
Nome / RG / Telefone

Responsável pelo Projeto:  
**Elaine Cristina Juvino de Araújo (83) 99624-7025/**  
**elaine@copin.ufcg.edu.br**

# **Apêndice G**

## **Perceptions about Teaching**

### **Programming in the Neurodiverse**

#### **Students' Context**

# Perceptions about Teaching Programming in the Neurodiverse Students' Context

Elaine Cristina Juvino de Araújo  
Federal Institute of Paraíba  
Federal University of Campina Grande  
Campina Grande, PB, Brazil  
elainec@ifpb.edu.br

Wilkerson L. Andrade  
Federal University of Campina Grande  
Campina Grande, PB, Brazil  
wilkerson@computacao.ufcg.edu.br

Ana Liz Souto Oliveira  
Federal University of Paraíba  
Rio Tinto, PB, Brazil  
analiz@dcx.ufpb.br

**Abstract**—This research full paper presents the perception of programming teachers and students in the context of neurodiverse students. Among the wide range of disabilities, Neurodevelopmental Disorders encompass types of disabilities that are not easily visible, such as Autism Spectrum Disorder (ASD) and Attention Deficit Hyperactivity Disorder (ADHD). A section of the literature does not consider these disorders to be disabilities. Therefore, neurodiverse describes a person whose brain functions differently than typical. Given this, we can look at these students as people who learn differently and who need to have their specific needs met. With the inclusion of students with disabilities in technical and undergraduate courses, they face several challenges, both for these students and teachers. Specifically, several peculiarities can impact this process regarding teaching programming, such as the creation of algorithms based on problems described in extensive texts. Despite efforts to meet the needs of students with disabilities in teaching programming, we still do not have enough studies that address this issue. The main objective of this research is to investigate the current reality of teaching programming to neurodiverse people, what the difficulties are, and how teachers and students are trying to solve them. Thus, given the phenomenon's complexity, we created a case study with two units of analysis: (1) neurodiverse students and (2) their respective programming teachers. The phenomenon studied was programming teaching, and the context was the first programming discipline of two undergraduate courses and a high school technical course. We carried out semi-structured interviews with six students and three teachers. We identified through research that teachers need adequate training to serve atypical students and that educational institutions still need an efficient inclusion process for these students. When we observe the students' vision, we realize that for them to learn to program, there are essential demands, such as the difficulty in understanding the application of programming concepts in practice and the creation of programs without assistance. This study is necessary to show the demand for new approaches to meet the specific needs of neurodiverse students in programming disciplines. To achieve this, inclusive pedagogical strategies and resources must be created and adapted to the characteristics and abilities of these students. Initiatives like this can promote equal opportunities in the classroom, including accessible learning environments, use of explicit language, and individualized support, among others.

**Index Terms**—teaching programming, students with disabilities, student perceptions

## I. INTRODUCTION

We can describe that people with disabilities have some type of impairment, be it physical, mental, intellectual, or sensory [15]. To consider a permanent disability, the person must have

a long-term disability. However, we can also have short-term impediments, such as someone temporarily losing mobility due to an accident. What characterizes disability is that the impediment can cause some type of difficulty for the person, whether permanent or temporary.

Among the different types of disabilities, we have those that are not easily visible, such as neurodevelopmental disorders [1] or cognitive disabilities. These deficiencies can affect brain development, impacting cognitive, emotional, and social skills. Some common examples of neurodevelopmental disorders include Autism Spectrum Disorder (ASD), Attention Deficit Hyperactivity Disorder (ADHD), and Intellectual Disability (ID).

Difficulties in social interaction, communication, and repetitive behaviors characterize ASD. ADHD involves symptoms of inattention, hyperactivity, and impulsivity. Individuals with ADHD may have difficulty concentrating, following instructions, and controlling their impulses, which can interfere with their academic and social performance. Individuals with ID may experience difficulties understanding concepts, reasoning, solving problems, and developing social skills [1].

Judy Singer created the term neurodiversity [8], which argues that neurodiverse people would not be people with disabilities to be treated, but rather people whose brains function in an atypical way. This author's view implies that if people have a brain that works differently, they need different strategies to deal with it, such as education.

The problem involving students with cognitive disabilities or neurodiverse people in teaching programming is a complex and challenging issue. Programming requires specific cognitive skills, such as logical reasoning, abstraction capacity, and problem-solving, which can be affected in different ways in these individuals [2]. The lack of adequate adaptations and resources can make it difficult for these students to fully and effectively learn. Traditional teaching approaches may not be flexible enough to meet your individual needs, resulting in accessibility and inclusion challenges. Furthermore, accessibility initiatives are traditionally designed to serve people with sensory or physical disabilities (vision, hearing, or mobility) [9], [18].

The Brazilian Inclusion Law guarantees the rights of people with disabilities [3]. Among these rights, access to education

stands out, which is guaranteed, considering these characteristics, interests, and learning needs.

This study aims to portray the perceptions of neurodiverse students and their teachers in teaching programming. More specifically, we discuss the main learning problems identified in programming classes by neurodiverse students and their teachers and how they are trying to solve them.

To achieve this objective, we established three research questions to be answered:

- RQ1 What are the difficulties of neurodiverse students concerning the programming class?
- RQ2 What is the perception of programming teachers regarding their work with neurodiverse students?
- RQ3 How are programming teachers acting towards neurodiverse students?

This research used a case study to understand this scenario due to the complexity of the phenomenon studied. The scenario is complex because Neurodevelopmental disorders are seen as non-visible disabilities. Therefore, we only use students who have a medical report confirming their neurodiversity, which makes our sample small. We collected data through interviews at two public educational institutions, with neurodiverse students from technical and undergraduate courses in Computing and their respective teachers from the first programming discipline.

As a contribution to this study, we highlight the presentation of an urgent demand for new approaches to serving neurodiverse students in programming disciplines. Even though it is a small sample, we get a realistic view of the experience of neurodiverse students in programming classes. In addition to the students' vision, we can also see the concern of teachers, who want to help these students but often need help knowing where to start because they often don't have adequate training.

We organize this paper as follows: Section II presents the Background, Section III presents related work, Section IV describes the methodology, and Section V presents the results. We discuss the results in Section VI, and in Section VII, we present threats to the validity of this research. Finally, Section VIII presents the conclusion and future works, and Section **anonymous** presents Acknowledgements.

## II. BACKGROUND

Some concepts found in the literature are important for this research. We must understand teaching programming and the characteristics and particularities of some of the main cognitive disabilities that come into the context of neurodiversity.

### A. Programming Teaching

Programming is an essential part of Computer Science (CS) and is a knowledge that demands dedication to learn. Programming disciplines are often considered complex. Besides, teachers constantly search for strategies and approaches to generate better student achievement. In general, the difficulties faced are related to understanding concepts that are not clear in theory but become through practice [25].

It is necessary to understand the concept of algorithms to learn how to program. An algorithm is a sequence of well-defined steps that can receive values as input and generate an output [12]. We can create different solutions to solve a particular problem, but some may be simpler or faster than others.

Some methods or approaches to teaching programming concern how teachers deliver knowledge to the students. One way to teach traditional programming is Thompson's approach [26]. This approach presents the following steps to solve a programming exercise: (1) understand the problem, (2) plan the solution, (3) implement the program, and (4) review. These steps can be easy and natural for students to solve their programming problems. However, the entire process can be challenging for some beginning students, increasing the need for guidance and assistance.

### B. Cognitive Disabilities and Neurodiversity

Cognitive Impairment is a generic term in the literature to discuss mental disabilities. The most common cognitive impairments include Neurodevelopmental Disorders such as Autism Spectrum Disorder (ASD), Attention Deficit Hyperactivity Disorder (ADHD), and Specific Learning Disorders such as Dyslexia [27]. Koushik and Kane [23] declare that little research still focuses on teaching informatics to people with diverse cognitive abilities. There are also few reports of strategies for this audience, which shows us a considerable gap to be explored.

An interesting approach came from Australian Judy Singer, who coined neurodiversity. Drawing a parallel with biodiversity, she showed that all human beings have their own brain characteristics. Therefore, being neurotypical, autistic, or having any other neurodevelopmental disorder is just having different brain characteristics, as well as ethnicity or gender [8].

1) *Autism*: Autism Spectrum Disorder (ASD) is part of the set of neurodevelopmental disorders and can affect individuals mainly in communication and repetitive behaviors [1]. It is a disorder with three different levels of severity. The most severe level is level 3, and the mildest level 1. According to each level, the autistic person can have more or less resistance, mainly in social interactions. Another important characteristic is the need to maintain a well-defined routine, making sudden changes painful for the autistic person.

Autism can manifest itself in different ways and will not permanently impair the intelligence of individuals, including research indicating that approximately half of autistics have average or above average intelligence [28]. Over the years, we have had more and more diagnoses of autism, which implies that there is an increase in autistic people entering the school environment, including higher education [29].

2) *Attention Deficit Hyperactivity Disorder*: ADHD is another neurodevelopmental disorder characterized by dysregulation in the form of inattention, disorganization, hyperactivity, and impulsivity. A person with ADHD may have only inattention, hyperactivity, or both as the central dysfunction.

This disorder usually appears in childhood and continues into adulthood, which can cause impairments in social, academic, and professional functioning. ADHD can manifest in three ways: (1) hyperactivity alone, (2) just attention deficit, and (3) hyperactivity and inattention combined [1].

Students with ADHD are more likely to have learning difficulties than neurotypicals and deficits in reading and math. This common comorbidity can reach up to 45% of those with this disorder [30].

3) *Learning Disorders*: Learning disorders can be presented by various difficulties throughout school life, such as dyslexia, which has characteristics of impaired word recognition and spelling. You may also have problems with reading and mathematical reasoning [1].

While learning disabilities can come from specific learning disorders such as dyslexia or other disorders such as ADHD [31], the specific learning disorder can affect people without another disorder. In some cases, they may even have high levels of intelligence but have academic difficulties due to psychological deficits.

### III. RELATED WORKS

Academic research has explored different approaches to teaching programming to different audiences. A recent study [2] aimed to investigate the impact of cognitive disabilities on the skills needed to learn programming. To this end, the primary skills involved in programming were identified, cataloged, and compared with the characteristics of some of the primary cognitive disabilities. The study revealed that programming requires various mathematical and logical skills. Furthermore, it has been observed that different types of cognitive impairment can affect the skills needed to program differently, and not all skills are necessarily affected by all types of impairment. These findings highlight the importance of inclusive and adaptive approaches to teaching programming to meet the needs of neurodiverse students or students with cognitive disabilities.

The difference between the research in [2] and this is that in the work proposed, we visualize the intersection between the skills involved in programming and the characteristics of cognitive disabilities. This analysis does not consider perceptions of neurodiverse people and their teachers, seeking to make its analysis based on literature.

Koushik and Kane's paper [23] presents observations about a program for teaching computing concepts to young adults with cognitive disabilities. They conducted interviews, observations, and classes for two instructors and 10 students. They used Universal Design concepts to learn but stated that the practices were done by trial and error. The work presented in [23] differs from this in that it analyzes the perceptions of students with cognitive disabilities and their instructors, but it collects data from a course that was already created to be a course to teach programming to this audience, so the course would already have an accessibility proposal. In our research, we sought the perceptions of students and teachers in secondary and undergraduate courses, which were

not designed with any prior accessibility recommendations in mind.

Silva et al. [4] offered an interesting perspective on teaching computational thinking through learning algorithms and programming for individuals with ADHD. A case study was conducted in the format of a 34-week course, focusing on strategies for teaching programming to a student with ADHD. The authors established as a success criterion that the student achieved a minimum accuracy rate of 80% in the assessment as an indicator of successful learning in programming. At the end of the course, the student achieved an accuracy rate of 84%, suggesting the experiment's effectiveness. Although the experience presented is relevant, it is essential to highlight that it was carried out as an extracurricular course for just one student. In addition to this, we still see a dearth of positive evidence for students with ADHD in regular Computer Science courses.

The difference between the paper presented in [4] and this research is that an extra-curricular course was also monitored, which already had in mind recommendations for accessibility for a student with ADHD. Our research starts from perceptions of courses not designed to be accessible.

The study by Israel et al. [6] analyzed the engagement of students with autism in a course that covered Computer Science concepts. This case study involved the collaborative participation of three elementary school students with Autism Spectrum Disorder (ASD) in an extracurricular course. As a result, we observed that student engagement could have been more varied, often low, with little problem-solving. The researchers also found that the experience could have been more positive if there had been individualized support for each student. They observed that students had difficulty persisting and received little help, in addition to the teacher's lack of help planning.

In [6], it is another case of an extracurricular course for K12 students, which is valid for observing student engagement but may not reflect the same experience of students on a regular course in Computing.

A paper [22] presents an educational game for teaching programming to children with ADHD. The objective was to teach programming to children in a playful way. Initially, they planned to create a game using the Scratch platform but could not, so they created it using the Unity 3D platform. The game has not been tested with the target audience, and we do not know if it works effectively with children with ADHD.

Haynes-Magyar performed an exploratory study [17] with five neurodiverse students. Students studied eight chapters of a digital Python programming book and solved problems. From this, the students reported through interviews what their accessibility recommendations would be for Parsons' problems. This study has limitations, such as not being able to generalize its results since we have a small and specific sample. Furthermore, the research is focused only on students' opinions and does not address teachers' opinions and strategies.

A paper [24] deals with the mental health of 53 students in

the field of Computing, who shared their positive and negative experiences. In this way, the work proposed presents these points by showing that some students believe that mental symptoms such as stress and panic attacks can affect their academic performance. They also believe that mental health conditions, because they are invisible, make it difficult for teachers and instructors to assist. For these reasons, students believe that it is necessary for teachers and instructors to actively observe their classes to try to identify students with mental health issues, as it is difficult for students to deal with the issue spontaneously.

The paper [24] differs from this research for several reasons. As for the participants, data was collected from students who identified themselves as having some mental health condition, which implies that they did not ask for proof of medical reports. It also did not collect data from these students' respective teachers and was not limited to specific content like us, who focused on the programming course. Furthermore, he collected information to find out how the course affected students' mental health, which is not the focus of our work.

#### IV. METHODOLOGY

This research has an exploratory and descriptive nature [7]. We aim to understand the current state of teaching programming to neurodiverse students. Participants in this research belong to one of two profiles: (1) neurodiverse students taking their first programming subject or (2) their respective teachers. This choice was made due to the complexity of the phenomenon researched, as we only considered students with a confirmed diagnosis of neurodevelopmental disorder, even though we know that some students may present characteristics of certain neurodiversity. Here, we have a single-case exploratory case study [10], [19], [20].

For this study, we considered the teaching of programming as a phenomenon. As a context, we have the first programming class in two undergraduate courses and a high school technical course. The units of analysis of this case study are (1) neurodiverse students and (2) their respective teachers within this context.

This research follows the ethical principles established by the **anonymous** by Resolution **anonymous** which supports research with human beings and is approved by the **anonymous**.

##### A. Participants Profile

For this study, we need to investigate the impressions of neurodiverse students and their respective teachers in the context of the first programming course. The research participants were, therefore, students with some diagnosis of neurodiversity enrolled in a course in the Computing area at a higher level or a technical course at a secondary level. In addition to these, we also selected their respective teachers in the first programming course.

Student participants were selected through contact with two educational institutions, seeking to locate students who had diagnoses issued by specialized professionals. In this context, the institutions had these diagnoses, which were received upon

admission of students to the respective selection processes. We know that some students may present characteristics of neurodiversity but do not have a report or do not want to be classified as such. As we do not have professional qualifications to diagnose them, we chose to select only those who already have a report and, of their own free will, agreed to participate in the research.

As our research field is very restricted, we established that all students who fit the profile and were enrolled in a first programming course would be selected. Six students and three teachers were located: five undergraduates and one high school technical course student. To preserve students' privacy, we identify them using an acronym, where undergraduate students are identified by the letter G followed by a control number, and the acronym T1 identifies technical course students.

Regarding diagnoses, it is possible to observe a diversity among them, although they all fall into the category of cognitive disability or neurodiversity as shown in Table I.

TABLE I  
STUDENTS PROFILE

Pseudonym	Diagnose
G1	Asperger's Syndrome (ASD)
G2	Asperger's Syndrome (ASD)
G3	Asperger's Syndrome (ASD) and ADHD
G4	ADHD
G5	ADHD and Mood Disorder
T1	Noonan Syndrome, ADHD, and mild ID

It is important to mention that what was previously known as Asperger's Syndrome is now classified within Autism Spectrum Disorder, according to the Diagnostic and Statistical Manual of Mental Disorders (DSM-5) [1]. Furthermore, Mood Disorder is included in Attention Deficit Hyperactivity Disorder (ADHD). On the other hand, Noonan Syndrome is a physical condition, but the student in question also with ADHD and Intellectual Disability (ID), which qualified him to participate in the study.

After identifying students eligible for the study, we contacted each student's teacher in the first programming subject. As we can see in Table II, we have teachers with training in Computing and extensive experience in teaching.

TABLE II  
TEACHERS PROFILE

Pseudonym	Graduation	Highest Academic Degree	Teaching Experience
TG1	Computer Science	Master Degree	11 Years
TG2	Computer Science	PhD	17 Years
TT1	Informatics	Postgraduate Specialization	16 Years

##### B. Data collect

Data collection took place through semi-structured interviews; in this way, we sought to obtain information from the

participant's perspective [16]. We created a script for each unit of analysis, dividing the questions into parts, one for neurodiverse students and the other for their teachers. To this end, the question sought to guide interviewees in presenting as much relevant information as possible to this research. The interviews were carried out individually at different times, outside the classroom, to make participants feel comfortable throughout the process.

The teachers' interview script was divided into three parts, as shown in Table III. Part 1 sought to collect the participants' profiles with their training data and whether any training supported classes involving students with disabilities. In part 2, we seek to detail the teacher's experience with students with disabilities in the first programming course. In part 3, we finish the interview by collecting data about the discipline.

The interview guide with students was organized into six parts, as shown in Table IV. Part 1 collects name and age data, which only serves as a control. The participant's name will not be disclosed, preserving their privacy. In part 2, we open space for the student to talk about their main doubts without reminding them of the content so that they can try to explain on their own. Part 3 specifies the main contents covered in the first programming course. The opportunity allowed the students to recall the content and report their doubts and understanding about each of them. Part 4 seeks to understand students' difficulties when doing programming exercises. Part 5 aimed to understand the difficulties in assessments. Finally, in part 6, we allowed the students to express anything important to conclude the interview.

TABLE III  
TEACHER INTERVIEW GUIDE

<b>Part 1 Participant's Profile</b>
Graduation;
How many years of classroom experience;
Experience involving students with disabilities in the classroom;
Training for Inclusive Education;
<b>Part 2 Experience with Students with Disabilities</b>
Students with disabilities from the programming course in previous semesters;
Types of Disabilities;
Details about students with disabilities this semester;
The adaptations made in classes for students with disabilities;
The adaptations made in the materials and activities for students with disabilities;
Interested in learning how to assist these students in programming lessons.
<b>Part 3 Conducting Discipline</b>
Detail how the discipline is currently conducted

## V. RESULTS

We performed a content analysis of the data collected during the interviews [11], [13], [14]. Our objective was to observe how the teaching and learning process occurs in the context of neurodivergent students in the first programming subject.

We interviewed six students who reported some neurodevelopmental disorder and their three respective teachers from the first programming subject. As one of the teachers had more

TABLE IV  
STUDENT INTERVIEW GUIDE

<b>Part 1 Participant Profile</b>
Name; Age.
<b>Part 2 Questions about Programming</b>
Detail programming doubts
<b>Part 3 Contents</b>
Input and output; Variables; Definition of Algorithm; Conditional Structures; Repetition Structures; Vectors; Functions.
<b>Part 4 Exercises</b>
Statement; Input; Exit
<b>Part 5 Assessment</b>
<b>Part 6 General Information</b>

than one student who fit into the study, we observed that the number of teachers was smaller than the number of students.

The data analysis was manual because the sample was small. We carefully read the interview responses and compiled the results according to the research questions below.

### A. RQ1 What are the difficulties of neurodiverse students concerning the programming class?

The first question to the students (Interview Guide - Part 2) was to seek information about the questions they had in programming, but without mentioning any specific content. We aimed to extract difficulties neutrally because they would inform the interviewer if the student had a particular doubt. In Table V, we see how students responded about their difficulties in programming without any reminders about the programming topics.

TABLE V  
PROGRAMMING DOUBTS

Difficulties	G1	G2	G3	G4	G5	T1
Exercises	X					
Materials	X					
Algorithms Application		X				
Looping				X		
No Doubt			X		X	
Doubt about almost everything						X

Observing the answers about difficulties in general, students gave varied answers, but we see that of the total number of students interviewed, two believe that they do not have difficulties in programming. Only one student believes that he has difficulty in all the topics mentioned.

After the student spontaneously expressed their doubts about programming, we asked them about their difficulties concerning programming topics. Based on each topic mentioned, the student said whether and what their question would be. In Table VI, we see how students answered their questions in the programming topics. It is marked with "X" if the student has questions about the topic asked.

We can observe that, of the students who responded that they had doubts, none indicated that they had doubts about the initial topics, input-output, and variables. Regarding functions' content, some students have not yet studied this topic.

TABLE VI  
PROGRAMMING DOUBTS BY TOPICS

Topics	G1	G2	G3	G4	G5	T1
Input/Output						X
Variables						X
Algorithm Definition					X	X
Condition	X			X		X
Loops	X			X	X	X
Vectors	X			X	X	X
Functions	X					

Then, the students responded about their difficulties in the exercises and assessments (Interview Guide Parts 5 and 6). Five of the six students interviewed reported that their greatest difficulty was understanding the questions statements. This difficulty is also reported concerning evaluations. One of the students also highlighted that many exercises involve mathematical calculations as a difficulty.

*B. RQ2 What is the perception of programming teachers regarding their work with neurodiverse students?*

Three programming teachers were interviewed to answer this question. In the interview, we asked each teacher about their experience with students with disabilities. All teachers interviewed were teachers of at least one of the neurodiverse students we also interviewed. All interviewees responded that they had students with disabilities in previous semesters, received students with different types of needs, and did not have any specific training to help these students.

The TG1 teacher noticed the difficulty of his neurodiverse student, especially in attention and memorization, looking for strategies to help him. The TG2 teacher, who has 4 neurodiverse students, realized that each student has different levels of interaction and attention to the subject. In the same way that there are neurodiverse students who perform well and interact well, there are also neurodiverse students who do not interact and do not seek to answer questions, harming their performance in the subject. Teacher TT1 noticed that his neurodiverse student has a lot of difficulties, low interaction, and a lack of attention.

Finally, all teachers interviewed believe that they need adequate training to deal with neurodiversity, which falls under non-visible deficiencies, and that educational institutions still need to improve support for teachers in this context.

*C. RQ3 How are programming teachers acting towards neurodiverse students?*

Based on the teachers' perception of neurodiverse students, related in RQ2, another important question arises, which is how these teachers try to support neurodiverse students, even without having adequate training. Given this, we have essential information that helps us reflect on how more than traditional practices may be needed to teach these students, as shown in Table VII.

Professor TG1 reported an attempt to increase the students' workload, making them use two semesters instead of one to complete all the programming content.

TABLE VII  
TEACHER'S STRATEGIES

Strategy	TG1	TG2	TT1
Increase in course workload	X		
Increased time to carry out activities or assessments	X	X	X
Monitor Use	X	X	X
Constant intervention	X		
Repetition of the explanation		X	X
Use of Images			X

All teachers indicated they sought to increase the deadline for completing exercises and activities for neurodiverse students. They also reported that they have a monitor to help the student.

Only the TG1 teacher sought constant intervention for the student as a strategy, which could be from himself or the monitor. Two of the three teachers used the trick to explain the same thing more than once, seeking understanding from neurodiverse students. Only teacher TT1 reported that he sought to adapt his class material, using more images and less text as much as possible.

## VI. DISCUSSION

By evaluating the interview data, we gained insights from each of our units of analysis, guided by the research questions, which are discussed below.

*A. Perceptions of the Difficulties of neurodiverse students in Programming classes*

The students interviewed pointed out difficulties in programming classes, generating reflection on what could be done to make these students successful in the course. We can point out some key questions below:

- **Materials and Exercises** - The students point out difficulties in the class materials and exercises, generating reflection on how these could be adapted to a neurodiverse audience. What would be the points to improve?
- **Understanding the application of Programming** - It was mentioned by the students that there is difficulty in understanding the application of Algorithms in the real world. This difficulty signals the need for greater contextualization between Programming concepts and the practical needs of the market, for example.
- **Interpretation of problems** - Students claimed difficulty in interpreting what the problems ask. This occurs both in exercises and in assessments. It is a fact that, in some types of cognitive disability, we have some characteristics that can impact, for example, logical reasoning and abstraction [2], which would require a reformulation in the way questions are written.
- **Creating Algorithms without help** - Another critical problem identified in the responses was the difficulty in knowing how to start an algorithm without help, which may suggest the need for devices to remind them of the structure and main elements of an algorithm within the scope of the problem they want to solve at that moment.

- **Remembering what to do in exercises and assessments** - Difficulty was reported in knowing which programming elements should be used for each problem situation.

The issues identified during data collection provide strong evidence that it is necessary to re-evaluate the approach to teaching programming in the classroom. Various strategies can be considered to improve the student learning experience, such as revising the texts of exercise statements and materials to minimize their negative impact. These measures aim to provide a more inclusive and accessible learning environment, ensuring all students can engage and learn to code effectively.

#### B. Perceptions of Programming Teachers regarding neurodiverse students

Teachers' perception regarding neurodiverse students indicates some issues that need to be explored, as can be seen below:

- **Lack of Adequate Training** - Teachers recognize that neurodiverse students need adequate assistance in programming classes and everyone is interested in adequately serving students with disabilities in their classes. There is a problem but there is no adequate training for these teachers. Looking for ways to help students with their specific needs takes extra time and work, making it difficult to find solutions without proper guidance.
- **Insufficient communication** - Both educational institutions have support centers for people with disabilities. These centers aim to welcome and seek adequate support for all students with specific needs within the teaching environment. In practice, this support is more effective for those with sensory or physical disabilities (hiring Brazilian Sign Language interpreters to support deaf students, for example). In the case of neurodivergent students, this support is limited to signaling that the student has a specific need, that they need support, but it is not clear what support they are owed.

Regarding the lack of communication on the part of the educational institution, it is observed that specialized sectors generally inform teachers when there are students with disabilities and, as far as possible, provide guidance that may be useful. However, support is still insufficient for students with disabilities that are not "visible", leaving teachers concerned about their students' progress and without adequate guidance on how to act. There is a clear need to improve support and provide more comprehensive guidance so that teachers can more effectively meet the needs of these students.

Faced with the demand for insufficient training, each teacher sought alternative solutions based on their perceptions. This generated our third research question, which addresses how teachers sought to address the inclusion of neurodiverse students in their programming classes.

#### C. Perceptions of Programming Teachers Acting towards neurodiverse students

With inadequate training, teachers reported looking for strategies to help neurodiverse students.

- **Increase in Time** - There was a consensus among the teachers that these students needed more time to carry out the activities than the other students in the class. One of the teachers even used the strategy of having the student study content that would only last one semester in two. The extra time can be interesting, especially for students who have levels of inattention and take longer to concentrate.

- **Explanation** A strategy adopted by two teachers was to explain the subject more than once, seeking to vary the ways of speaking to minimize the students' difficulties. This way, students with different levels of difficulty can benefit.

- **Adaptation of Materials and Activities** Only one teacher reported that he tries to adapt his classes. His strategy is to increase the number of images. Using images is insufficient for adequate adaptation, but it is a start.

As we can see, teachers' actions toward neurodiverse students still lack better strategies to effectively help these students in programming classes. Teachers lack adequate support to show them what can be done. We know that adaptations require time and knowledge, but we must understand that they are essential for effectively including these students in programming classes.

#### VII. THREATS TO VALIDITY

Any research presents threats to validity. In this case, we identified the number of participants as a limitation, which we justify because the number of eligible subjects is small in the population. Therefore, we cannot guarantee that the findings would generalize to other participants with different types of cognitive impairment at different levels.

Another limitation would be human factors. These can affect the interview screening and content analysis process, as the same researcher defined, carried out, and analyzed the research.

#### VIII. CONCLUSION AND FUTURE WORKS

This research aimed to portray the perceptions of neurodiverse students and their teachers when teaching programming. The main difficulties reported by teachers were the need for more adequate training and communication. From the students' point of view, our study highlighted the main difficulties in understanding the application of programming, interpreting problems, creating algorithms without assistance, and remembering what to do in exercises and assessments.

We saw that there is still a lot to be discussed and explored and that teachers are interested in how to proceed but need guidance. With adequate support, we believe that teachers can positively reach this group of students, who will benefit their learning through conditions suited to their needs. Educational institutions must invest in specialized support for teachers, which can make a big difference in welcoming students with specific needs. Furthermore, it can be decisive in whether these students remain in the course.

Given the difficulties reported by students and teachers, we saw an apparent demand to consider new approaches to teaching programming to neurodiverse people. It is essential to develop inclusive pedagogical strategies and resources adapted to the characteristics and abilities of these students, providing them with equal opportunities. This may involve creating accessible learning environments, using clear language and instructions, and providing individualized support and differentiated learning structures. Overcoming this problem requires a collective commitment to ensure that all students have equal access to programming learning opportunities, regardless of their cognitive abilities.

Inclusion initiatives benefit students who have permanent limitations, but they can also help any student who has a temporary limitation or even difficulties arising from poor educational training, as programming is not something that is traditionally learned throughout school life.

Considering this context as a suggestion for future work, we propose the development of educational artifacts that can assist teachers in planning and conducting their classes in the context of neurodiversity. These educational resources can support educators by providing specific strategies, materials, and approaches to promote an inclusive and effective learning environment for these students. This initiative aims to strengthen teachers' ability to meet the individual needs of each student, promoting quality education for all.

#### REFERENCES

- [1] American Psychiatric Association: Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition. Arlington, VA: American Psychiatric Association, 2013
- [2] E. C. Juvino de Araújo, W. L. Andrade and A. L. Souto Oliveira, "Identifying Programming Skills Impacted in Students with Cognitive Disabilities," 2022 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2022, pp. 1-8, doi: 10.1109/FIE56618.2022.9962703.
- [3] Brazilian Law on the Inclusion of Persons with Disabilities (Statute of Persons with Disabilities) Law 13.146/2015. URL: [http://www.planalto.gov.br/ccivil\\_03/\\_ato2015-2018/2015/lei/l13146.htm](http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/l13146.htm)
- [4] F. F. da Silva, L. Beatrys Ruiz Aylon, and D. E. Flór, "Teaching Computational Thinking to a Student with Attention Deficit Through Programming," in 2020 IEEE Frontiers in Education Conference (FIE), 2020, pp. 1-9, doi: 10.1109/FIE44824.2020.9273915.
- [5] M. Israel, G. Jeong, M. Ray, and T. Lash, "Teaching Elementary Computer Science through Universal Design for Learning", in Proc. of SIGCSE 20, 2020, pp. 1220-1226. doi: 10.1145/3328778.3366823.
- [6] Maya Israel, Moon Y. Chung, Quentin M. Wherfel & Sadeddine Shehab (2020): A descriptive analysis of academic engagement and collaboration of students with autism during elementary computer science, Computer Science Education, DOI: 10.1080/08993408.2020.1779521
- [7] A. C. Gil, Como Elaborar Projetos de Pesquisa. Atlas, 2022.
- [8] J. Singer, Neurodiversity: The birth of an idea. 2017.
- [9] "Making content usable for people with cognitive and learning disabilities", Www.w3.org. [Online]. Available: <https://www.w3.org/TR/cog-usable/>.
- [10] R. K. Yin, Case study research: Design and methods, 5o ed. Thousand Oaks, CA, USA: SAGE Publications, 2014.
- [11] Bardin L. Análise de conteúdo. Lisboa: Edições 70; 2015.
- [12] Cormen, T.H. and Leiserson, C.E. and Rivest, R.L. and Stein, C. Introduction to Algorithms, fourth edition. MIT Press, 2022.
- [13] Johnny Saldaña. 2021. The coding manual for qualitative researchers. (2021), 1-440.
- [14] Lindgren, B. M., Lundman, B., & Graneheim, U. H. (2020). Abstraction and interpretation during the qualitative content analysis process. *International journal of nursing studies*, 108, 103632. <https://doi.org/10.1016/j.ijnurstu.2020.103632>
- [15] "Disability - PAHO/WHO — Pan American Health Organization". [Online] Available: <https://www.paho.org/en/topics/disability>.
- [16] Leitão, Carla. "A entrevista como instrumento de pesquisa científica em Informática na Educação: planejamento, execução e análise. In: PIMENTEL, Mariano; SANTOS, Edméa." (Org.) Metodologia de pesquisa científica em Informática na Educação: abordagem qualitativa. Porto Alegre: SBC, 2021. (Série Metodologia de Pesquisa em Informática na Educação, v. 3) [Online] Available: <https://metodologia.ceie-br.org/livro-3/>
- [17] Carl Haynes-Magyar. Neurodiverse Programmers and the Accessibility of Parsons Problems: An Exploratory Multiple-Case Study. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 491-497. <https://doi.org/10.1145/3626252.3630898>
- [18] E. C. J. De Araújo and W. L. Andrade, "A Systematic Literature Review on Teaching Programming to People with Cognitive Disabilities," 2021 IEEE Frontiers in Education Conference (FIE), Lincoln, NE, USA, 2021, pp. 1-8, doi: 10.1109/FIE49875.2021.9637361.
- [19] Castro Filho, Jose Aires; Freire, Raquel Santiago; Maia, Denny Leite. Estudo de Caso como método de pesquisa em Informática na Educação. In: PIMENTEL, Mariano; SANTOS, Edméa. (Org.) Metodologia de pesquisa científica em Informática na Educação: abordagem qualitativa. Porto Alegre: SBC, 2021. (Série Metodologia de Pesquisa em Informática na Educação, v. 3) [Online] Available: <https://metodologia.ceie-br.org/livro-3/>
- [20] Gomes, Alex Sandro; Gomes, Claudia Roberta Araújo. Classificação dos Tipos de Pesquisa em Informática na Educação. In: JAQUES, Patrícia Augustin; PIMENTEL, Mariano; SIQUEIRA, Sean; BITTENCOURT, Ig. (Org.) Metodologia de Pesquisa Científica em Informática na Educação: Concepção de Pesquisa. Porto Alegre: SBC, 2020. (Série Metodologia de Pesquisa em Informática na Educação, v. 1) [Online] Available: <https://metodologia.ceie-br.org/livro-1/>
- [21] Francisco, Deise Juliana; Azevedo, Edjane Mikaelly Silva; Ferreira, Adilson Rocha Ferreira; Caitano, Alexandre Rodrigues. Análise de Conteúdo: como podemos analisar dados no campo da educação e tecnologias. In: PIMENTEL, Mariano; SANTOS, Edméa. (Org.) Metodologia de pesquisa científica em Informática na Educação: abordagem qualitativa. Porto Alegre: SBC, 2021. (Série Metodologia de Pesquisa em Informática na Educação, v. 3) [Online] Available: <https://metodologia.ceie-br.org/livro-3/>
- [22] C. Galeos, K. Karpouzis and G. Tsatiris, "Developing an educational programming game for children with ADHD," 2020 15th International Workshop on Semantic and Social Media Adaptation and Personalization (SMA, Zakynthos, Greece, 2020, pp. 1-6, doi: 10.1109/SMAP49528.2020.9248458.
- [23] Varsha Koushik and Shaun K. Kane. 2019. "It Broadens My Mind": Empowering People with Cognitive Disabilities through Computing Education. In Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems (CHI '19). Association for Computing Machinery, New York, NY, USA, Paper 514, 1-12. <https://doi.org/10.1145/3290605.3300744>
- [24] Jie Ji, Christian Murphy, Brianna Blaser, and Jennifer Akullian. 2024. Experiences of Undergraduate Computer Science Students Living with Mental Health Conditions. In Proceedings of the 55th ACM Technical Symposium on Computer Science Education V. 1 (SIGCSE 2024). Association for Computing Machinery, New York, NY, USA, 597-603. <https://doi.org/10.1145/3626252.3630846>.
- [25] Waite, J. & Sentance, S. 2021 Teaching programming in schools: A review of approaches and strategies. Raspberry Pi Foundation Research Report. ISSN 2514-586X (19)
- [26] Thompson, S. Where do I begin? A problem solving approach to teaching functional programming. In K. Apt, P. Hartel, & P. Klint (Eds.), First International Conference on Declarative Programming Languages in Education. Springer-Verlag, 1997.
- [27] Qisheng Li, Josephine Lee, Christina Zhang, and Katharina Reinecke. 2021. How Online Tests Contribute to the Support System for People With Cognitive and Mental Disabilities. In Proceedings of the 23rd International ACM SIGACCESS Conference on Computers and Accessibility (ASSETS '21). Association for Computing Machinery, New York, NY, USA, Article 10, 1-15. <https://doi.org/10.1145/3441852.3471229>

- 
- [28] K. Eiselt and P. Carter, "Integrating Social Skills Practice with Computer Programming for Students on the Autism Spectrum," 2018 IEEE Frontiers in Education Conference (FIE), San Jose, CA, USA, 2018, pp. 1-5, doi:10.1109/FIE.2018.8659252.
- [29] Gelbar NW, Smith I, Reichow B. Systematic review of articles describing experience and supports of individuals with autism enrolled in college and university programs. *J Autism Dev Disord*. 2014 Oct;44(10):2593-601. doi: 10.1007/s10803-014-2135-5. PMID: 24816943.
- [30] Molitor SJ, Langberg JM, Evans SW. The written expression abilities of adolescents with Attention-Deficit/Hyperactivity Disorder. *Res Dev Disabil*. 2016 Apr-May;51-52:49-59. doi: 10.1016/j.ridd.2016.01.005. Epub 2016 Jan 21. PMID: 26802631; PMCID: PMC5134244.
- [31] S. Wille, J. Century and M. Pike, "Computer Science Principles (CSP) and students with learning differences: Expanding opportunities for a hidden underrepresented group," 2016 Research on Equity and Sustained Participation in Engineering, Computing, and Technology (RESPECT), Atlanta, GA, USA, 2016, pp. 1-8, doi: 10.1109/RESPECT.2016.7836160.

## **Apêndice H**

# **Guia de Elaboração de Material de Programação Acessível para Pessoas Neurodivergentes**

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**



### **Guia de Elaboração de Material de Programação Acessível para Pessoas Neurodivergentes**

O propósito central na elaboração deste guia é a criação de um artefato educacional inovador, cuidadosamente concebido para atender às necessidades particulares das pessoas neurodiversas. O guia busca ser uma ferramenta versátil destinada a professores, a fim de que estes possam adaptar seus próprios materiais de ensino de maneira a oferecer um suporte eficaz e sob medida para os alunos com neurodiversidade, garantindo assim uma experiência educacional mais inclusiva e enriquecedora. E que materiais de ensino poderiam ser adaptados com este guia? Roteiros de Aulas de Programação que o professor já tenha, bem como ele pode usar o template e criar um material próprio a partir dele.

*Este guia está sendo elaborado de forma a orientar a criação ou adaptação de roteiros para aulas de programação, que sejam acessíveis do ponto de vista de dificuldades cognitivas. Foram inseridos os itens principais de atenção para que o professor possa observar e padronizar seus materiais buscando favorecer o entendimento de alunos que possuam algum tipo de dificuldade dessa natureza, seja ela temporária ou permanente. Cada parte do roteiro foi pensada para adequar o material às necessidades específicas de pessoas neurodiversas. O guia apresenta o template a seguir, contendo as seções que precisam ser preenchidas, bem como a justificativa e instruções sobre o que colocar em cada seção.*

O professor deve dividir o conteúdo em unidades, e, para cada unidade deve criar o plano de aula e o roteiro de aulas.

Por exemplo:

- Contextualização
- Entrada e Saída
- Variáveis
- Estruturas de Decisão
- Estruturas de Repetição
- Etc...

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

#### Template para o plano de aula

Unidade de Conteúdo
Para melhor entendimento é sugerido que se divida o conteúdo em unidades menores.
Habilidades exploradas nesta unidade
<p>Várias habilidades estão envolvidas para que o aluno aprenda programação. Diante disso, é importante que tenhamos de forma clara quais habilidades o aluno precisa para compreender esta unidade de conteúdo.</p> <p>Na literatura, podemos identificar diversas habilidades envolvidas na aprendizagem de programação. Em [Araujo2022] temos algumas das principais, sendo elas:</p> <ol style="list-style-type: none"><li>1. Resolução de Problemas,</li><li>2. Habilidades Matemáticas,</li><li>3. Abstração,</li><li>4. Pensamento Crítico,</li><li>5. Compreensão de Texto,</li><li>6. Conhecimento em Inglês,</li><li>7. Atenção aos Detalhes,</li><li>8. Criatividade e</li><li>9. Gerenciamento de Tempo.</li></ol> <p>Quando pensamos no sujeito neurodivergente, temos que atentar para o fato de que algumas dessas habilidades podem ser comprometidas. Isto não significa que o aluno não consiga adquirir esta habilidade, mas sim que o professor precisa pensar em estratégias que reforcem o exercício da habilidade no contexto da aula de programação.</p> <p><i>Podemos observar que, alunos autistas podem ser muito bons em resolução de problemas, porém tem dificuldade em abstração. Já alunos com TDAH podem ser especialmente bons em criatividade, mas ter dificuldade em atenção aos detalhes e gerenciamento de tempo.</i></p> <p>Por exemplo, a habilidade de resolução de problemas é constantemente exigida nas aulas de programação, então o desafio é prender a atenção do neurodivergente. Podemos inserir nas aulas resolução de problemas mais simples, e gradativamente aumentando a complexidade deles. Ou seja, que os primeiros problemas mostrados nas aulas poderiam ser com apenas uma entrada e uma saída, caso esteja trabalhando algo que precise de fórmulas matemáticas que sejam fórmulas que tenham menos argumentos, etc.</p> <p><i>(Para ver um exemplo na prática acesse o roteiro exemplo de condicional)</i></p> <p>Sobre habilidades matemáticas, podemos sempre que for utilizar um exemplo que utilize um cálculo matemático explicar do que se trata antes.</p>

***Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.***

### **Orientações para dificuldades cognitivas e neurodiversidade**

Essa seção foi transformada em orientações fora dos templates.

### **Roteiro Teórico de Estudo**

Alunos neurodiversos precisam ter roteiros de estudo para saber de forma clara o que e como irá aprender algo.

Esse roteiro deverá ser criado com base em orientações específicas que podem facilitar o processo de aprendizagem e serve como guia para consulta sempre que o aluno precisar.

As orientações constam no final deste guia.

### **Exercícios**

De acordo com cada unidade, é interessante que seja atrelada uma lista de atividades que contemplem a unidade de conteúdo e também levem em consideração as orientações para minimizar as dificuldades cognitivas.

As orientações constam no final deste guia.

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

#### Template Roteiro de Estudos por Unidade de Conteúdo

<b>Qual é nossa unidade de conteúdo?</b>
Devemos iniciar o roteiro mostrando de forma clara qual é a unidade de conteúdo que estamos abordando aqui. Para o contexto de alunos neurodivergentes é de suma importância fornecer estratégias que possam auxiliá-los a estudar mais efetivamente. Alunos neurodivergentes podem ter dificuldades em algumas funções executivas, como por exemplo a memória de trabalho, que é aquela que utilizamos para entender coisas e solucionar problemas. Nesse caso o aluno pode se desorganizar ao tentar estudar e a ideia desses roteiros é justamente ajudá-lo a manter o foco e organizar os conceitos a serem estudados.
<b>O que queremos aprender nesta unidade</b>
Para alunos neurodiversos, é importante trabalhar com a previsibilidade, de forma que ele saiba desde o início da leitura o que ele irá aprender com aquele conteúdo. Isto é uma estratégia para buscar minimizar o fato de não saber o que virá em seguida, o que pode deixar o aluno ansioso. Um resumo sucinto do que será aprendido nesta unidade de conteúdo, de preferência na forma de tópicos, deve constar no início do roteiro.
<b>Contextualizando</b>
Fazer ligações do conteúdo com o mundo real e concreto.
<b>O que vamos precisar</b>
É preciso deixar claro quais materiais ou ferramentas serão necessárias para esta aula. É importante que o professor não apenas disponibilize os materiais e ferramentas, mas também mostre como utilizá-las. No caso de alunos neurodivergentes, não precisamos fazer uso de tecnologias assistivas como leitores de tela por exemplo, mas precisamos que o aluno aprenda a utilizar ferramentas como IDE's ou até mesmo como executar os códigos presentes no roteiro do Google Collab.
<b>Roteiro para Estudo</b>
Neste item será apresentado de fato o conteúdo de forma teórica. É importante dividir as ideias em frases mais curtas, usar imagens ilustrativas quando possível e destacar conceitos importantes no texto, utilizando cores diferentes e negrito, por exemplo. Orientações sobre o texto estão no final deste guia.
<b>Resumo do que foi apresentado na unidade</b>

***Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.***

De forma a facilitar o entendimento e a consulta posterior, é importante ao final do roteiro apresentar um resumo em forma de tópicos, com conceitos chaves do que foi aprendido na unidade.

Alunos neurodivergentes podem ter dificuldades em planejamento e organização, o que pode dificultar a aprendizagem mesmo em alunos que tenham inteligência acima da média. Desta forma é importante que ao final do roteiro se estruture os principais conceitos em forma de tópicos para facilitar a organização mental do conhecimento para o estudante.

#### **Para saber mais**

Como adicional interessante, podemos acrescentar ao roteiro algumas formas e fontes de estudo adicional caso ele queira fazer pesquisa, aqui podemos inserir links para materiais extras, nomes de livros, etc...

#### **Exercícios e Exemplos de Código**

De acordo com cada unidade, é interessante que seja atrelada uma lista de atividades que contemplem a unidade de conteúdo e também levem em consideração as orientações para minimizar as dificuldades cognitivas. As orientações constam no final deste guia.

### **Orientações para adaptação de conteúdo nos roteiros para estudo**

Alunos com deficiência cognitiva ou neurodiversos podem ter dificuldades em algumas habilidades, como por exemplo abstração e resolução de problemas. Por outro lado, devemos também pensar nas potencialidades que a neurodiversidade pode proporcionar. Por exemplo, estudos revelam que alunos com dislexia demonstram habilidades ágeis na resolução de problemas, destacando a programação como um campo que pode potencializar seus pontos fortes e mitigar suas fraquezas [Wille2016, Powel2004]. Indivíduos com autismo e TDAH podem desenvolver hiperfoco, uma característica na qual concentram intensa atenção em assuntos de interesse. Essa peculiaridade pode facilitar o aprendizado da programação se o aluno manifestar interesse. O professor pode estimular essa inclinação por meio de abordagens envolventes, como a integração de jogos, caso o aluno aprecie essa forma de interação [Silva2020].

Para adaptarmos um roteiro de aula, utilizamos princípios do Design Universal para Aprendizagem [DUA2020] que é um conjunto de diretrizes criadas nos Estados Unidos e visa proporcionar flexibilidade na aprendizagem, partindo do princípio que cada aluno pode aprender de formas diferentes. Além deste também temos as Recomendações do W3C para acessibilidade Cognitiva [COGA-W3C2021], que foram geradas focando-se em recomendações para a Internet, o que pode enriquecer o material gerado, já que estamos tratando de aulas de Programação, onde se faz uso constante de tecnologia. Por fim

***Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.***

também buscamos informações relevantes no Manual Diagnóstico de Transtornos Mentais[DSM-5], documento norteador sobre os tipos de condições neurológicas que podem afetar as pessoas bem como suas características.

Focando em alunos neurodivergentes e pensando nas especificidades do ensino de Programação, temos recomendações que mais se encaixam, são algumas delas:

- Oferecer ao estudante **múltiplas formas de apresentação da informação** - Pode-se por exemplo oferecer ao aluno o roteiro de estudos na forma de **áudio e vídeo**; criando o roteiro usando o Google Collab temos a flexibilidade de inserir trechos de código que podem ser testados pelo aluno em tempo real ; Outra forma de apresentação pode ser a **utilização de imagens ou diagramas**;
- Oferecer ao estudante **esclarecimentos sobre itens específicos de vocabulário e símbolos ou fórmulas matemáticas** - Em Programação temos o uso de muitos termos em Inglês, que não é nossa língua nativa, podendo gerar problemas no entendimento então é importante deixar claro o significado dos termos sempre que necessário. Outra coisa bastante utilizada em Programação são exemplos utilizando fórmulas matemáticas, então também é importante explicar ou deixar uma explicação de fácil acesso para o estudante consultar caso necessário.
- Oferecer ao estudante **pontos de conexão com coisas aprendidas anteriormente** - Alunos neurodivergentes podem possuir dificuldades de memorização e como o conhecimento de Programação é cumulativo, é importante que se lembre sempre que possível o que já foi visto antes e que será necessário para a unidade de conteúdo atual;
- **Facilitar o entendimento de textos**: textos longos e cheios de expressões complicadas podem desestimular alunos neurodivergentes, é importante que essa parte do roteiro de estudo que é mais teórica seja elaborada fazendo uso de quebras no texto, como parágrafos mais curtos, uso de figuras, tabelas, destaques como negrito ou outra estratégia que possa deixar a leitura mais confortável para esse público alvo;
- Complementar o texto com **outras formas de apresentação**: Ao longo da parte teórica, é importante que se acrescentem exemplos de código para que o aluno tenha noção de como aquele conceito seria aplicado na prática no Algoritmo. O uso do Google Collab é interessante pois permite que o aluno teste o código em tempo real naquele ponto específico do roteiro, dessa forma temos como prender a atenção daquele aluno que tem o perfil mais disperso;
- Buscar ao longo do texto, fazer **ligações com o mundo real e com exemplos mais concretos** de uso daquele conteúdo: alunos neurodivergentes podem ter dificuldade com abstração então fazer essa ponte com o concreto pode ajudar o aluno a compreender melhor como poderia aplicar os conceitos de Programação fora da sala de aula;
- Alunos neurodivergentes podem apresentar dificuldades na aprendizagem, como por exemplo a dificuldade em entender o que é lido. Muitas vezes em Programação, os professores fazem uso de um grande texto para que o aluno extraia dali informações para criar um algoritmo que resolva aquele problema. A forma como esses textos são elaborados pode se tornar um empecilho para que o aluno neurodivergente consiga entender o que se busca. Deve-se sempre prezar pelo **texto simples e**

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

**literal** na medida do possível, também é uma característica desse público a dificuldade em entender as coisas em sentido figurado.

Vejamos um exemplo, utilizando um link público do livro *Pense em Python*, de Allen B. Downey, traduzido e publicado no Brasil pela Editora Novatec sob licença da O'Reilly Media. Você pode ter acesso ao conteúdo na íntegra clicando na imagem abaixo.

## Capítulo 5: Condicionais e recursividade

O tópico principal deste capítulo é a instrução `if`, que executa códigos diferentes dependendo do estado do programa. Mas primeiro quero apresentar dois novos operadores: divisão pelo piso e módulo.

### 5.1 - Divisão pelo piso e módulo

O operador de divisão pelo piso, `//`, divide dois números e arredonda o resultado para um número inteiro para baixo. Por exemplo, suponha que o tempo de execução de um filme seja de 105 minutos. Você pode querer saber a quanto isso corresponde em horas. A divisão convencional devolve um número de ponto flutuante:

```
>>> minutos = 105
>>> minutos / 60
1.75
```

Mas não é comum escrever horas com pontos decimais. A divisão pelo piso devolve o número inteiro de horas, ignorando a parte fracionária:

```
>>> minutos = 105
>>> hours = minutos // 60
>>> hours
1
```

Para obter o resto, você pode subtrair uma hora em minutos:

```
>>> remainder = minutos - hours * 60
>>> remainder
45
```

**Figura 1 : Trecho do Livro Pense em Python**

Observe que o conteúdo exposto nesse capítulo trata de Condicionais e Recursividade, nesse caso agregar muitos conceitos em um só capítulo pode ser difícil para os alunos assimilarem.

Vamos criar o plano de aula e o roteiro de acordo com os templates e recomendações.

Para criar um roteiro de aula sobre estruturas condicionais em Python que seja acessível a pessoas neurodiversas, o professor utilizaria o template apresentado neste guia, encaixando o conteúdo que ele já possui, alterando quando necessário.

Poderíamos então separar em duas unidades: Condicionais e Recursividade. Se o professor julgar necessário pode quebrar em unidades ainda menores.

### Plano de Aula Estrutura Condicional

<b>Unidade de Conteúdo</b>
Estrutura Condicional
<b>Habilidades exploradas nesta unidade</b>

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

<p>Algumas habilidades que podem ser trabalhadas com Estrutura Condicional</p> <ul style="list-style-type: none"> <li>• <b>Resolução de Problemas:</b> Através da criação de Algoritmos;</li> <li>• <b>Habilidades Matemáticas:</b> Exercícios que abordem conceitos matemáticos e usem condicional, como por exemplo cálculo de média aritmética e ponderada;</li> <li>• <b>Abstração:</b> Trabalhar diferentes exemplos de uso de condicional na prática;</li> <li>• <b>Compreensão de Texto:</b> Incentivar o aluno a ler com calma o problema e tirar dúvidas antes de iniciar a resolução;</li> <li>• <b>Conhecimento em Inglês:</b> Ao se familiarizar com a sintaxe da linguagem Python, que usa o Inglês como base para suas palavras reservadas;</li> <li>• <b>Atenção aos Detalhes:</b> Ajudar o aluno a aprender a depurar seu código, ajudando sempre que necessário a identificar erros;</li> </ul>
<p><b>Roteiro para Estudo (Teoria da Unidade de Conteúdo)</b></p>
<p>Alunos neurodiversos precisam ter roteiros de estudo para saber de forma clara o que e como irá aprender algo. Esse roteiro deverá ser criado com base em orientações específicas que podem facilitar o processo de aprendizagem e serve como guia para consulta sempre que o aluno precisar. As orientações constam no final deste guia.</p>
<p><b>Exercícios</b></p>
<p>Exemplos a seguir.</p>

#### Roteiro Estrutura Condicional

**Obs: Este roteiro está dentro do documento como exemplo, mas poderia ser criado diretamente no Google Collab, como mostrado na primeira rodada.**

<p><b>Qual é nossa unidade de conteúdo?</b></p>
<p>A unidade de conteúdo é Estrutura Condicional.</p>
<p><b>O que queremos aprender nesta unidade</b></p>
<p>O tópico principal desta unidade é a estrutura if, que executa códigos diferentes dependendo do estado do programa.</p>
<p><b>Contextualizando</b></p>

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

No mundo ao nosso redor, estamos sempre tomando decisões. Por exemplo, para sair de casa e chegar na universidade, o aluno pode ir por vários caminhos. E qual caminho seguir?

Se ele quiser chegar mais rápido podemos imaginar que ele vai utilizar o caminho mais curto.

E como ele pode saber qual seria esse caminho? Comparando a distância em quilômetros, usando por exemplo o Google Maps.

No mundo da programação também temos essa possibilidade.

### O que vamos precisar

Iremos acessar o <https://pythontutor.com/python-compiler.html#mode=edit> nesse momento para criarmos e testarmos código Python.

Não sabe ainda utilizar o Python Tutor? Avise ao professor que ele irá te ajudar.

### Roteiro para Estudo (Teoria da Unidade de Conteúdo)

#### Divisão inteira:

O operador de divisão inteira é //

Ele divide dois números e arredonda o resultado para um número inteiro para baixo.

Por exemplo, temos um filme que dura 105 minutos. Você pode querer saber a quanto isso corresponde em horas. Se dividirmos normalmente, o resultado terá casas decimais e não é comum falarmos de tempo com casas decimais, então se fizermos o cálculo assim:

```
minutos = 105
minutos / 60
Saída 1.75
```

Ficaria estranho falar que o filme durou 1.75 horas.

Por isso podemos usar a divisão inteira assim:

```
minutos = 105
horas = minutos // 60
horas
Saída 1
```

E temos quantas horas inteiras o filme possui: 1 hora inteira.

Mas e o resto do tempo do filme?

Podemos utilizar o cálculo do resto da divisão para isso. O operador de resto da divisão inteira é o %

```
restante = minutos % 60
restante
Saída 45
```

Então podemos dizer que o filme dura 1 hora e 45 minutos.

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

**(Este é só um exemplo de adaptação do trecho que está na Figura 1 então você continuaria o seu conteúdo aqui...)**

#### Resumo do que foi apresentado na unidade

Nessa unidade aprendemos a utilizar estruturas condicionais em Python. Vimos também como calcular a divisão inteira e o resto da divisão inteira.

#### Para saber mais

Você pode acessar o livro *Pense em Python* que está público nesse link: <https://penseallen.github.io/PensePython2e/05-cond-recur.html>

#### Para praticar

*Ver exercício adaptado abaixo*  
*Links para repositórios de exemplos de código*

### Orientações para adaptação de exercícios e avaliações de programação

**Exercícios** são estratégias importantes para praticar os conceitos aprendidos. No ensino de programação, o uso de listas de exercícios é amplamente utilizado já que, quanto mais algoritmos o aluno constrói, mais experiência ele ganha para resolver outros.

Pensando nas especificidades de programação, podemos adaptar os exercícios levando-se em consideração as seguintes recomendações:

- **A lista de exercícios deve ter um cabeçalho**, contendo informações importantes sobre a execução das atividades, como qual será a ferramenta a ser utilizada, a linguagem de programação, além de um primeiro **exercício resolvido** para que o aluno possa consultar quando precisar e servir de base para os outros. Alunos **autistas** por exemplo, tendem a se sentir mais confortáveis com a previsibilidade e a rotina, então um cabeçalho pode ajudar nesse sentido. Já para o aluno com **TDAH**, o cabeçalho pode ser um recurso para não deixá-lo ansioso sem saber o que está por vir.
- Enunciados o mais objetivos possíveis, buscando um **texto mais simples e com instruções claras**. O uso de negrito em palavras chave pode ser uma boa estratégia para chamar atenção dentro do texto. Para os **autistas**, pode ser mais difícil entender analogias que não estejam literalmente explicadas, então deve-se ter cuidado com isso. Para alunos com **TDAH**, o negrito pode ser uma boa estratégia para chamar atenção para os pontos chave.
- Cada questão deve ter o **enunciado, exemplos de entrada de dados e exemplos de saída de dados correspondentes**. Isso ajuda o aluno neurodivergente a saber por onde começar.

---

***Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.***

- **Exercícios em grupo** podem ser interessantes, desde que o grupo possua no máximo três alunos. Um grupo muito grande pode deixar o neurodivergente ansioso e disperso.
- Nas disciplinas de programação é comum fazermos atividades comumente chamadas **projetos**, onde existe um problema grande a ser resolvido criando-se um programa. Nesse caso é importante que a entrega seja **dividida em etapas**, dessa forma o aluno irá ser condicionado a se focar em apenas uma parte por vez, diminuindo a chance dele buscar fazer várias atividades ao mesmo tempo, característica do neurodivergente que possui hiperatividade.
- Estabelecer **limites de tempo mínimo e máximo** previamente pode ser interessante para que o aluno não desista de resolver as questões na primeira dificuldade.

No caso de **atividades avaliativas**, as recomendações para exercícios também se aplicam, mas além delas temos algumas específicas:

- É importante que todos os **critérios de avaliação e instruções de realização** fiquem claros e sejam colocados no **cabeçalho da folha**, para que o aluno possa ler quantas vezes forem necessárias, já que durante a avaliação ele pode esquecer deles.
- **Mencionar claramente a pontuação** de cada questão ajuda o aluno a não ficar ansioso e se organizar melhor.
- Da mesma forma que dividimos a disciplina em unidades de conteúdo, também é interessante **dividir as avaliações** por essas unidades, sendo o aluno avaliado ao final de cada unidade, o que vai ajudá-lo a manter o foco e não se perder no que foi aprendido.
- O professor (ou monitor, ou um profissional leitor) pode **ler as questões para o aluno**, trazendo explicações adicionais de forma oral que ajudem no entendimento.

**Exemplo de aplicação das recomendações em um exercício do mesmo material que utilizamos no roteiro anterior**

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

### Exercício 5.3

Se você tiver três gravetos, pode ser que consiga arranjá-los em um triângulo ou não. Por exemplo, se um dos gravetos tiver 12 polegadas de comprimento e outros dois tiverem uma polegada de comprimento, não será possível fazer com que os gravetos curtos se encontrem no meio. Há um teste simples para ver se é possível formar um triângulo para quaisquer três comprimentos:

Se algum dos três comprimentos for maior que a soma dos outros dois, então você não pode formar um triângulo. Senão, você pode. (Se a soma de dois comprimentos igualar o terceiro, eles formam um triângulo chamado "degenerado".)

1. Escreva uma função chamada `is_triangle` que receba três números inteiros como argumentos, e que imprima "Yes" ou "No", dependendo da possibilidade de formar ou não um triângulo de gravetos com os comprimentos dados.
2. Escreva uma função que peça ao usuário para digitar três comprimentos de gravetos, os converta em números inteiros e use `is_triangle` para verificar se os gravetos com os comprimentos dados podem formar um triângulo.

Exercício Exemplo disponível em:

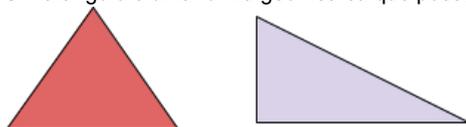
<https://penseallen.github.io/PensePython2e/05-cond-recur.html>

### Exemplo de Exercício

Lista de Exercícios Para Praticar	
<b>Unidade de Conteúdo</b>	Estruturas Condicionais
<b>Exemplo de código para relembra</b>	
<pre>def media (n1, n2, n3) :     return ( (n1+n2+n3) /3)  #programa principal nota1 = int(input("Informe a primeira nota")) nota2 = int(input("Informe a segunda nota")) nota3 = int(input("Informe a terceira nota")) resultado = media (nota1, nota2, nota3) print (resultado) if (resultado &lt; 4.0) :     print ("REPROVADO") elif (resultado &gt;= 4.0 and resultado &lt; 7.0) :     print ("FINAL") else :     print ("APROVADO")</pre>	
<b>Tempo aproximado para resolver essa questão</b>	15 minutos
<b>Enunciado</b>	

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

Um triângulo é uma forma geométrica que possui três lados.



Se você tiver três gravetos (pedaços de madeira), pode ser que consiga arranjá-los em um triângulo ou não.

As três partes que formam um triângulo não podem ter qualquer tamanho. Por exemplo, se um dos gravetos tiver 12 cm de comprimento e outros dois tiverem 1 cm de comprimento, não será possível fazer com que os gravetos curtos se encontrem no meio.

Há um teste simples para ver se é possível formar um triângulo dados três comprimentos:

- **Se algum dos três comprimentos for maior que a soma dos outros dois, então você não pode formar um triângulo.**
- **Senão, você pode.**

Agora crie um algoritmo que resolva o problema de saber se três valores podem formar um triângulo.

Você precisa fazer duas funções no seu programa:

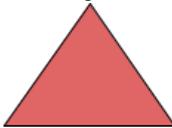
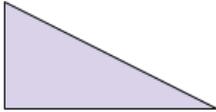
1. Escreva uma função chamada **is\_triangle** que receba **três números inteiros** como **entrada**, e que tenha a saída "Yes" se formar um triângulo ou "No", se não formar um triângulo com os valores passados de entrada.
2. Escreva **outra função** que **peça** ao usuário para digitar **três comprimentos de gravetos**, os converta em números inteiros e use a função criada is\_triangle para verificar se os gravetos com os comprimentos dados podem formar um triângulo.

- Temos dois exemplos abaixo:
  - No exemplo azul temos como entrada para o lado 1 o valor 1, para o lado 2 o valor 1 e para o lado 3 o valor 12. Nesse exemplo a saída seria No.
  - Já no exemplo laranja temos como entrada para o lado 1 o valor 5, para o lado 2 o valor 5 e para o lado 3 o valor 10. Nesse exemplo a saída seria o valor Yes.

Exemplo de entrada	Exemplo de entrada
1 1 12	5 5 10
Exemplo de saída	Exemplo de saída
No	Yes

**Exemplo de aplicação das recomendações em uma avaliação**

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

Avaliação da Aprendizagem	
<b>Unidade de Conteúdo</b>	Estruturas Condicionais
<b>Instruções para realização da avaliação</b>	
<ul style="list-style-type: none"> <li>• A avaliação é individual;</li> <li>• A avaliação será realizada no computador utilizando o editor Python;</li> <li>• Cada questão vale 50 pontos;</li> <li>• A avaliação toda vale 100 pontos;</li> <li>• Qualquer dúvida deve ser reportada ao professor.</li> </ul>	
<b>Exemplo de código para lembrar</b>	
<pre>def media (n1,n2,n3) :     return ((n1+n2+n3)/3)  #programa principal nota1 = int(input("Informe a primeira nota")) nota2 = int(input("Informe a segunda nota")) nota3 = int(input("Informe a terceira nota")) resultado = media(nota1,nota2,nota3) print(resultado) if(resultado&lt;4.0):     print("REPROVADO") elif(resultado&gt;=4.0 and resultado&lt;7.0):     print("FINAL") else:     print("APROVADO")</pre>	
<b>Pontuação total da avaliação</b>	100 pontos
<b>Pontuação desta questão</b>	50 pontos
<b>Tempo aproximado para resolver essa questão</b>	15 minutos
<b>Enunciado</b>	
<p>Um triângulo é uma forma geométrica que possui três lados.</p> <div style="display: flex; justify-content: space-around; align-items: center;">   </div> <p>Se você tiver três gravetos (pedaços de madeira), pode ser que consiga arranjá-los em um triângulo ou não.</p> <p>As três partes que formam um triângulo não podem ter qualquer tamanho. Por exemplo, se um dos gravetos tiver 12 cm de comprimento e outros dois tiverem 1 cm de comprimento, não será possível fazer com que os gravetos curtos se encontrem no meio.</p>	

**Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.**

<p>Há um teste simples para ver se é possível formar um triângulo dados três comprimentos:</p> <ul style="list-style-type: none"> <li>• <b>Se algum dos três comprimentos for maior que a soma dos outros dois, então você não pode formar um triângulo.</b></li> <li>• <b>Senão, você pode.</b></li> </ul> <p>Agora crie um programa em Python que resolva o problema de saber se três valores podem formar um triângulo.</p> <p>Você precisa fazer duas funções no seu programa:</p> <ol style="list-style-type: none"> <li>3. Escreva uma função chamada <b>is_triangle</b> que receba <b>três números inteiros</b> como <b>entrada</b>, e que tenha a saída “Yes” se formar um triângulo ou “No”, se não formar um triângulo com os valores passados de entrada.</li> <li>4. Escreva <b>outra função</b> que <b>peça</b> ao usuário para digitar <b>três comprimentos de gravetos</b>, os converta em números inteiros e use a função criada <b>is_triangle</b> para verificar se os gravetos com os comprimentos dados podem formar um triângulo.</li> </ol>	
Exemplo de entrada	Exemplo de entrada
1 1 12	5 5 10
Exemplo de saída	Exemplo de saída
No	Yes

#### Referências para elaboração desse guia

[DUA2020] SEBASTIÁN-HEREDERO, E.. Diretrizes para o Desenho Universal para a Aprendizagem (DUA). Revista Brasileira de Educação Especial, v. 26, n. 4, p. 733–768, out. 2020.

[LBI2015] Brasil (2015). Lei nº 13.146, de 06 de julho de 2015. Diário Oficial da República Federativa do Brasil.

[Singer2017] Singer, J. (2017). Neurodiversity: The Birth of an Idea. Judy Singer.

[DSM-5] DSM-5, A. P. A. (2013). Diagnostic and statistical manual of mental disorders, fifth edition. (Acesso em 10/05/2023).

[ARAUJO2022] Araujo, E. C. J. d., Andrade, W. L., and Oliveira, A. L. S. (2022). Identifying programming skills impacted in students with cognitive disabilities. Em 2022 IEEE Frontiers in Education Conference (FIE), páginas 1–8.

[COGA-W3C2021] <https://w3c.github.io/coga/user-research/>

[WILLE2016] S. Wille, J. Century, and M. Pike, “Computer Science Principles (CSP) and

***Este conteúdo faz parte da pesquisa de Doutorado de Elaine Cristina Juvino de Araújo, seu conteúdo não pode ser compartilhado nem copiado sem autorização.***

students with learning differences: Expanding opportunities for a hidden underrepresented group, em Proceedings of the RESPECT'16, 2016, pp. 1–8.

[POWEL2004] Norman Powell, David Moore, John Gray, Janet Finlay & John Reaney (2004) Dyslexia and learning computer programming, Innovation in Teaching and Learning in Information and Computer Sciences, 3:2, 1-12, DOI: 10.11120/ital.2004.03020005

[DOWNEY2016] Esta versão digital de Pense em Python em português foi gerada por Luciano Ramalho da ThoughtWorks a partir de arquivos cedidos por Rubens Prates da Editora Novatec. Acesso em: <https://penseallen.github.io/PensePython2e/>

[ARRUDA2014] Comunidade Aprender Criança. Cartilha da Inclusão Escolar: inclusão baseada em evidências científicas (Ed. Instituto Glia, 2014).