



**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO SERTÃO  
PERNAMBUCANO  
CAMPUS SALGUEIRO  
MESTRADO PROFISSIONAL EM EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**

**JOSÉ WALTER PAULINO JÚNIOR**

**PENSAMENTO COMPUTACIONAL COMO METODOLOGIA PARA ENSINO DE  
LÓGICA DE PROGRAMAÇÃO: UMA EXPERIÊNCIA NO ENSINO MÉDIO  
INTEGRADO**

**Salgueiro-PE**

**2024**

**JOSÉ WALTER PAULINO JÚNIOR**

**PENSAMENTO COMPUTACIONAL COMO METODOLOGIA PARA ENSINO DE  
LÓGICA DE PROGRAMAÇÃO: UMA EXPERIÊNCIA NO ENSINO MÉDIO  
INTEGRADO**

Dissertação apresentada ao Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo campus Salgueiro do Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Orientador: Dr. Francisco Kelsen de Oliveira

Salgueiro - PE

2024

P 324f Paulino Júnior, José Walter..
Ficha catalográfica para trabalhos acadêmicos / João de Paula. - Salgueiro,2024.
120p.
Dissertação (mestrado) – Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, Campus Salgueiro, Curso de Mestrado Profissional em Educação Profissional e Tecnológica (PROFEPT), Salgueiro, 2024..
Orientador: Francisco Kelsen De Oliveira
1. Ficha Catalográfica. 2. Método de Estudo. 3. Trabalhos Científicos. I. João de Paula. II. Título
CDD 001.4

Catalogado por: (Nome do Bibliotecário e Registro no CRB)

**PROGRAMA DE PÓS GRADUAÇÃO EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**

---

**José Walter Paulino Júnior**

**PENSAMENTO COMPUTACIONAL COMO METODOLOGIA PARA ENSINO DE  
LÓGICA DE PROGRAMAÇÃO: UMA EXPERIÊNCIA NO ENSINO MÉDIO  
INTEGRADO**

Dissertação apresentada ao Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Aprovado em 06 de maio de 2024.

**COMISSÃO EXAMINADORA**

---

Prof. Dr. Francisco Kelsen De Oliveira  
(ProfEPT/IFSertãoPE)

Orientador

---

Prof. Dr. Ricardo de Andrade Araújo  
(ProfEPT/IFSertãoPE)

---

Prof. Dr. Leandro Marques Queiros (UFPE)



# INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO SERTÃO PERNAMBUCANO

Autarquia criada pela Lei nº 11.892 de 29 de Dezembro de 2008



## PROGRAMA DE PÓS GRADUAÇÃO EM EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

---

**JOSÉ WALTER PAULINO JÚNIOR**

### UM GUIA DE BOAS PRÁTICAS: PENSAMENTO COMPUTACIONAL UTILIZANDO SCRATCH

Produto Educacional apresentado ao Programa de Pós-graduação em Educação Profissional e Tecnológica, ofertado pelo Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Validado em 06 de maio de 2024.

#### **COMISSÃO EXAMINADORA**

---

Prof. Dr. Francisco Kelsen De Oliveira  
(ProfEPT/IFSertãoPE)

Orientador

---

Prof. Dr. Ricardo de Andrade Araújo  
(ProfEPT/IFSertãoPE)

---

Prof. Dr. Leandro Marques Queiros (UFPE)

Dedico esta dissertação a minha mãe,  
Lúcia Sousa, pelo exemplo de coragem, dedicação  
e cuidado, que com muito carinho, me ensinou os  
bons e corretos caminhos da vida.

## **AGRADECIMENTOS**

Impossível não começar agradecendo a Deus, pela saúde, disposição, coragem e paz, para que eu pudesse desenvolver todas as etapas do mestrado. A Deus, ao mestre Jesus, toda glória e honra.

Aos meus familiares, em especial meus pais, Walter e Lúcia, pela criação e pela educação que me proporcionaram durante a minha formação. Ao meu orientador, Prof. Dr. Francisco Kelsen De Oliveira, por ter sido tão acessível, presente, solícito e paciente, sempre disposto a ajudar.

Aos meus colegas de mestrado e de caronas do Cariri Cearense ao Pernambuco, Eliane, Luiz Roberto, Wilson e Ricardo.

A todos os professores e servidores do IF Sertão, em especial ao amigo e colega de mestrado, Heraldo Lima, profissional de muita estima.

A amiga e professora Cícera Alves Agostinho de Sá (Jucy) pelas conversas e debates sobre pesquisa científica nos corredores da escola, por todo apoio e palavras de incentivo.

A todos os estudantes do Ensino Médio Integrado ao Técnico em Informática da EEEP Irmã Ana Zélia da Fonseca, participantes desta pesquisa.

Enfim, a todos que direta ou indiretamente contribuíram para a realização desta dissertação, gratidão!

## RESUMO

O presente trabalho tem como objetivo investigar como o Pensamento Computacional (PC) podem contribuir na disciplina de Lógica de Programação (LP) por meio da conexão entre teoria e prática, propondo uma metodologia alternativa que utiliza em sua base os fundamentos do PC (Abstração, Decomposição, Generalização e Algoritmos) para promover o Ensino de Lógica de Programação a partir de materiais didáticos complementares de fácil linguagem e aderente ao contexto da Educação Profissional e Tecnológica (EPT). Fazendo com que os estudantes possam perceber a necessidade e a importância de desenvolver habilidades de programação para acompanhar as transformações da sociedade. Dentro do ProfEPT, seguirá a linha de pesquisa Práticas Educativas em Educação Profissional e Tecnológica, no âmbito do macroprojeto 1 - Propostas metodológicas e recursos didáticos em espaços formais e não formais de ensino na EPT. A metodologia deu-se a partir da abordagem qualitativa sob uma perspectiva descritiva e observacional. Onde ocorreu uma intervenção com aplicação de minicurso e em seguida, atividades avaliativas a partir da resolução de problemas. As análises das opiniões, desempenho e experiências dos estudantes foram observadas no âmbito qualitativo, onde os sujeitos da pesquisa foram 25 estudantes e 3 docentes do curso do Ensino Médio Integrado ao Técnico em Informática da Escola Estadual de Educação Profissional Irmã Ana Zélia da Fonseca. Os instrumentos e procedimentos para a coleta de dados foram questionários, submetidos também a apreciação pelo Comitê de Ética em Pesquisa do IFSertãoPE. O modelo de formação, análise e interpretação dos questionários e seus dados resultantes deu-se a partir da análise de conteúdo proposta por Bardin (2010). Nas avaliações das interpretações dos dados percebeu-se uma evolução considerável dos estudantes, tomando como base o desenvolvimento de habilidades de programação a partir dos desafios propostos pela intervenção, onde 40% validaram a implementação da metodologia alternativa utilizando PC na disciplina de LP e 52% acreditaram também a possibilidade de implementar em outras disciplinas, porém, ao final do estudo através de análise e comparação dos resultados foi perceptível que utilizar somente a metodologia de PC não seria o suficiente, pois não há unanimidade entre os participantes da pesquisa. Conclui-se então que o mais indicado é utilizar a metodologia de PC como complemento a LP e não como uma metodologia que venha a substituir.

**Palavras-Chave:** Educação Profissional. Técnico em Informática. Aprendizado em computação.



## ABSTRACT

The present work aims to investigate how Computational Thinking (CP) can contribute to the discipline of Programming Logic (LP) through the connection between theory and practice, proposing an alternative methodology that uses at its base the fundamentals of CP (Abstraction, Decomposition, Generalization and Algorithms) to promote the Teaching of Programming Logic using complementary teaching materials that are easy to use and adhere to the context of Professional and Technological Education (EPT). Making students realize the need and importance of developing programming skills to keep up with changes in society. This work, within ProfEPT, follows the line of research Educational Practices in Professional and Technological Education, within the scope of macroproject 1 - Methodological proposals and teaching resources in formal and non-formal teaching spaces at EPT. The methodology was proposed from a qualitative approach from a descriptive and observational perspective. Where an intervention took place with the application of a mini-course and problem solving. The analyzes of the students' opinions, performance and experiences were observed in a qualitative scope, where the research subjects were 25 students and 3 teachers from the Integrated High School course for IT Technician at the Irmã Ana Zélia da Fonseca State School of Professional Education. The instruments and procedures for data collection were questionnaires, also submitted for evaluation by the Research Ethics Committee of IFSertãoPE. The model for training, analyzing and interpreting the questionnaires and their resulting data was developed based on the content analysis proposed by Bardin (2010). In evaluating the data interpretations, we observed a specific evolution of the students, considering the development of skills. programming based on the challenges proposed by the intervention, where 40% validated the implementation of the alternative methodology using PC in the LP discipline and 52% also believed in the possibility of implementation in other disciplines, however, at the end of the study through analysis and comparison of the The results were guaranteed that using only the PC methodology would not be enough, as there is no unanimity among the research participants. It is therefore concluded that the best option is to use a PC methodology as a complement to LP and not as a methodology that will replace it.

**Keywords:** Professional Education. Computer Technician. Computing learning.

## LISTA DE FIGURAS

Figura 1 - Tela inicial do Scratch	26
Figura 2: Os Quatro Pilares do Pensamento Computacional	31
Figura 3 - Computação na educação básica	33
Figura 4 - Aplicação do minicurso	46
Figura 5 - Capa do Livro:um guia de boas práticas: Pensamento Computacional utilizando Scratch .	61

## LISTA DE QUADROS

Quadro 1 - Descritores de competências	43
Quadro 2 - Cronograma de execução de atividades	46
Quadro 3- Entrevista com os especialistas docentes	57

## LISTA DE GRÁFICOS

Gráfico 1- Busca de conteúdo a partir da metodologia de Pensamento Computacional	49
Gráfico 2 - Objetivos da disciplina de programação	50
Gráfico 3 - Tópicos de aprendizagem	51
Gráfico 4- Efeito da metodologia de Pensamento Computacional	52
Gráfico 5 - Implementação da metodologia de Pensamento Computacional	52
Gráfico 6 - Validação da metodologia de PC na disciplina de LP	53
Gráfico 7- Desempenho na disciplina de LP	54
Gráfico 8 - Contribuição dos pilares do PC	55

## LISTA DE FLUXOGRAMAS

Fluxograma 1 - Etapas da Pesquisa	37
Fluxograma 2 - Etapas da intervenção	37

## **LISTA DE ABREVIATURAS E SIGLAS**

ABNT – Associação Brasileira de Normas Técnicas

CNCT -Catálogo Nacional de Cursos Técnicos

EEEP - Escola Estadual de Educação Profissional

EMI - Ensino Médio Integrado

EPT - Educação Profissional e Tecnológica

LP - Lógica de Programação

PC - Pensamento Computacional

PROFEPT – Mestrado Profissional em Educação Profissional e Tecnológica

IFSertãoPE - Instituto Federal de Educação, Ciência e Tecnologia do Sertão  
Pernambucano

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>17</b>
<b>2 REFERENCIAL TEÓRICO</b>	<b>19</b>
2.1 - EPT Educação Profissional e Tecnológica	19
2.2 - O ensino de Lógica de Programação na disciplina de Linguagens de Programação no curso técnico.	21
2.3 Dificuldades no ensino de Lógica de Programação	23
2.4 Linguagem de programação visual	25
2.5 Pensamento Computacional (PC)	27
2.6 - Pensamento Computacional na Educação Básica	32
<b>3 METODOLOGIA DA PESQUISA</b>	<b>34</b>
3.1- Cuidados éticos que foram tomados ao realizar a pesquisa	38
3.2 - Coleta de dados	38
3.3- A construção do produto educacional	40
<b>4 ANÁLISE DOS DADOS</b>	<b>40</b>
4.1 - Resultados das Revisões Sistemáticas de Literatura	41
4.2 - A ferramenta Scratch	42
4.3 Análise dos resultados	43
4.4 Descrição e análise da intervenção	45
4.5 Entrevista com Especialistas	55
4.6 Recomendações para a aplicação do Produto Educacional no curso Técnico em Informática:	60
<b>5 CONSIDERAÇÕES FINAIS</b>	<b>62</b>
<b>REFERÊNCIAS</b>	<b>64</b>
APÊNDICE A - AVALIAÇÃO DA METODOLOGIA PC UTILIZANDO SCRATCH	73
APÊNDICE B – ANÁLISE DE METODOLOGIA	75
APÊNDICE C - ENTREVISTA SEMIESTRUTURADA COM ESPECIALISTAS	76
APÊNDICE D - PROBLEMAS ABORDADOS NA INTERVENÇÃO	77
APÊNDICE E - SEQUÊNCIA DIDÁTICA	82
APÊNDICE F - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO	84
APÊNDICE G - Matriz Curricular	85
APÊNDICE H - CARTA DE ANUÊNCIA	86
APÊNDICE I - TERMO DE ASSENTIMENTO LIVRE E ESCLARECIDO PARA MENORES DE 18 ANOS	87
APÊNDICE J - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA MAIORES DE 18 ANOS	92
APÊNDICE L - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA PAIS OU RESPONSÁVEIS	96
APÊNDICE M - Revisão Sistemática de Literatura 01	101

APÊNDICE N - Revisão Sistemática de Literatura 02	115
APÊNDICE O - um guia de boas práticas: Pensamento Computacional utilizando Scratch	137
ANEXO A – MATRIZ CURRICULAR SEDUC	181



## 1 INTRODUÇÃO

A Lógica de Programação é uma disciplina relatada como complexa e de difícil aprendizagem para os discentes. De acordo com Moreira (2020), o aprendizado de algoritmos é um dos problemas mais recorrentes para os estudantes que ingressam em cursos de tecnologia. Moreira (2020) ainda afirma que o educando muitas vezes não aprende a pensar e propor uma solução. Geralmente, ele é induzido de forma indireta a decorar um padrão e aplicá-lo quando necessário. Porém, em muitas situações, não é aquele protótipo que irá resolver o problema, mas sim, o modelo de pensamento que o formula. Desse modo, a Lógica de Programação é um componente curricular considerado de suma importância, entretanto, de laboriosa aprendizagem, pois foge do senso comum e do método de ensino, até então, aplicado.

Nesse sentido, a transformação do ensino da programação pode ser apoiada pela utilização de uma abordagem voltada ao uso do Pensamento Computacional (PC). De acordo com Wing (2006), o PC pode ser caracterizado como um processo que envolve a resolução de problemas, capacidade de projetar sistemas e a compreensão do comportamento do ser humano alinhados aos conceitos da ciência da computação. Bundy (2007) e Nunes (2011) dão ao PC a definição de habilidades utilizadas na construção de programas computacionais como metodologias para resolver problemas em diversas outras áreas. McMaster *et al.* (2010) entendem o PC como um dos mecanismos fundamentais da Ciência da Computação. Contudo, seu conceito não é exclusivamente aplicável na Computação, podendo ser expandido para outros contextos.

São conteúdos que precisam se tornar objetos de estudo de grupos de docentes, com a proposta de desenvolver e encontrar novas abordagens pedagógicas, simultaneamente, ao fundamento de potencializar a aprendizagem. Exposto o contexto e problematização destacados acima, este trabalho tem como finalidade propor uma metodologia alternativa por meio de uma abordagem voltada aos pilares do PC. Ao discorrer sobre a incorporação do Pensamento Computacional ao ensino básico, uma questão se abre para debate.

Seria esta, uma nova competência que integra e daria suporte a um aprendizado mais eficaz das áreas de difícil compreensão, como a Lógica de Programação? Nesse contexto, o objetivo é fazer os estudantes perceberem a necessidade e a importância de desenvolverem habilidades de programação, a partir

da aplicação prática em laboratório de todo o componente curricular de Lógica de Programação de computadores de um curso técnico em Informática, através da abordagem do Pensamento Computacional pela utilização da ferramenta *Scratch*.

Tratar cada item da ementa como forma de apresentar uma proposta para que o estudante siga em busca de solucionar problemas ao identificar os marcos ou pontos-chaves, os quais eles teriam que resolver a partir do Pensamento Computacional. O público-alvo de estudo da presente pesquisa foram os estudantes do primeiro ano do curso técnico em Informática, da turma de 2022 da Escola Estadual de Educação Profissional Irmã Ana Zélia da Fonseca, localizada no município de Milagres, interior do estado do Ceará.

O objetivo geral deste trabalho é investigar as contribuições da aplicação do Pensamento Computacional na disciplina de Lógica de Programação em um curso técnico em Informática, por meio da conexão entre teoria e prática através da resolução de problemas. Nesse ínterim, considera-se alguns pontos significativos:

- Identificar experiências com o Pensamento Computacional (PC) para o ensino de Lógica de Programação;
- Propor a adoção do Pensamento Computacional (PC) como estratégia metodológica para o ensino de Lógica de Programação;
- Avaliar a aprendizagem dos discentes acerca da Linguagem de Programação ministrada através do Pensamento Computacional;

Para isso, a metodologia foi conduzida por meio de uma abordagem quantitativa e qualitativa, combinando elementos de ambos enfoques com uma perspectiva descritiva e observacional. O estudo envolveu a implementação de minicursos como intervenção principal, complementada por questionários para coleta de dados. Durante a pesquisa, os participantes foram submetidos a minicursos que abordavam temas específicos relacionados ao objeto de estudo. As intervenções foram desenhadas para fornecer informações teóricas e práticas, permitindo uma compreensão aprofundada do fenômeno em análise.

O produto educacional oriundo deste trabalho trata-se de um guia de boas práticas de Lógica de Programação (Apêndice O) desenvolvido com base nos quatro pilares do Pensamento Computacional (Abstração, Decomposição, Generalização e Algoritmos), sendo aplicado durante o minicurso tratado no parágrafo anterior, como uma das alternativas de intervenção.

As seções a seguir, tratam do referencial teórico, metodologia, análise dos resultados e conclusões.

## **2 REFERENCIAL TEÓRICO**

A presente seção deste trabalho objetiva investigar a aplicação do Pensamento Computacional (PC) como metodologia de ensino de Lógica de Programação no ensino médio integrado ao técnico. Desse modo, o referencial teórico abordou os seguintes temas: A Educação Profissional e Tecnológica (EPT) no conceito das escolas estaduais de educação profissional do estado do Ceará, o ensino e aprendizagem de Lógica de Programação no curso técnico em Informática da rede de ensino do estado do Ceará e as principais habilidades relacionadas ao pensamento computacional.

Foi realizada também uma Revisão Sistemática de Literatura (RSL) para identificar os trabalhos recentes relacionados à aplicação do pensamento computacional como metodologia de ensino de linguagem de programação em contexto nacional. O diferencial da presente pesquisa desenvolvida em relação aos trabalhos estudados na RSL foi a questão da prática e teoria, no qual os trabalhos verificados apresentam apenas a teoria ou a prática de forma separada, sem necessariamente utilizar de conceitos de programação e codificação.

### **2.1 - EPT Educação Profissional e Tecnológica**

No início da década de 90 acontece um marco na educação brasileira: a promulgação da Lei de Diretrizes e Bases – LDB em 1996, implementa uma educação voltada para a cidadania e meios para progredir no trabalho (FRIGOTTO & CIAVATTA, 2003; COELHO, 2014). A Educação Profissional e Tecnológica (EPT) em nível de Ensino Médio, em 1997, tem uma alteração significativa nas suas bases com a promulgação do Decreto nº 2.208, de 17 de abril de 1997, que versava sobre a separação do ensino técnico de nível médio e o ensino regular, sendo prioritariamente ofertada na forma concomitante e subsequente e, assim, o ensino médio obtém um caráter propedêutico e dual (MENDES, 2011; PACHECO, 2012).

Ramos (2002) afirma que as transformações não aconteceram apenas no formato de oferta da EPT. Houve também uma reformulação curricular, pautada em

competências para traçar o perfil do futuro profissional através de metodologias aportadas em resolução de problemas e projetos. Dessa maneira, a formação técnica deve proporcionar ao estudante a integração também com sua instrução básica escolar de ensino médio.

Os cursos técnicos são formações de nível médio com objetivo de ingressar os discentes de uma forma mais rápida no mercado (DEITOS; LARA; DE BARROS, 2016). Ao contrário das graduações que têm uma visão mais ampla de conhecimento, fornecendo uma formação mais qualificada para exercer atividades, tais como pesquisadores e cientistas, e dar continuidade aos estudos, como por exemplo, mestrado e doutorado (SEVERINO, 2008). Por esse motivo, o intuito dos cursos técnicos é fornecer conhecimentos mais práticos em relação às graduações, objetivando uma inserção mais rápida no mercado de trabalho.

Nesse sentido, a computação quando trabalhada através da Educação Profissional e Tecnológica (EPT) como meio e fim, possibilita ao estudante a construção do seu próprio conhecimento e o desenvolvimento de sua capacidade resolutive de problemas. Assim, não tem como falar de EPT sem relacionar com tecnologia da informação, informática e computação, pois o mundo caminha para um ensino cada vez mais conectado.

A utilização da programação como instrumento pedagógico e didático favorece a aprendizagem ativa, uma vez que a interação estabelecida entre as ações dos estudantes e o *feedback* do programa favorece a dinâmica. Na perspectiva de Valente (1999), o qual trata da formação de professores para a área da informática na educação, a intenção é preparar os participantes para o uso do computador em sala de aula. Sendo assim, as atividades versam na aplicabilidade do mesmo como mecanismo para que os estudantes possam construir seu conhecimento.

Almeida (2009) afirma que o estudante sintetiza os passos necessários para atingir determinado objetivo, sendo direcionado a transformar seus conhecimentos em procedimentos. O ensino dos princípios da computação proporciona o desenvolvimento do Pensamento Computacional, sendo uma habilidade fundamental para todos, não só de cientistas da Computação.

Almeida (2009), Severino (2008) e Deitos, Lara e de Barros (2016) discutem e conceituam a tecnologia, e propõem a ideia de que ela possa ser uma vertente aliada ao ensino de forma alternativa às metodologias tradicionais, dado o contexto em que está inserida atualmente, cujas transformações são constantes. Em tempo real, se faz necessário que sua aplicação esteja bem fundamentada no ambiente escolar. Porém, apenas utilizá-la como fim não é o suficiente para um bom desenvolvimento, pois precisa fazer sentido ao alunado. O ensino de Computação, às vezes é confundido com as aulas de Informática, que tem em seu propósito instruir o estudante quanto ao uso de softwares. Restringindo, dessa maneira, o ensino de Computação ao simples manuseio de aplicativos e equipamentos eletrônicos.

Portanto, a utilização do Pensamento Computacional na educação básica não visa apenas a empregabilidade e desenvolvimento econômicos, mas também a construção de habilidades. A seção seguinte irá tratar do Pensamento Computacional a partir dos seus fundamentos, e dentre algumas citações, estarão as de Papert (1980) e Wing (2014), considerados como pioneiros nos estudos dessa ferramenta.

## **2.2 - O ensino de Lógica de Programação na disciplina de Linguagens de Programação no curso técnico.**

A partir do ano de 2008 o estado do Ceará implantou uma rede de Escolas de Educação Profissional (EEEP) em tempo integral, objetivando assim uma formação profissional junto ao ensino médio regular, oportunizando aos jovens cearenses uma nova perspectiva de vida. Ofertando diversos cursos, sendo um deles o de técnico em Informática, cuja ementa traz disciplinas que envolvem o raciocínio lógico e a Matemática como principais mecanismos para a resolução de problemas propostos. Segundo Gomes *et al* (2017), a Lógica de Programação e o pensamento algorítmico é considerado por muitos estudos como sendo uma tarefa de relativa complexidade, por envolver não somente a sintaxe e semântica das linguagens como também a habilidade resolutiva de problemas.

O Catálogo Nacional de Cursos Técnicos (CNCT) traz diversos cursos que possuem a disciplina Lógica de Programação como componente curricular de suas respectivas ementas, a exemplo do eixo Informação e Comunicação (Computação

Gráfica, Jogos Digitais, Redes de Computadores, Desenvolvimento de Sistemas e Informática).

Os cursos técnicos de nível médio objetivam o ingresso de discentes de forma mais rápida ao mercado de trabalho. (Deitos; Lara; de Barros, 2016). Segundo Severino (2008), os cursos técnicos diferem das graduações tradicionais que possuem uma formação mais ampla e generalista. Dessa forma, o objetivo dos cursos técnicos de nível médio é proporcionar um tipo de conhecimento mais prático em relação às graduações, em vistas a um ingresso mais rápido no campo de trabalho.

De acordo com a SEDUC (2018), o curso Técnico em Informática das Escolas Estaduais de Educação Profissional (EEEPs) é confeccionado para discentes egressos do 9º ano do ensino fundamental II. Esses estudantes em sua grade curricular, nunca tiveram nenhuma disciplina de programação, lógica ou qualquer outra disciplina relacionada à computação. O propósito do curso é que conforme avancem nas disciplinas, possam aprender a desenvolver softwares para computadores, internet e celulares, bem como páginas virtuais, seguindo especificações e paradigmas das linguagens de programação.

O currículo do curso de Ensino Médio Integrado (EMI) em Técnico de Informática obedece ao normativo legal do Guia Curricular Nacional do Ensino Secundário e Técnico Profissional, aos parâmetros do Currículo Nacional do Ensino Secundário e aos referenciais do Currículo Nacional do Ensino Profissional.

A organização do curso está estruturada como uma matriz curricular formada com base no conhecimento científico e técnico, que é a seguinte:

- Educação geral, educação básica, que combina especialidades das três áreas de competência do ensino médio (línguas e códigos e suas tecnologias, disciplinas humanitárias e suas tecnologias e disciplinas naturais, Matemática e suas tecnologias) com educação profissional, seguindo as peculiaridades do currículo integrado;
- Educação profissional combinando áreas especializadas em TI;
- Uma parte multifacetada que combina disciplinas com o objetivo de melhor compreender as relações trabalho-vida existentes e articulá-las com o conhecimento acadêmico.

O curso técnico de informática integrado ao ensino médio é organizado em

regime de série semestral, no qual a carga horária de disciplinas técnicas é de 1240 horas e 300 horas de estágio em campo, totalizando 1.540 h/a.

As turmas contam com 45 vagas, as quais são disponibilizadas nas turmas de primeiro ano das Escolas Estaduais de Educação Profissional (EEEPs) do estado do Ceará, no primeiro semestre de cada ano, e são preenchidas de acordo com os determinados critérios:

- Os estudantes devem ter concluído o 9º ano do ensino fundamental;
- Ter disponibilidade para cumprir jornada integral das 7h às 17h, de segunda a sexta-feira;
- Ter no mínimo 14 anos completos na data do censo escolar;
- Ter ciência das normas de funcionamento e oferta do curso técnico de sua preferência;
- Enviar os documentos exigidos pela escola;
- Obter classificação no âmbito das vagas abertas, através das notas do 6º ao 9º ano do ensino fundamental, de acordo com a média geral das disciplinas estudadas.

O curso abrange uma variedade de tópicos, incluindo fundamentos de computação, programação de computadores, redes de computadores, sistemas operacionais, banco de dados, desenvolvimento de software, segurança da informação e suporte técnico. Entretanto, é preciso verificar como a metodologia de ensino de Lógica de Programação está sendo aplicada.

Para dar continuidade a esta pesquisa, as próximas subseções apresentam as dificuldades no ensino de LP e como o Pensamento Computacional pode adentrar ao contexto.

### **2.3 Dificuldades no ensino de Lógica de Programação**

De acordo com Bennedsen e Caspersen (2007), muitas reprovações e desistências se devem à dificuldade de aprender programação de computadores.

Os estudantes ingressantes no curso técnico em Informática devem ser melhor acompanhados para diminuir alguns fatores que contribuem ou reprovam os mesmos, entre eles: Deficiência de abstração nos problemas utilizados em aula, dificuldades de aprendizagem em linguagens de programação e na aplicação do conteúdo na prática, falta de identidade com o curso, ausência de motivação e

domínio da matemática.

O referido curso é muito procurado por jovens sem nenhuma experiência profissional, que em sua maioria, optam pelo técnico em informática sem saber quais são as disciplinas trabalhadas. São atraídos pela grande quantidade de recursos disponíveis referentes à tecnologia, sobretudo influenciados também pelas redes sociais, utilização para entretenimento e lazer, acreditando que os conhecimentos necessários para se manterem cursando serão básicos e mínimos.

Neste contexto, é necessária mudança de motivação, identidade com o curso, compreensão das mudanças do mercado de trabalho com a transformação digital, e acima de tudo metodológico para superar essas dificuldades e alcançar sucesso no ensino desta disciplina.

Uma das primeiras dificuldades verificadas é a escassez de tempo para o exercício dos conteúdos teóricos e adoção de metodologia tradicional para ensinar os conteúdos práticos. De acordo com Aureliano, Tedesco e Giraffa (2016) os estudos enviesados em uma metodologia tradicional, por muitas vezes são superficiais, decorando-os apenas. Outro problema relacionado ao planejamento do tempo na abordagem tradicional é a falta de *feedback* durante a resolução dos exercícios, o que está diretamente relacionado à motivação do estudantes (ROBINS; ROUNTREE; ROUNTREE, 2003).

Outro contratempo percebido é a falta de abstração das operações de criação do algoritmo. Barbosa, Ferreira e Costa (2014) e Henrique e Tedesco (2017) explicam que a dificuldade de programar decorre do conceito de abstração, pois as atividades de programação não estão relacionadas ao cotidiano dos discentes.

No entanto, Gomes, Henrique e Mendes (2008) mencionam, ao falar sobre dificuldades abstratas, que a ansiedade obriga o estudante a praticar imediatamente, e acaba deixando para trás os fundamentos da resolução de problemas. Santos e Gama (2019) recomendam o uso de computação desplugada para melhor abstrair esses conceitos. No entanto, o ensino tradicional de Lógica de Programação se baseia em abordagens convencionais, como aulas expositivas, leituras de materiais didáticos, resolução de exercícios e avaliações. O foco está na transmissão de conhecimento do professor para os alunos, com menos ênfase em atividades e experiências práticas.

todavia, a disciplina de Lógica de Programação (LP) durante os anos de 2016 até 2022 foi ministrada pelo autor desta dissertação no qual foi verificado um imenso



bloqueio para interpretação e resolução de problemas que envolvessem algoritmos e programação, constatando que a grande maioria das turmas tem essa dificuldade.

## 2.4 Linguagem de programação visual

Os aportes destinados à elaboração de estratégias, visando contribuir para o aprendizado de programação, têm experimentado um aumento considerável nos últimos anos. Souza, Silva e Barbosa (2016, p. 48) conduziram uma pesquisa para identificar os desafios e obstáculos no ensino e na aprendizagem de programação. Como parte de sua amostragem, examinaram 70 estudos relacionados ao tema, abrangendo o período de 2010 a 2014.

Entre os principais resultados desse estudo, destaca-se a constatação de que muitos estudantes enfrentam dificuldades em compreender conceitos, tais como, ponteiros, recursão e variáveis. Entretanto, outros conseguem entender essas definições, porém enfrentam entraves ao aplicá-los, efetivamente, durante o desenvolvimento de programas. Além disso, os resultados revelam uma escassez de motivação por parte dos estudantes para a realização de atividades de programação. A pesquisa também identificou alternativas que estão sendo elaboradas pelos professores para superar essas dificuldades, como a adoção de visualizações de programas e algoritmos, jogos educativos e a criação de ambientes pedagógicos voltados para o ensino e aprendizagem de programação.

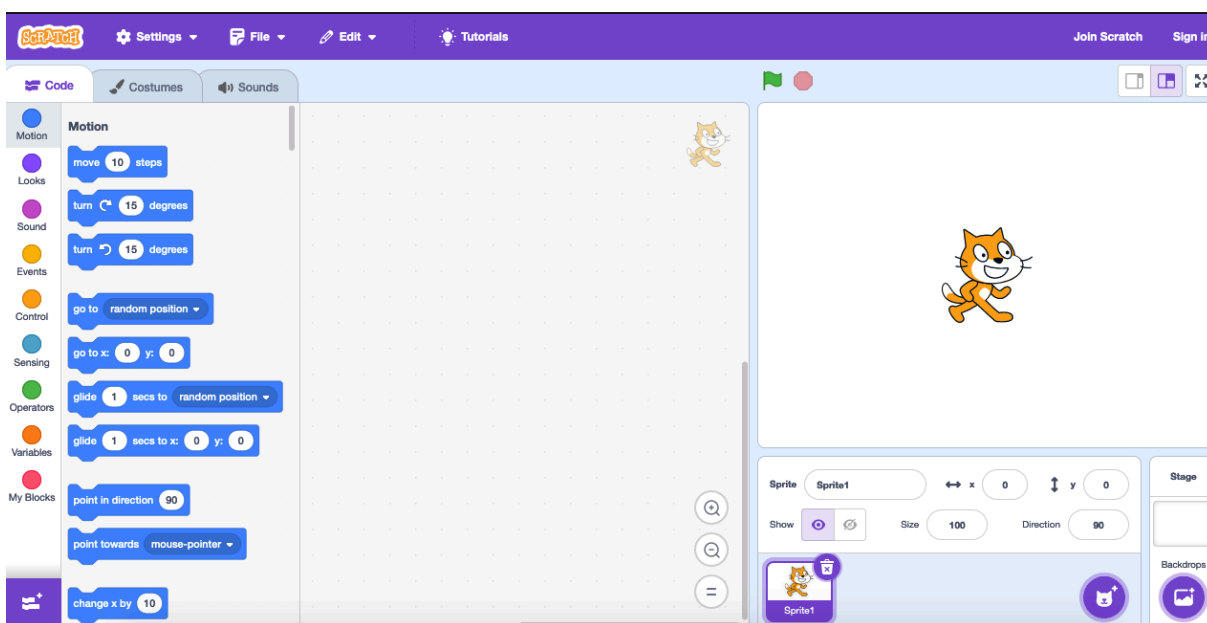
Um levantamento com propósito semelhante foi conduzido por Silva *et al.* (2015, p. 183) e revelou que, dentre as abordagens adotadas pelos professores no ensino de programação, há uma inclinação para o desenvolvimento de recursos destinados ao ensino de algoritmos. Além disso, pesquisas correlatas indicam que atividades como jogos, utilização de robótica, ferramentas, metodologias e técnicas empregadas no ensino de programação têm impacto positivo na motivação dos estudantes e em sua atitude em relação aos materiais de aprendizagem. Entre os resultados destacados na pesquisa realizada pelos autores, há evidências de que a adoção de linguagens de programação visual auxilia na simplificação do processo de aprendizagem, ajudando os estudantes na resolução de problemas.

O *Scratch* (figura 1) é uma linguagem de programação visual desenvolvida pelo grupo *Lifelong Kindergarten* do *MIT Media Lab*, com o objetivo de permitir que jovens e iniciantes aprendam a programar de forma intuitiva e divertida,

proporcionando um ambiente gráfico de programação em blocos, podendo encaixá-los de modo a fazer sentido. Desde o seu lançamento em 2007, o *Scratch* tem sido amplamente adotado em escolas e instituições educacionais como uma ferramenta poderosa para promover a criatividade, o Pensamento Computacional (PC) e a aprendizagem colaborativa.

A figura 1 apresenta a tela principal do *Scratch*, na qual os blocos são identificados e destacados por cores, (movimento, visual, áudio, caneta, comando, detectores, operações e variáveis), o que simplifica a compreensão no processo de aprendizado. Cada ação é associada a um *script*, arrastando esses blocos para a área de trabalho, conforme demonstrado.

Figura 1 - Tela inicial do Scratch



Fonte: *Scratch MIT*, 2023.

Resnick *et al.* (2009) indicam que o *Scratch* promove habilidades cognitivas, como resolução de problemas, pensamento crítico e raciocínio lógico. O *Scratch* faz parte de um conjunto de recursos que podem promover o domínio tecnológico, as aptidões e habilidades fundamentais para a prática da cidadania, como a colaboração, o raciocínio computacional e inovador. (RESNICK apud MARQUES, 2009). Além disso, o *Scratch* tem o potencial de desenvolver competências

socioemocionais, como a colaboração, a perseverança e a autoexpressão criativa (Kafai & Burke, 2014). Segundo Chadha (2014), o *Scratch* se popularizou nos últimos anos, usando blocos em um ambiente virtual para desenvolver jogos e animações, sendo, portanto, uma abordagem de aprendizagem criativa, elemento fundamental na educação.

Essa metodologia encoraja os estudantes a criarem projetos pessoais significativos e a compartilhá-los com a comunidade. Através da criação de histórias interativas, jogos e animações, os estudantes podem expressar sua criatividade, explorar ideias e conceitos, e se engajar em um processo de construção de conhecimento (Brennan & Resnick, 2012).

Diversos estudos de caso têm explorado as aplicações do *Scratch* em diferentes contextos educacionais. Noss *et al.* (2018) investigaram o uso do mesmo em escolas primárias, destacando o impacto positivo na motivação e no envolvimento dos estudantes. Outro estudo de Gresalfi *et al.* (2020) averiguou como o *Scratch* pode apoiar a aprendizagem matemática em uma sala de aula de terceiro grau. Desse modo, a utilização deste como ferramenta de implementação das habilidades do PC pode ser uma alternativa ao ensino de Lógica de Programação.

Para mais, a seção a seguir apresenta conceitos sobre o Pensamento Computacional e como reproduzi-los a partir da utilização do *Scratch*, sendo possível, assim, desenvolver o raciocínio lógico adequado para a construção de soluções.

## **2.5 Pensamento Computacional (PC)**

O Pensamento Computacional (PC) refere-se a um conjunto de habilidades cognitivas e processos mentais que são fundamentais para a resolução de problemas e a tomada de decisões na era da computação. Ele envolve a capacidade de decompor um problema complexo em partes menores, identificar padrões e relações, projetar soluções algorítmicas e utilizar abstrações para representar informações. O termo introduzido em Wing (2006) é um método para ensinar a solucionar problemas, conceber sistemas e compreender o comportamento humano, por meio de conceitos da Ciência da Computação ou, como colocado por Paula, Valente e Burn (2014) como um modo de pensar e analisar de forma específica uma situação ou artefato, independente de tecnologia.

Ao longo dos últimos anos, o PC tem recebido crescente atenção tanto na educação quanto na pesquisa acadêmica. Pesquisadores têm explorado sua importância no desenvolvimento de habilidades transversais e na promoção do raciocínio lógico, da criatividade e da resolução de problemas. A Sociedade Brasileira de Computação (SBC, 2017) define PC como a capacidade de sistematizar, representar, analisar e resolver problemas computacionais e do cotidiano. O sítio *Google For Education* (2015) apresenta o PC como um mecanismo de resolução de problemas que inclui diversas características, a exemplo da autonomia em utilizar ferramentas digitais, planilhas, textos, sistemas em nuvem dentre outros. Em resumo, o PC é essencial no desenvolvimento de sistemas computacionais, mas também pode ser utilizado para dar suporte em todas as disciplinas do conhecimento humano, a exemplo de Ciências e Matemática.

Para Wing (2019) o PC é o processo de pensamento envolvido na formulação de problemas e suas soluções, de modo que estas sejam representadas e efetivamente realizadas por um agente de processamento de informações. Barr, V., & Stephenson, C. (2019) complementam que o PC envolve um conjunto de habilidades e práticas mentais que todos podem usar para resolver problemas, projetar sistemas e entender o comportamento humano, e é um componente essencial da base da Ciência da Computação. Grover e Pea (2019) afirmam que o ensino do PC nas escolas pode promover o desenvolvimento de habilidades cognitivas, como resolução de problemas, pensamento crítico e criatividade, que são essenciais para o sucesso no século XXI.

Cavalcante, Costa e Araújo (2016, p.1117) argumentam que o PC consiste em "processo com vista à resolução de problemas por meio de conceitos, recursos e ferramentas computacionais", bem como complementam que é uma forma de aplicar os conhecimentos computacionais em qualquer campo do saber, uma vez que não são, exclusivamente, da computação. Shute, Sun e Asbell-Clarke (2017) declaram também que essas habilidades pertencentes ao PC são a base conceitual e necessária para a resolução eficiente de problemas, uma vez que com ou sem a assistência de computadores, as soluções podem ser generalizadas em diferentes contextos.

O PC é reconhecido como uma habilidade fundamental na era digital e tem recebido destaque nas diretrizes educacionais em todo o mundo, incluindo a Base Nacional Comum Curricular - BNCC (Brasil, 2018). Ele representa uma abordagem abrangente para resolver problemas complexos, promovendo o desenvolvimento de habilidades cognitivas essenciais e competências transversais. Ainda de acordo com a BNCC (Brasil, 2018), o PC envolve uma série de habilidades, como a capacidade de analisar e decompor problemas, identificar padrões, criar algoritmos, abstrair informações e avaliar soluções. Todas estas consideradas imprescindíveis para os estudantes enfrentarem os desafios do século XXI, tanto na vida pessoal quanto profissional.

O PC vai além da programação e do uso de computadores. Ele envolve a aplicação de princípios computacionais em diferentes contextos, permitindo que os estudantes resolvam problemas complexos em diversas áreas do conhecimento. Isso significa que o Pensamento Computacional pode ser integrado a disciplinas como Matemática, Ciências, Linguagens e Artes, possibilitando uma abordagem interdisciplinar, promovendo a aprendizagem significativa, além de colaborar na formação de cidadãos críticos e participativos na sociedade digital. Também incentiva os estudantes a desenvolverem habilidades de colaboração, comunicação, resolução de problemas e pensamento criativo. Por isso, capacita os estudantes a entenderem e participarem ativamente de um mundo cada vez mais orientado pela tecnologia, pois promove a ética digital e a segurança online.

A BNCC destaca a importância do desenvolvimento profissional dos educadores para a implementação do PC de forma efetiva, destacando que eles precisam estar preparados para integrar o PC em sua prática pedagógica. Para tanto, utilizando metodologias ativas, recursos digitais e propiciando a resolução de problemas autênticos. Ademais, a BNCC incentiva a criação de parcerias com instituições e profissionais da área de tecnologia, visando enriquecer as experiências de aprendizagem dos estudantes.

Wing (2006) afirma que o PC deve ser uma habilidade inerente e básica a ser ensinada a crianças, assim como ler ou escrever, por exemplo. Alguns currículos, enquanto componentes extracurriculares, já trazem em suas práticas o ensino de programação tanto no ensino fundamental quanto no ensino médio. Práticas como

estas podem levar o estudante ao desenvolvimento de aptidões para abstração e, conseqüentemente, à resolução de problemas, experiências muito importantes no atual contexto tecnológico. Sendo assim, começar a utilizar o PC como metodologia de ensino poderá impactar positivamente na construção desses saberes.

O processo de ensino tem como objetivo fazer com que o outro adquira o conhecimento, utilizando-se do professor como transmissor de técnicas e informações para que esse processo seja viabilizado. Na perspectiva de Libâneo (1994), o processo de ensino deve estabelecer exigências e expectativas que os estudantes possam cumprir, e com isso, mobilizarem suas energias. A relação de ensino e aprendizagem não deve ter como base a memorização, deve-se tornar os estudantes aptos para aprenderem os conteúdos, de forma que possam tomar um papel autônomo. Assim, o professor pode ser considerado um facilitador, em alguns casos.

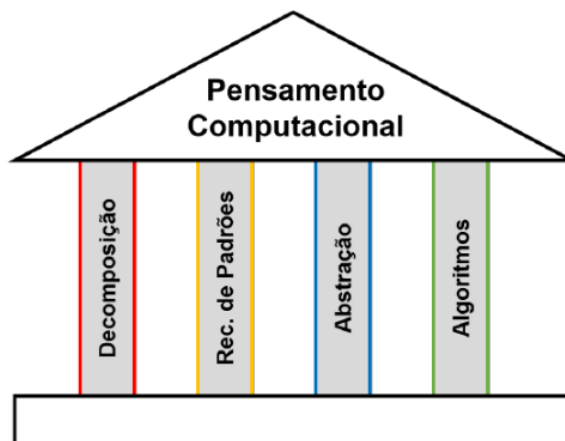
Ainda, de acordo com Libâneo (1994), aprender é o processo de assimilação de qualquer forma de conhecimento, desde o mais simples, o qual a criança aprende a manipular os brinquedos, a fazer contas, lidar com coisas, nadar, andar de bicicleta, etc. Segundo o autor, para que se possa existir aprendizagem é preciso um processo de assimilação ativa, e que se efetive através de atividades práticas em várias modalidades e exercícios, nos quais se possam verificar a consolidação e aplicação usual de conhecimentos e habilidades.

Diante do que foi apresentado pelos teóricos, todos concordam que o PC é uma forma eficiente de resolução de problemas, desde o processo de abstração da realidade até a implementação, de fato, da solução. Em outras palavras, as habilidades inerentes ao PC auxiliam em diversas situações. Autores como Brackmann (2017), Liukas (2015) e instituições como a *Google For Education* (2015) e *BBC Learning* (2015), visualizam e apontam quatro habilidades distintas como pilares do Pensamento Computacional: decomposição, reconhecimento de padrões, abstração e a escrita do algoritmo. A partir desses fundamentos é possível afirmar que:

[...] o PC envolve tomar o problema complexo e dividi-lo em uma série de problemas pequenos e mais gerenciáveis (decomposição). Cada um desses problemas menores pode ser examinado individualmente, considerando como problemas semelhantes que foram resolvidos anteriormente (reconhecimento de padrões) e focalizando

apenas os detalhes importantes, ignorando a informação irrelevante (abstração). Em seguida, etapas simples ou regras para resolver cada um dos problemas menores podem ser projetadas (algoritmos) (BBC LEARNING, 2015, s/p.).

Figura 2: Os Quatro Pilares do Pensamento Computacional



Fonte: (Brackmann, 2017, p.33)

Desenvolvido pelo Centro de Inovação para a Educação Brasileira (CEIB) temos o Currículo de Referência - Itinerário Formativo em Tecnologia e Computação para o Ensino Médio, que também explora os quatro pilares do PC, pontuando os aspectos mais relevantes (CIEB, 2020, p.21):

- **Abstração:** o conceito envolve a filtragem dos dados e sua classificação, ignorando elementos que não são necessários, visando os que são relevantes. Envolve também formas de organizar informações em estruturas que possam auxiliar na resolução de problemas.
- **Algoritmos:** agregam todos os demais. É um plano, uma estratégia ou um conjunto de instruções claras e necessárias para a solução de um problema. Em um algoritmo, as instruções são descritas e ordenadas para que o objetivo seja atingido; podem ser escritas em formato de diagramas, pseudocódigo (linguagem humana) ou escritos em códigos, por meio de uma linguagem de programação.
- **Decomposição:** trabalha o processo pelo qual os problemas são divididos em partes menores e mais fáceis de resolver. Compreende também a prática de analisar problemas, a fim de identificar quais partes podem ser separadas, e também as possíveis de serem reconstituídas para a solução de um problema global. Essa prática também viabiliza aumentar a atenção aos detalhes.
- **Reconhecimento de Padrões:** trabalha a identificação de características comuns

entre os problemas e suas soluções. Resulta do fato de realizar a decomposição de um problema complexo para encontrar padrões entre os subproblemas gerados. Esses padrões são similares ou têm características, das quais alguns dos problemas compartilham e que podem ser explorados, com intenção de serem solucionados de forma mais eficiente.

Ao verificar as habilidades e competências do PC, pode-se constatar a sua aplicabilidade em diversos contextos, dentre eles a Educação, sendo possível utilizar referências ao PC nas práticas docentes. Assim, a seção a seguir, apresenta o Pensamento Computacional no contexto da Educação Básica.

## **2.6 - Pensamento Computacional na Educação Básica**

No Brasil, existe o Currículo de Referência em Tecnologia e Computação (CRTC), desenvolvido com base em um estudo encomendado pelo Centro de Inovação para a Educação Brasileira (CIEB), que apresenta uma proposta fundamentada nos parâmetros da BNCC e em referências internacionais, com sugestões de diretrizes para a inclusão do tema Tecnologia e Computação nos currículos das redes de ensino.

O CRTC delinea três eixos principais: Cultura digital, Tecnologia Digital (ou Mundo Digital) e Pensamento Computacional, delineando um total de 147 habilidades a serem cultivadas pelos estudantes. O documento orienta os professores da Educação Básica por meio de práticas pedagógicas, buscando promover o desenvolvimento dessas competências.



Figura 3 - Computação na educação básica



Fonte: CIEB, 2020. Disponível em: < <https://curriculo.cieb.net.br/> >

Queiroz, Sampaio e Santos (2019) debatem a implementação do PC nas escolas em duas linhas argumentativas: a preparação para o mercado de trabalho e a formação destinada a desenvolver habilidades nos estudantes relacionadas às áreas de Ciências, Tecnologias, Engenharias, Artes e Matemática. Destaca-se aqui o esclarecimento à luz da EPT, que essa primeira possibilidade será voltada ao mundo do trabalho. A BNCC destaca a integração do PC ao designá-lo como uma habilidade e/ou competência a ser cultivada ao longo do ensino. Portanto, a responsabilidade recai, principalmente, sobre os professores, não apenas para abordar os conteúdos tradicionais, mas também para incorporar os princípios do PC de forma efetiva no ambiente da sala de aula.

### 3 METODOLOGIA DA PESQUISA

De acordo com Apolinário (2012), a pesquisa científica é caracterizada como um processo sistemático, controlado, empírico e crítico de investigação que tem como objetivo produzir conhecimentos novos e verificáveis sobre a realidade. Apolinário (2012) complementa que a mesma é sistemática, pois segue uma série de procedimentos e etapas bem definidas, desde a formulação da hipótese até a análise dos resultados. É controlada, pois busca minimizar os erros e as influências externas que possam afetar os resultados, é empírica, pois se baseia na observação da realidade, em dados e informações coletados por meio de instrumentos e

técnicas adequadas.

Além disso, a pesquisa é crítica, pois avalia e questiona os resultados obtidos, buscando a validade e a confiabilidade dos mesmos. Aponta ainda como outra característica importante, a busca pela produção de conhecimentos novos e verificáveis, os quais possam ser validados por outros pesquisadores. Para isso, é fundamental a utilização de métodos e técnicas confiáveis e rigorosas, além de apresentar os resultados de forma clara e objetiva.

A pesquisa do tipo quanti-qualitativa, também conhecida como pesquisa mista, é uma abordagem que combina elementos tanto da pesquisa quantitativa quanto da pesquisa qualitativa. Essa metodologia busca obter uma compreensão mais abrangente e profunda do fenômeno estudado, explorando tanto aspectos mensuráveis quanto aspectos subjetivos e interpretativos. Integrar essas duas metodologias, permite aos pesquisadores examinar as relações causais e os padrões estatísticos por meio de dados numéricos, ao mesmo tempo em que busca compreender os significados, experiências e perspectivas dos participantes através de métodos qualitativos.

Ao longo do trabalho foram elaboradas duas revisões sistemáticas de literatura (RSL) objetivando entender e identificar como o PC está sendo trabalhado e suas práticas, utilizando o *Scratch* como ferramenta para o ensino. A primeira RSL foi construída através de trabalhos limitados ao idioma português (nacional), a segunda contemplou um espaço maior, abrangendo trabalhos com o idioma inglês.

O desenvolvimento das fases das revisões de literatura foi baseado em aspectos da metodologia PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) baseada no trabalho de Moher D *et al* (2015). Esta abordagem visa ajudar os autores a melhorar o desenho de revisões sistemáticas e meta-análises e pode ser usada na preparação de outros relatórios de revisão. Durante a construção da Revisão Sistemática da Literatura (RSL) foram definidas as palavras-chave: Pensamento Computacional, Lógica de Programação, Linguagem de Programação, Ensino de Computação e *Scratch*, os quais, posteriormente, se tornaram termos de busca. Dessa forma, a pesquisa foi realizada em bases acessíveis disponíveis na internet pelos portais da CAPES, *Google Scholar*, *SBC Open Lib*, *Scopus* e *Plataforma Sucupira*, *Science Direct*, através de teses e dissertações relacionadas.

Após a escolha das principais palavras-chave, a estrutura da pesquisa foi montada nas plataformas *SBC Lib Online*, *Google Scholar*, *Scopus*, utilizando a ferramenta de revisão sistemática *StArt*. Nas quais, foram buscadas apenas obras em português na primeira RSL e em inglês na segunda. Os resultados e as respectivas revisões podem ser visualizadas nos anexos a e b.

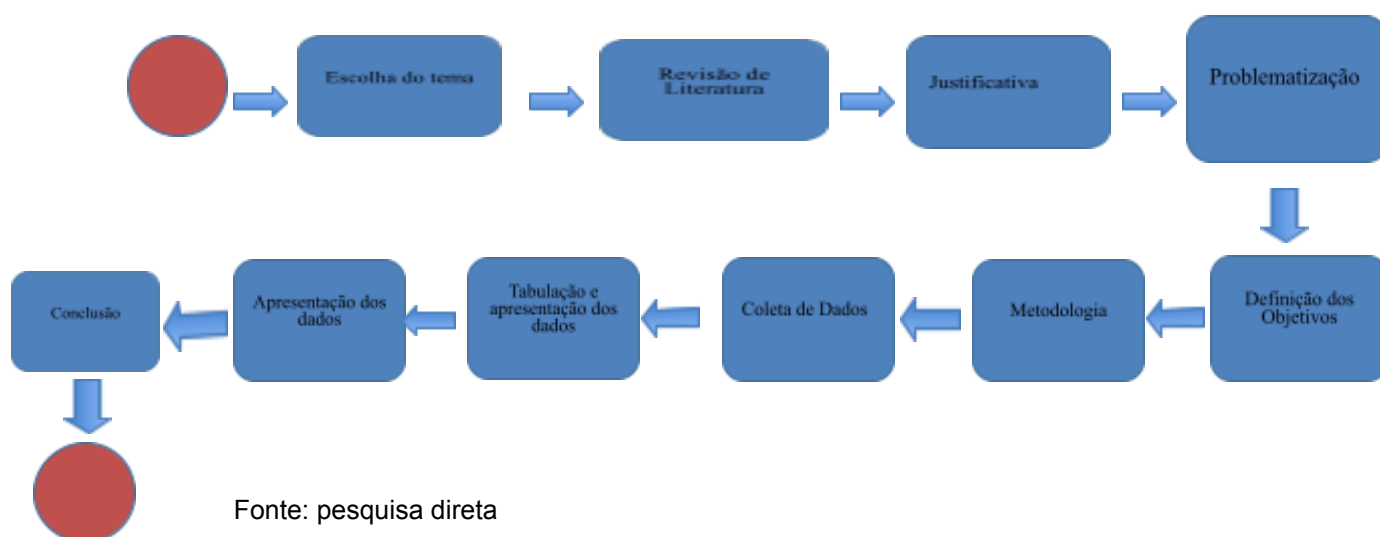
As RSL foram desenvolvidas considerando práticas pedagógicas, e as que descrevem e/ou implementam a utilização de meios auxiliares de ensino. A pesquisa consiste em uma intervenção pedagógica aplicando o PC como alternativa ao ensino de Lógica de Programação.

Esse levantamento embasou o planejamento e oferta do minicurso ministrado, o qual compõe o produto educacional deste trabalho, como um guia de boas práticas utilizando o PC e a ferramenta *Scratch*.

Em seguida, um questionário foi utilizado para verificar o nível de conhecimento dos estudantes sobre a metodologia de resolução de problemas de computador. Após isso, a turma obteve um curso curto para entender melhor o que são os computadores. Na etapa seguinte, foram aplicados problemas envolvendo Lógica de Programação em caráter experimental para que eles desenvolvessem, utilizando a metodologia de PC através da ferramenta *Scratch*, em sequência, foi feita a coleta de dados via questionário e sua interpretação.

Para avaliar melhor os resultados, também foi realizada uma apresentação do tema PC por meio de sequência didática para especialistas, docentes do EMI (Ensino Médio Integrado) de outras instituições, a fim de validar a partir de um questionário, suas percepções em referência à aplicação da metodologia. A pesquisa ocorreu na turma de primeiro ano do ensino médio integrado ao curso de Informática, da Escola Estadual de Educação Profissional Irmã Ana Zélia da Fonseca, localizada na cidade de Milagres, no interior do Ceará. O critério de escolha da instituição deu-se pelo fato de que o autor desta pesquisa é docente da mesma, de modo que ao longo de sua trajetória percebeu uma grande dificuldade por parte dos estudantes em relação aos conteúdos que tinham a Lógica de Programação como componente curricular. Já na escolha dos docentes, para a pesquisa, seguiu-se o critério da experiência, e foram selecionados três docentes de instituições de Educação Profissional da rede pública do Estado do Ceará que tinham vivências na EPT há mais de 10 anos. O fluxograma 1 abaixo, apresenta as etapas da pesquisa:

Fluxograma 1 - Etapas da Pesquisa



A turma tem como seu primeiro componente curricular, a disciplina de Lógica de Programação, no qual foi trabalhada de forma integral por meio da abordagem de Pensamento Computacional. Depois do experimento de aplicação, foi realizada uma coleta de dados sobre a opinião e validação dos estudantes, através de formulários do *Google* sobre a metodologia de Pensamento Computacional utilizando *Scratch*. Abordando acerca de usabilidade, experiência, praticidade, facilidade, manutenibilidade, e também aspectos técnicos conteudistas relacionados a lógica e programação. O fluxograma 2 a seguir, apresenta as etapas inerentes apenas à metodologia da intervenção, a partir do entendimento do contexto.

Fluxograma 2 - Etapas da intervenção



Fonte: Pesquisa direta.

Tal política de educação profissional articula o Ensino Médio à educação profissional, ao mundo do trabalho e ao ensino superior. Neste contexto, em 2012, foi criada, no município de Milagres, a 92ª Escola Estadual de Educação Profissional. A mesma está embasada em um Projeto Político-Pedagógico (PPP) que propõe desenvolver em seus estudantes as competências e habilidades imprescindíveis ao ingresso no ensino superior, inserção no mercado de trabalho e organização e gestão eficiente de negócio autônomo.

O público da pesquisa foi composto por 25 estudantes com 16 anos de idade e que estavam no 1º ano do curso técnico em Informática, pois já tinham conhecimento e cursavam alguma disciplina sobre programação. Como citado anteriormente, também foram alvo de estudo da presente pesquisa, três profissionais especialistas que validaram os questionários a partir de suas experiências na EPT.

O questionário do apêndice A trata sobre a coleta de dados inicial aplicado aos estudantes a respeito da aplicação da metodologia de Pensamento Computacional utilizando *Scratch*. O apêndice B, por sua vez, traz o exemplo do questionário de avaliação do ensino de Lógica de Programação por meio da metodologia tradicional. Já o apêndice C apresenta um questionário modelo para entrevista semiestruturada com os especialistas na área.

O método de análise dos questionários aconteceu através da Análise de Conteúdo, procedimento desenvolvido por Laurence Bardin (1977), no qual é dividido em três etapas principais: pré-análise, análise e interpretação, cujos detalhes estão expostos no item 3.2. Na pré-análise, os objetivos da pesquisa foram definidos, o material foi organizado e coletado, e foi realizada uma leitura exploratória para familiarizar-se com o conteúdo. Na etapa seguinte, a análise, todo conteúdo foi selecionado e codificado de acordo com suas categorias e por fim, na etapa de interpretação foi feita uma relação entre as categorias e suas possíveis interpretações.

### **3.1- Cuidados éticos que foram tomados ao realizar a pesquisa**

Diante do exposto e de posse dos questionários embasados na pesquisa, o projeto foi encaminhado ao Comitê de Ética em Pesquisa (CEP) do IFSertãoPE via

Plataforma Brasil, no qual foi validado e assim, foi possível dar continuidade às etapas da presente pesquisa. Os documentos enviados ao CEP estão disponíveis e anexos a este trabalho: Termo de Assentimento Livre e Esclarecido (TALE) para estudantes menores de idade (Apêndice H), Termo de Consentimento Livre e Esclarecido (TCLE) para maiores de 18 anos (Apêndice I), Termo de Consentimento Livre e Esclarecido (TCLE) para pais, responsáveis e docentes (Apêndice J). Também foi realizado um seminário com os estudantes, pais e responsáveis e gestão da instituição, campo da pesquisa, aos quais foram esclarecidas a proposta e a relevância do trabalho.

### **3.2 - Coleta de dados**

A coleta de dados desta pesquisa serviu para mensurar e entender os comportamentos dos discentes com o uso do Pensamento Computacional como alternativa ao ensino de Lógica de Programação. De início foi aplicado um questionário (Apêndice B) para verificar o nível dos estudantes em relação à metodologia tradicional de Lógica de Programação, cuja ideia foi entender a percepção dos estudantes quanto ao método tradicional de ensino.

Em seguida foi ministrado um minicurso para os discentes de como resolver problemas utilizando o PC, logo após, iniciou-se a etapa de coleta de dados, que ocorreu em duas fases: Primeiro, um instrumento de coleta, um questionário (apêndice B) respondido pelos discentes, no qual eles deveriam responder suas perspectivas a respeito da metodologia estudada. Na maioria dos questionários foi utilizado a escala *Likert* de cinco posições possíveis, indicando: Concordo, Discordo, Neutro, Discordo Fortemente e Concordo Plenamente. Nos demais, apenas sim ou não como resposta.

Todos os questionários foram criados no *Google Forms* e respondidos com questões objetivas, visando à avaliação do método Pensamento Computacional por parte de cada discente participante da pesquisa, a fim de verificar suas percepções sobre a nova metodologia, na tentativa de identificar se o desempenho e a cognição dos estudantes havia melhorado, fator importantíssimo para a aprendizagem. O questionário utilizado foi composto por 12 questões, sendo todas objetivas, as quais buscavam avaliar a apreensão da nova estratégia estudada e se seria possível implementá-la dentro do currículo da disciplina de Lógica de Programação como algo fixo.

Cabe ressaltar, que de acordo com Rattray e Jones (2007), um instrumento válido, seja questionário ou formulário, deve ser simples, viável e aceitável para os usuários e pesquisadores. Os autores ainda afirmam que deve ser confiável e preciso (confiabilidade); ser apropriado ao problema de pesquisa (validade de conteúdo); refletir a teoria subjacente ao fenômeno ou conceito a ser medido (validade de construto); e ser capaz de medir as mudanças, tanto em indivíduos diferentes quanto nas respostas de um único indivíduo ao longo do tempo (sensibilidade a alterações). Calibrar questionários em qualquer pesquisa científica é essencial para garantir que as respostas coletadas sejam válidas. Desse modo, os questionários apresentados no Apêndice A e B foram baseados nos trabalhos de Melo (2020) e Machado (2020).

### **3.3- A construção do produto educacional**

Produto educacional é um recurso, material ou ferramenta, projetado e desenvolvido com o objetivo principal de facilitar a aprendizagem e o ensino. Projetados para atenderem necessidades específicas no contexto educacional, esses produtos podem abranger uma ampla variedade de formatos, como softwares e jogos educativos, aplicativos móveis, livros didáticos, vídeos, cursos online, entre outros.

O produto em questão trata-se de um livro digital, como mencionado anteriormente, um guia de boas práticas sobre o PC, utilizando como forma de exercício funcional, a ferramenta de programação em blocos, o *Scratch*. O mesmo foi desenvolvido de acordo com o que o autor desta pesquisa, dado a sua experiência em sala de aula, verificou diante dos tópicos mais relevantes de programação, nos quais os estudantes apresentavam dificuldades. Vale frisar que este, atua na Educação Profissional e Tecnológica há mais de 10 anos na rede estadual do Ceará, especificamente, nas disciplinas de Lógica de Programação.

## **4 ANÁLISE DOS DADOS**

A interpretação dos resultados deu-se a partir da Análise de Conteúdo proposta por Bardin (1977), e através desta, os dados foram coletados e preparados. O primeiro passo foi transcrever todas as respostas dos formulários do *Google* em um documento; a segunda etapa contemplou a codificação, na qual o autor deste

trabalho realizou uma leitura cuidadosa de todas as respostas, para então, desenvolver uma compreensão inicial do conteúdo, identificando as unidades de análise relevantes, palavras, textos e frases.

A etapa seguinte foi a criação de um sistema de codificação, separando por categorias de respostas, e na sequência, avaliando-o. Cada resposta analisada é atribuída a uma ou mais categorias, de acordo com seu conteúdo. Uma vez que todas as respostas foram codificadas, o pesquisador realizou a tabulação dos dados via planilha eletrônica para determinar a frequência de cada categoria. Dispostos em listas, os dados foram analisados. A interpretação envolveu a comparação de diferentes grupos de respondentes e a análise de relações entre categorias. Os resultados foram compilados e dispostos no decorrer desta seção.

Através das intervenções realizadas com o *Scratch*, observou-se um aumento do interesse e da participação dos estudantes na resolução de problemas, contribuindo para o desenvolvimento do Pensamento Computacional e a construção do conhecimento. A análise de estudos mostrou uma lacuna na abordagem de tópicos de programação utilizando o *Scratch* e o PC em cursos técnicos de Informática, ressaltando a importância de integrar teoria e prática para resolver problemas computacionais.

A fim de continuar o desenvolvimento do tema, as próximas seções apresentam os resultados da Revisão Sistemática de Literatura (RSL), (Apêndices M e N) e como se deu a análise dos dados

#### **4.1 - Resultados das Revisões Sistemáticas de Literatura**

Os resultados apresentados na RSL (Anexos M e N) de acordo com Júnior e De Oliveira (2022) indicaram que o ambiente *Scratch* é amplamente difundido na esfera acadêmica e, predominantemente, produz resultados favoráveis.

A integração de jogos digitais como um complemento ao processo educacional também é identificada como um estímulo para o desafio e recompensa. No entanto, a computação desplugada se apresenta como uma abordagem viável, focada em ensinar os fundamentos da computação, sem a necessidade de dispositivos computacionais, resultando em um método de ensino de custo acessível.

A análise abrangeu 20 artigos, com o intuito de fornecer uma avaliação crítica da área de pesquisa e delinear as principais linhas pesquisadas, além de resultados



de estudiosos no contexto nacional. A segunda RSL foi realizada apenas em bases de dados estrangeiras, trazendo trabalhos, preponderantemente, em Inglês.

Essas revisões abrangeram 20 textos (Anexo M) e 16 artigos (Anexo N), sendo o primeiro em bases de dados nacionais (*Google Acadêmico* e *SBC Open Lib*), escritos em Língua Portuguesa e o segundo em bases de dados estrangeiras, grafados em Língua Inglesa (*Scopus*, *Science Direct*, *Google Scholar* e *SBC Open Lib*). Assim, foi permitido identificar o que está sendo estudado no Brasil e no estrangeiro sobre metodologias de ensino utilizando o PC.

Alguns dos autores estudados nas revisões citadas, tais como POLONI; SOARES; WEBBER, (2019) identificaram que ao utilizar o ambiente *Scratch*, as possibilidades de aprendizagem são expandidas, porém, o professor também deve atuar como mediador, desenvolvendo estratégias e intervenções para auxiliar os estudantes. Vieira e Sabatini (2021) argumentam que a partir da intervenção da ferramenta *Scratch*, os discentes desenvolveram interesse e curiosidade na resolução de problemas, tornando-se mais participativos. MASSA (2019), em sua revisão de literatura, realizou um mapeamento de trabalhos publicados entre 2012 e 2017, no qual identificou que o estudante é construtor do seu próprio conhecimento, desenvolvendo um trabalho cooperativo por meio do uso do pensamento computacional e das tecnologias de construção.

As pesquisas publicadas em Inglês na RSL, em sua maioria, diferem dos nacionais, a exemplo de JING, Sifeng *et al* (2020), nos quais os autores propõem um modelo que leva em consideração não apenas os processos de pensamento e aprendizado do estudante (aspectos cognitivos), mas também sua capacidade de monitorar, avaliar e regular seu próprio aprendizado (aspectos metacognitivos) por meio da utilização do *Scratch*.

É verificada uma preocupação maior não só com a metodologia, mas também com os aspectos cognitivos envolvidos. GEROSA, Anaclara *et al* (2021), investigaram as habilidades cognitivas e o pensamento computacional em crianças de 5 anos, buscando evidências de associações entre essas habilidades e a capacidade de sequenciamento e comparação de números simbólicos. Diante das razões expostas a partir das observações das dificuldades enfrentadas pelos discentes e com base em 7 anos de experiência em docência do autor deste

trabalho, verificou-se a necessidade de uma reflexão a respeito do ensino de Ciência da Computação, principalmente em Lógica de Programação.

Nesse sentido, o uso do *Scratch* apropriando-se do Pensamento Computacional se mostra fundamental para que a escola possa criar espaços de desenvolvimento, despertando no estudante o interesse em "criar" a partir do "fazer". Ao analisar as práticas desenvolvidas nos trabalhos estudados nas revisões de literatura, percebeu-se que nenhuma trabalhou de fato a utilização de tópicos de programação por meio do *Scratch* e do PC em curso técnico em Informática, aliando a teoria com a prática a partir da resolução de problemas computacionais.

#### **4.2 - A ferramenta *Scratch***

A introdução do *Scratch* como ferramenta no ensino de Lógica de Programação é um marco significativo. Ao usar essa plataforma, os estudantes podem ser imersos em um ambiente visual e interativo, proporcionando uma abordagem *hands-on* para compreender noções complexas de programação. É como se o *Scratch* fosse uma ponte que conecta a teoria à prática. Através dessa plataforma, os alunos conseguem traduzir conceitos abstratos em projetos tangíveis. Essa experiência prática não apenas fortalece o entendimento da lógica por trás dos algoritmos, mas também estimula a criatividade e a resolução de problemas.

Um fator importante é a aceitação ou rejeição dos discentes à metodologia alternativa para o ensino de Lógica de Programação, utilizando o Pensamento Computacional e enfatizando a legitimidade da sistemática proposta como algo capaz de melhorar o desempenho dos estudantes e a construção do conhecimento conceitual, aspectos processuais e atitudinais da programação de computadores e Lógica de Programação.

Antes de descrever a forma como ocorreu a aplicação dos primeiros problemas, cabe frisar que ao primeiro contato com a ferramenta *Scratch*, os estudantes se mostraram bastante interessados por conta de toda a experiência visual e amigável que a ferramenta trazia. A realização da intervenção aconteceu em duas etapas, e a primeira constituiu-se em realizar uma entrevista por meio de um questionário para entender um pouco mais sobre a visão dos estudantes em relação à abstração e à solução de problemas através da metodologia tradicional.

#### **4.3 Análise dos resultados**

O quadro 1, a seguir, foi retirado do anexo B que trata das competências básicas e necessárias para o entendimento de problemas baseados nas habilidades

do Pensamento Computacional e seus 4 pilares.

Quadro 1 - Descritores de competências

Fonte:

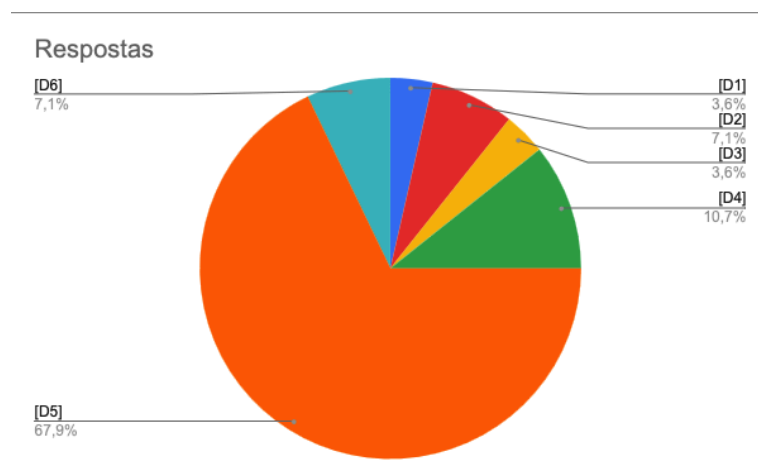
Pesquisa  
direta

Após  
a coleta das  
respostas,  
as mesmas  
foram

Descritor	Competência
D1	Dar sentido aos dados, encontrar padrões e obter conclusões a partir observação de dados (análise de dados).
D2	Coletar informações adequadas e selecionar informações relevantes para o contexto da atividade desenvolvida (coleta de dados).
D3	Organizar e descrever dados em representações apropriadas, como gráficos, palavras, frases, imagens, tabelas, etc, (representação de dados).
D4	Dividir tarefas grandes em partes menores e mais fáceis de gerenciar resolver (decomposição).
D5	Planejar e organizar sequências de passos para resolver um problema ou alcançar algum objetivo.
D6	Reduzir a complexidade para criar ideias, encontrar características básicas e gerar modelos (abstração).

apresentadas no gráfico abaixo:

**Gráfico 1 - Avaliação dos descritores**



Fonte: Pesquisa direta

Ao observar as afirmações dos estudantes quanto aos descritores da metodologia tradicional, percebe-se alguns vieses interessantes. A partir do gráfico é

possível analisar, de acordo com o D1, que apenas 3,6%, ou seja, apenas 1 estudante consegue compreender e dar sentido aos dados através de sua interpretação para encontrar padrões. É verificado que os demais não conseguem perceber essa competência, de acordo com a estratégia que vem sendo aplicada.

Em D2, 7,1% (um total de 2 estudantes) conseguiram enxergar na metodologia tradicional uma abertura para a compreensão de coleta de informações para entendimento de um contexto. Isto é, a coleta de dados, um saber de muita importância para a resolução de qualquer problema algorítmico.

D3 diz respeito à representação de dados, e de como estes serão expostos. Pois a explanação adequada dos mesmos ajuda os programadores a entenderem melhor as informações com as quais estão lidando. Seja através de variáveis, estruturas de dados ou tipos de dados complexos, uma exposição clara e precisa é essencial para compreender o que está sendo manipulado, porém, apenas um estudante consegue compreender a partir da metodologia tradicional.

O descritor D4 discute a decomposição, referindo-se à prática de dividir conjuntos complexos de dados em partes menores e mais gerenciáveis, os quais podem ser difíceis de entender e manipular. Ao decompor esses conjuntos em partículas menores e mais simples, torna-se mais fácil gerenciar e trabalhar com esses dados.

A representação foi de 10,7%, que implica em 3 estudantes com a capacidade de compreender o processo de decomposição. O descritor mais significativo foi o D5, no qual 19 estudantes, um total de 67,9% dos entrevistados, afirmaram conseguir compreender o mínimo que é planejar a sequência básica de passos para resolver um problema, por exemplo: os comandos necessários para fritar um ovo. Foi observado que essa habilidade está diretamente associada a situações do cotidiano.

Por fim, o D6, a abstração, na qual 2 estudantes se conectaram através da metodologia tradicional. Foi verificado, no entanto, com base no D5, o qual obteve maior avaliação positiva, que de fato, os outros descritores, os quais servem como fundamento para compreensão e solução de problemas, não estão sendo trabalhados para tornar possível a associação com outras áreas.

A partir da análise dos resultados do D5 é possível inferir que os estudantes tentam resolver os pequenos problemas sem uma estratégia definida, ou sem um pensamento que lhes dê um norte, por desconhecimento dos fundamentos ou

pilares do PC. Desse modo, em seguida foi dado início ao minicurso de Fundamentos do PC através da ferramenta *Scratch*.

Os descritores da educação referem-se a declarações ou padrões específicos, os quais descrevem o que os estudantes devem ser capazes de fazer ou saber em determinado estágio de sua aprendizagem. Os citados no quadro 1 são usados para descrever metas claras e mensuráveis quanto ao ensino e à avaliação, a partir das competências atribuídas ao Pensamento Computacional. Desse modo, as subseções a seguir apresentam a interpretação dos descritores à luz das práticas realizadas durante a intervenção com os estudantes.

#### **4.4 Descrição e análise da intervenção**

O desenvolvimento das atividades planejadas aconteceu, especificamente, em três momentos: exploração do ambiente *Scratch*, associação com os conceitos de Pensamento Computacional e desenvolvimento das soluções pelos estudantes. Antes de expor a forma como ocorreu a aplicação e o desenvolvimento da intervenção, cabe ressaltar a empolgação dos discentes na primeira aula sobre *Scratch*, algo totalmente diferente do que eles já vivenciaram, pois, a partir de então, a programação passou a ser visual.

Para a aplicação do minicurso referente ao Pensamento Computacional com a utilização de *Scratch*, em seus fundamentos foi utilizado o produto educacional desenvolvido durante este trabalho, que se trata de um manual digital (*ebook*). O mesmo é constituído de boas práticas de Programação, utilizando o Pensamento Computacional como estratégia metodológica, além dos conteúdos básicos e fundamentais propostos no componente curricular de Lógica de Programação. O minicurso ofertado teve duração de 4 encontros com 4 aulas cada, totalizando 16 horas/aula, todos realizados nas sextas-feiras, nos últimos horários da tarde.

Estas aulas foram voltadas para análise e interpretação de problemas que envolviam algoritmos, vídeos, leituras de artigos, resolução de exercícios e avaliação da metodologia. Para o desenvolvimento dos problemas propostos, o grupo foi dividido em equipes de 5 estudantes, simulando uma fábrica de software, conforme a figura 4 com o registro da aplicação do minicurso.

Figura 4 - Aplicação do minicurso



Fonte: Pesquisa direta

O quadro 2 a seguir apresenta o cronograma de execução das atividades utilizadas em cada um dos problemas.

**Quadro 2** - Cronograma de execução de atividades

Número	Título/Problema	Conteúdos Necessários	Tempo para realização da prática
0001	Introdução ao Pensamento Computacional	Apresentação, contexto histórico e pilares do Pensamento Computacional	60 minutos (1h/a)
0002	Introdução à ferramenta <i>Scratch</i>	Apresentação da ferramenta	60 minutos (1h/a)
0003	Pensamento Computacional e <i>Scratch</i> : apresentação do guia de boas práticas	Apresentação dos tópicos de programação a partir da representação da ferramenta <i>Scratch</i> utilizando como guia o produto educacional.	120 minutos (2h/a)
0004 -	Construção de um Sistema para cálculo de média com a opção de informar se o aluno foi aprovado, reprovado ou está em recuperação	Algoritmo, elementos utilizados para construção de um algoritmo, tipos de algoritmo: pseudocódigo (portugol), fluxograma e descrição narrativa.	120 minutos (2h/a)
0005	Construção de uma Calculadora de Índice de Massa Corporal (IMC)	Variáveis, constantes e tipos de dados.	120 minutos (2h/a)

0006	Sistema para prescrição de medicamentos	Estrutura sequencial, estrutura de seleção simples, composta e encadeada, múltiplas escolhas e estrutura de repetição, <i>for</i> , <i>while</i> e <i>do-while</i> .	120 minutos (2h/a)
0007	Sistema de cadastro de Livros para Bibliotecas	Vetores, Matrizes, procedimentos, escopo de variáveis, funções e parâmetros.	120 minutos (2h/a)
0008	Urna eletrônica para votação do grêmio estudantil	Vetores, Matrizes, procedimentos, escopo de variáveis, funções e parâmetros.	120 minutos (2h/a)

Fonte: Pesquisa direta

Como já afirmado anteriormente, os problemas utilizados foram concebidos pelo professor após ministrar o minicurso com as práticas do Pensamento Computacional e apresentar a ferramenta *Scratch*. Durante o mesmo foi utilizado como exemplo o produto educacional resultante deste trabalho, o livro digital intitulado: "Pensamento Computacional & *Scratch*: um guia de boas práticas em Programação", o qual condensa um material teórico que relaciona e fundamenta o Pensamento Computacional através do *Scratch* para resolver problemas de nível básico, intermediário e avançado. Porém, o professor responsável pela pesquisa utilizou problemas que se adequaram mais à realidade dos discentes, usados como instrumento de intervenção, disponíveis como Apêndice.

No primeiro problema, denominado 'Sistema de cálculo de notas', os estudantes, em grupo, demonstraram grande atenção na resolução, revisando cuidadosamente cada movimento executado e a solução concebida. Ao perceber a possibilidade de obter um resultado diferente, eles optaram por uma nova abordagem, refazendo o exercício. Ao serem questionados sobre a estratégia utilizada, explicaram que o início deu-se a partir da análise da situação deles enquanto estudantes, ou seja, o sistema deveria apresentar as características de uso, de modo que atendessem aos professores e aos estudantes, coletando as informações (notas) via teclado e realizando o cálculo, informando o resultado final e os critérios de aprovação, reprovação ou exame.

Ao abordar a solução do desafio, os grupos afirmaram: "Começamos do mais simples ao mais complexo. Inicialmente, dedicamos um tempo para refletir e buscar

uma abordagem mais fácil para começar o processo". É interessante destacar que o problema citado faz parte do cotidiano dos estudantes, já que bimestralmente eles são submetidos a avaliações e cálculos de média, proporcionando um contexto em que possam aplicar pilares do PC para construir a solução: decomposição, algoritmos, análise e abstração.

No segundo problema, denominado Sistema de Calculadora de Índice de Massa Corporal, em geral, os participantes conseguiram associar ao primeiro, resolvido anteriormente, pois verificaram que existia uma relação lógica entre as situações apresentadas, no tocante à entrada de valores, processamento do cálculo e a solução com possíveis critérios de decisão. Afirmaram conseguir estruturar o pensamento com maior facilidade para relacionar ambos e elaborar a solução. Informaram também que perceberam a habilidade de analisar o problema, contexto e situação, para em seguida raciocinar e montar o desenvolvimento da solução. A resposta dada faz alusão aos elementos do PC, tais como análise e abstração, por exemplo.

O problema 3 é o sistema de cadastro de livros da biblioteca. Neste, os estudantes consideraram uma situação desafio do dia a dia, já que, ao frequentarem este ambiente da escola passavam por um processo manual de busca e reserva de livros. Situações cotidianas, segundo eles, são mais fáceis de identificar. Assim, eles observaram o que estava em volta, verificando o problema e o que precisava, de fato, ser feito.

Por fim, o problema 4, foi considerado por eles o mais complexo, porque encontraram uma certa dificuldade para analisar e traçar uma estratégia inicial. Os grupos se uniram para discutir a solução e chegarem a um consenso, pois se tratava de um sistema de votações, no qual demandava segurança para armazenamento e apresentação dos dados, legitimidade no processo, tudo de forma ordenada e coesa, a fim de eleger uma chapa vencedora.

Essa narrativa evidencia os princípios do Pensamento Computacional (PC), uma vez que envolveu processos de análise, abstração e automação para resolver o problema proposto. Isso fica claro na resposta à pergunta seguinte: Vocês acreditam que essa mesma abordagem poderia ser aplicada para resolver outros problemas? "Sim, poderia. Acreditamos que todos os problemas solucionados anteriormente, porque é mais eficiente analisar e tentar resolver rapidamente várias partes do problema do que o problema como um todo". Nesse contexto, os participantes



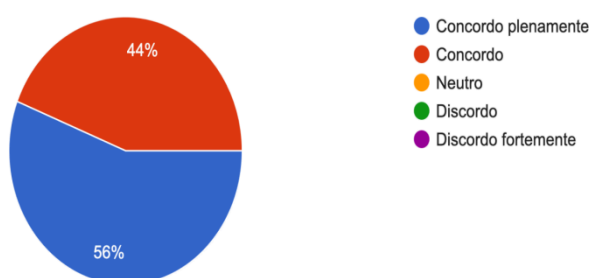
também destacaram a decomposição como um elemento que facilita a resolução de problemas.

Após a conclusão do minicurso e o desenvolvimento das soluções propostas pelos estudantes, foi aplicado um questionário com os 25 participantes para verificar se, realmente, o PC atende como uma metodologia alternativa para o ensino de Lógica de Programação.

O gráfico 1 apresenta o primeiro questionamento feito aos participantes em relação ao estímulo para o aprendizado no desenvolvimento de atividades, assim buscando entender como isso pode contribuir para a solução de problemas, utilizando o Pensamento Computacional.

**Gráfico 1-** Busca de conteúdo a partir da metodologia de Pensamento Computacional

Utilizando a metodologia de pensamento computacional, você é estimulado e desafiado a buscar aprendizado a partir do conteúdo que está aprendendo ?  
25 respostas



Fonte: Pesquisa direta

Em unanimidade, como mostra o gráfico, todos os estudantes afirmaram que concorda e/ou concordam plenamente, pois ao incorporar o Pensamento Computacional como uma ferramenta para o estímulo ao desafio na resolução de problemas, cria-se um ambiente que fomenta a curiosidade, a persistência e a capacidade de enfrentar obstáculos de maneira estratégica.

Essa mentalidade é valiosa não apenas no campo da tecnologia, mas em todas as áreas da vida, contribuindo para a formação de indivíduos resilientes e solucionadores de problemas. O gráfico 2, a seguir, faz o questionamento aos estudantes a respeito dos tópicos e objetivos da Lógica de Programação. Essa abordagem não se limita apenas à programação ou à tecnologia. Pode ser

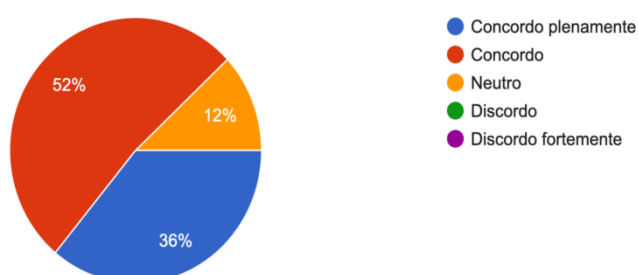
incorporada em várias disciplinas e contextos, incentivando uma atuação estruturada e lógica para a resolução de problemas. Porém, foi verificada uma maior adesão à área, pois é o cerne de seu desenvolvimento.

**Gráfico 2** - Objetivos da disciplina de programação

Fonte: Pesquisa direta

Os objetivos (conhecimentos, habilidades e atitudes) da disciplina de lógica do programação foram alcançados?

25 respostas

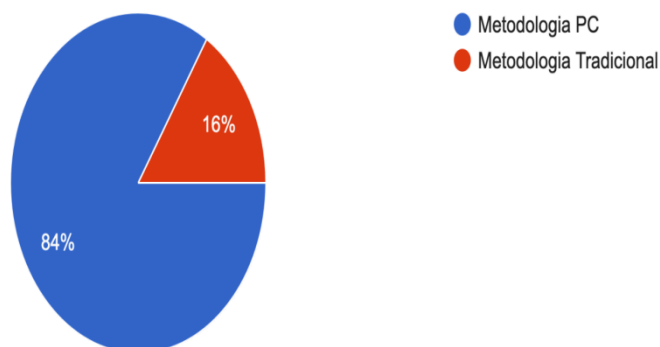


O público entrevistado, equivalente a 12% (3 estudantes), afirmaram neutralidade, não conseguindo associar todos os tópicos da metodologia dentro do Pensamento Computacional. Desse modo, o gráfico 3 a seguir faz o questionamento a respeito desse ponto em questão:

**Gráfico 3** - Tópicos de aprendizagem

Como você acha que pode aprender mais sobre esses tópicos?

25 respostas

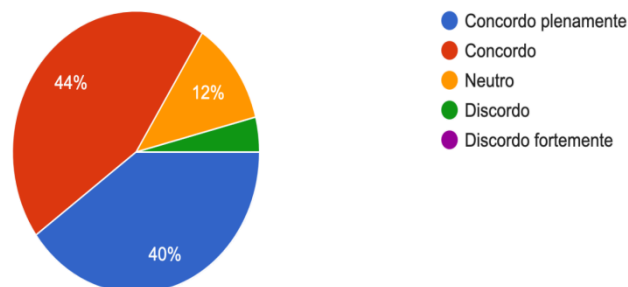


Fonte: Pesquisa direta

A escolha entre Pensamento Computacional e Metodologia Tradicional para aprender Lógica de Programação depende de vários fatores. Porém, os participantes da pesquisa foram submetidos a um processo formativo, no qual tiveram contato na teoria e na prática com as referidas metodologias. Entretanto, 16% (4 estudantes) ainda afirmaram que a metodologia tradicional é a ideal para este tipo de conhecimento, devendo ser incluída ao estilo de aprendizado do indivíduo, seus objetivos e a abordagem pedagógica preferida. No processo formativo, deve-se levar em consideração também a forma como os conteúdos são abordados e trabalhados. O gráfico 4, abaixo, questiona se caso a metodologia de Pensamento Computacional fosse aplicada de forma integral no curso, teria efeito positivo no desenvolvimento do discente.

**Gráfico 4-** Efeito da metodologia de Pensamento Computacional

Você acredita que a metodologia PC teve um efeito positivo na sua nota no curso de programação lógica?  
25 respostas



Fonte: Pesquisa direta

Verificou-se, afirmativamente para a grande maioria, pois a aplicação do Pensamento Computacional dentro da metodologia tradicional de ensino de Lógica de Programação pode ter efeitos positivos e complementares. Porém, 12% declararam estarem neutros, o que colabora na conclusão de que ambas as abordagens podem se beneficiar, mutuamente, proporcionando uma experiência de aprendizado mais rica e abrangente. Combinar a estrutura e a sequência didática da metodologia tradicional com a abordagem prática e orientada a problemas do

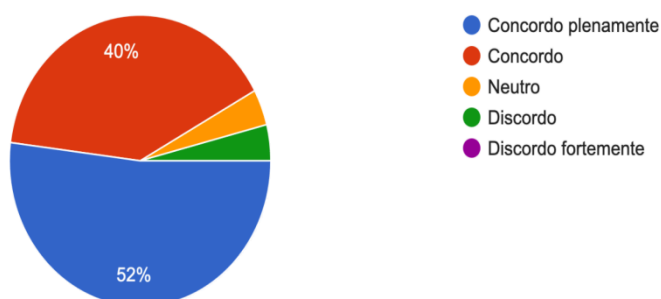
Pensamento Computacional, pode criar uma experiência de aprendizado mais holística e eficaz na Lógica de Programação.

O gráfico 5, a seguir, apresenta o questionamento sobre a adaptação de todas as disciplinas do curso técnico para a metodologia de PC.

**Gráfico 5** - Implementação da metodologia de Pensamento Computacional

Você gostaria que as disciplinas do curso técnico utilizassem a metodologia PC?

25 respostas



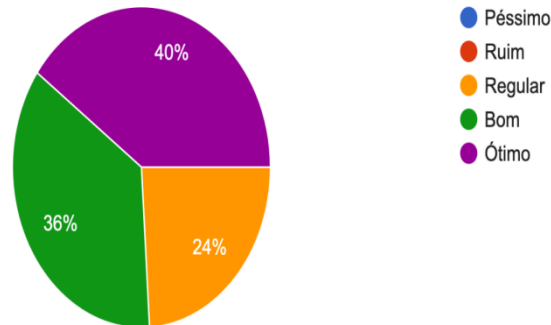
Fonte: Pesquisa direta

Constatou-se que houve 92% de aceitação entre os que concordam plenamente e os que concordam. Foi verificado que a adaptação pode ser realizada, porém existe o grupo de 8%, equivalente a 2 estudantes que não acreditam. Quando se fala em disciplinas do curso de Informática, o questionamento se torna mais abrangente, pois estão incluídas diversas matérias da base técnica, tais como: Banco de Dados, Redes de Computadores, Design, etc. Sendo assim, o gráfico 6, a seguir, faz outro questionamento, especificamente relacionado à disciplina de Lógica de Programação.

**Gráfico 6** - Validação da metodologia de PC na disciplina de LP

Qual a sua validação quanto a utilização da metodologia de PC na disciplina de Lógica de Programação?

25 respostas



Fonte: Pesquisa direta

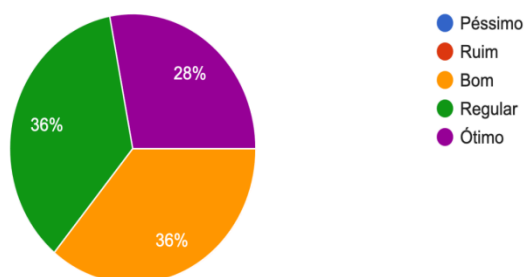
O Pensamento Computacional é muitas vezes considerado mais abrangente, abordando não apenas a codificação, mas também a formulação de problemas, a decomposição de tarefas complexas, o reconhecimento de padrões e a abstração. Desse modo, a Lógica de Programação pode ser vista como uma subcategoria deste, focando, principalmente, nas habilidades específicas necessárias para escrever código eficiente.

O Pensamento Computacional, porém, pode ser mais fácil de contextualizar e tornar relevante em situações do mundo real, pois se concentra em habilidades transferíveis para várias disciplinas e profissões. De acordo com o gráfico 6, 24% dos entrevistados validaram que a implementação de uma metodologia alternativa utilizando o Pensamento Computacional ainda se mostra regular, o que reforça a necessidade primária de implantar uma metodologia híbrida. O gráfico 7, a seguir, apresenta a opinião dos discentes quanto ao desempenho na disciplina de LP utilizando a metodologia de PC.

**Gráfico 7-** Desempenho na disciplina de LP

Qual a sua opinião sobre seu desempenho na disciplina de lógica de programação utilizando a metodologia de PC?

25 respostas



Fonte: Pesquisa direta

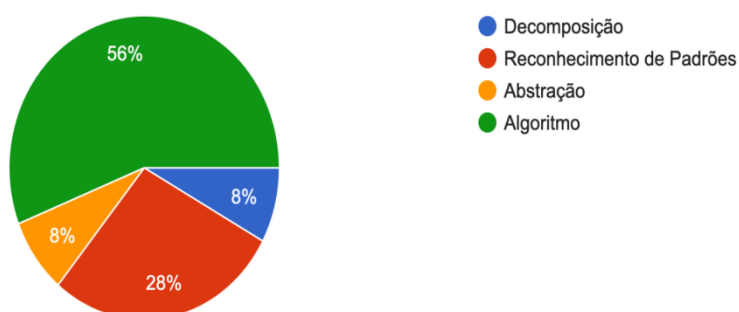
A avaliação do Pensamento Computacional pode se concentrar em diversas competências, incluindo resolução de problemas, pensamento algorítmico e comunicação. Já a avaliação na Lógica de Programação pode ser mais voltada para a precisão na codificação, capaz de não refletir completamente as habilidades abrangentes necessárias para o Pensamento Computacional. A depender de como a disciplina é conduzida, os resultados podem ser diferentes.

O gráfico apresentado é um cenário de equilíbrio, pois 36% confirmaram terem tido um desempenho regular, enquanto 36% um bom desempenho. Logo, aponta para validar a metodologia do Pensamento Computacional como uma alternativa ao ensino de Lógica de Programação. É crucial adaptar continuamente os métodos de ensino com base nos resultados obtidos. A flexibilidade e a adaptabilidade são fundamentais na constante evolução da educação em Ciência da Computação. Por fim, o gráfico 8, em sequência, levanta um questionamento sobre os 4 pilares do pensamento computacional.

#### Gráfico 8 - Contribuição dos pilares do PC

Enumere de 1 a 5 acerca dos pilares do PC que contribuiram para a sua aprendizagem

25 respostas



Fonte: Pesquisa direta

Ao fracionar um problema, os estudantes podem abordar cada componente de maneira independente, facilitando a compreensão e a resolução eficaz, em que reconhecer padrões é uma habilidade crucial. A identificação de regularidades permite antecipar resultados e simplificar a abordagem a desafios computacionais. Após analisar o perfil dos estudantes através do questionário sobre a aplicação da metodologia tradicional, verificou-se a dificuldade em identificar o que a maioria dos descritores contemplam sobre conhecimentos básicos, para abstrair e resolver problemas algorítmicos. Dessa maneira, foi aplicado um minicurso como citado anteriormente, a fim de apresentar aos estudantes a metodologia alternativa que utiliza o Pensamento Computacional e o *Scratch* como veículos de aprendizagem.

#### **4.5 Entrevista com Especialistas**

A entrevista semiestruturada (Apêndice C) com especialistas é um método de coleta de dados combinando elementos de estruturação e flexibilidade. Nesse tipo de entrevista, o pesquisador em questão, preparou um conjunto de perguntas pré-determinadas que abrangem o Pensamento Computacional (PC). Foi ministrado uma formação aos docentes sobre o PC e seus pilares, sequencialmente, decidiu-se utilizar a abordagem de entrevista semiestruturada, na qual o entrevistado teve a oportunidade de compartilhar suas experiências, centrando-se nos principais temas delineados pelo pesquisador.

Essa escolha permite tanto respostas espontâneas e livres por parte do entrevistado quanto destaca a habilidade do entrevistador. As perguntas desenvolvidas para a entrevista foram elaboradas considerando o embasamento teórico da pesquisa e as informações previamente coletadas sobre o fenômeno social. (Triviños, 1987).

Foram selecionados três docentes experientes na Educação Profissional e Tecnológica que atuavam em cursos técnicos de nível médio em Informática. Todos os profissionais entrevistados contavam com mais de 10 anos de exercício na carreira de forma ininterrupta e estavam familiarizados com a metodologia tradicional.

O quadro 5, a seguir, apresenta o questionário tabelado, a sequência apresentada em pergunta e as respectivas respostas dos docentes, aqui chamados de A, B e C.

Quadro 3- Entrevista com os especialistas docentes

Pergunta	Docente A	Docente B	Docente C
Q001- O que você entende por Pensamento Computacional?	É uma forma de pensar baseada em como os computadores tomam decisões, por meio de algoritmos e programas.	A capacidade de relacionar conhecimentos da computação com atividades rotineiras, desenvolvendo assim o raciocínio lógico e o pensamento analítico.	Entendo que pensamento computacional e a forma que aprendemos a resolver problemas de forma eficiente igual os computadores resolvem.
Q002-Quais as principais dificuldades encontradas pelos estudantes ao estudar Lógica de Programação?	As principais dificuldades dos estudantes são: entendimento do problema (Os estudantes focam na linguagem e não na resolução do problema), abstração (Dificuldade de abstrair apenas o que importa), decomposição (quebra do problema em pequenas partes) e colocar os pensamentos em ordem na forma de algoritmo.	Resolução de problemas matemáticos, decomposição de problemas, raciocínio lógico, capacidade de resolução de problemas, autonomia, interpretação de texto, além de falta de infraestrutura tecnológica adequada em casa para estudo.	A maior dificuldade dos estudantes está nas operações, até mesmo, básicas da Matemática, e compreender como os computadores funcionam. E também está na pouca experiência de vida. Com isso, não sabem solucionar os problemas.
Q003-Você acredita que a metodologia de Pensamento Computacional pode vir a contribuir com o desenvolvimento do raciocínio lógico para a resolução de problemas computacionais?	Acredito. Essa metodologia faz uma engenharia reversa, onde ao invés de pensarmos diretamente no problema, pensamos em quais passo a passo o computador fez para resolver um problema. Nesse	Sim, melhorando a capacidade de abstração e resolução de problemas dos estudantes.	Sim. Acredito que muitas das dificuldades dos estudantes está em raciocínio dos problemas para uma futura solução. O estudante entendendo como os computadores resolvem os



	<p>processo, estamos trabalhando o entendimento do problema, abstração, decomposição e construção de algoritmos.</p>		<p>problemas, já irá ajudar a resolver os problemas.</p>
<p>Você acredita no Pensamento Computacional como alternativa ao ensino tradicional de Lógica de Programação?</p>	<p>Não. Vejo o Pensamento Computacional como um complemento ao tradicional ensino de Lógica de Programação.</p>	<p>Não acho que seja uma alternativa, mas um complemento ao ensino de lógica. Acredito que o uso do Pensamento Computacional deva ser feito em todas as disciplinas, de forma transversal, tendo em vista o mundo altamente tecnológico em que vivemos.</p>	<p>Sim. Acredito que umas das maiores dificuldades dos estudantes em programação se dar ao fato de não saber como os computadores funcionam. Com o conteúdo de Pensamento Computacional viria a ajudar no entendimento dos problemas.</p>
<p>Na sua opinião, o que faz o Pensamento Computacional ser um tipo de metodologia multidisciplinar?</p>	<p>É uma metodologia utilizada para a solução de problemas complexos ou no dia a dia, mas que podem ser resolvidos em etapas.</p>	<p>A sociedade moderna, altamente tecnológica, necessita de jovens que dominem ao máximo essas tecnologias e consigam resolver problemas complexos em todas as áreas do conhecimento. Os pilares do Pensamento Computacional proporcionam a esses estudantes o desenvolvimento de habilidades que os tornarão</p>	<p>O desenvolvimento tecnológico já é uma das atividades básicas para qualquer pessoa. No campo profissional é mais importante ainda. A metodologia computacional ajudaria os estudantes a se tornarem seres com maior capacidade de compreender o mundo tecnológico e a resolver problemas computacionais.</p>

		profissionais e cidadãos mais preparados para o mercado de trabalho e sociedade do futuro.	
--	--	--	--

Fonte: Pesquisa direta.

A questão 1 (Q001) é unânime para os três docentes, pois todos têm visões alinhadas sobre o Pensamento Computacional. Acreditam firmemente que essa metodologia desempenha um papel crucial no desenvolvimento do raciocínio lógico, especialmente, para resolver problemas computacionais. Ao integrar conceitos como decomposição, reconhecimento de padrões, abstração e algoritmos, essa abordagem não apenas capacita os estudantes a compreenderem melhor o funcionamento de sistemas de computação, mas também fortalece suas habilidades analíticas e a capacidade de enfrentar desafios lógicos em diversas áreas.

A questão 2 (Q002) aborda as dificuldades que são encontradas pelos estudantes no que diz respeito à Lógica de Programação, e para os docentes entrevistados, as principais dificuldades versam entre a assimilação inicial de conceitos abstratos, a compreensão da lógica de controle de fluxo, como *loops* e condicionais, e a transição para pensar de maneira algorítmica. Muitos estudantes também enfrentam desafios ao traduzir desafios do mundo real para soluções algorítmicas e ao dominar a sintaxe de uma linguagem de programação específica. A pouca habilidade na resolução de problemas práticos também pode ser uma barreira. Interessante também ressaltar a fala do docente C, quando afirma que a pouca idade e inexperiência também são possíveis fatores para a dificuldade no entendimento, e por consequência, na resolução de problemas.

É indagado aos especialistas na questão 3 (Q003), a possibilidade de a metodologia trabalhada ser, de fato, útil para a resolução de problemas. Os quais afirmaram que a metodologia de Pensamento Computacional é fundamental para o desenvolvimento do raciocínio lógico dos estudantes na resolução de problemas computacionais, oferecendo uma estrutura sólida. O Pensamento Computacional é uma ferramenta valiosa para preparar os estudantes não apenas como programadores proficientes, mas também como pensadores analíticos, capazes de abordar questões complexas de maneira sistemática.

A questão 4 trata de uma possível substituição da metodologia tradicional

pela alternativa, utilizando o Pensamento Computacional, no qual dois docentes afirmaram que não. Para estes, ainda seria mais eficiente combinar as duas, criando uma espécie de metodologia híbrida, integrando elementos da abordagem tradicional com a metodologia de Pensamento Computacional, propiciando uma experiência de aprendizado abrangente e equilibrada. Essa Metodologia Híbrida visa equilibrar o rigor conceitual da abordagem tradicional, com a capacidade de resolver problemas de maneira abstrata e algorítmica proporcionada pelo Pensamento Computacional, oportunizando uma formação mais abrangente e contextualizada aos estudantes.

#### **4.6 Recomendações para a aplicação do Produto Educacional no curso Técnico em Informática:**

O produto educacional, resultante da presente pesquisa, trata-se de um livro digital. “Um guia de boas práticas: Pensamento Computacional utilizando *Scratch*”, que explora o uso do pensamento computacional e da plataforma *Scratch* como ferramentas para aprender Lógica de Programação. A figura 5, a seguir, apresenta a capa do livro.

**Figura 5** - Capa do Livro:um guia de boas práticas: Pensamento Computacional utilizando *Scratch* .



Fonte: Pesquisa direta

O livro introduz os conceitos básicos do pensamento computacional e sua importância na resolução de problemas cotidianos. Em seguida, apresenta a plataforma *Scratch*, ensinando passo a passo como criar programas simples usando blocos gráficos, abordando desde conceitos básicos até tópicos avançados, além de oferecer exercícios práticos para aplicar os conhecimentos adquiridos.

Também inclui estudos de caso e exemplos reais para mostrar como o Pensamento Computacional pode ser aplicado em diferentes áreas. Escrito de forma clara e acessível, o livro é adequado para estudantes, professores e qualquer pessoa interessada em aprender Lógica de Programação. Com este guia prático, os leitores serão capazes de dominar os fundamentos da LP, usando o PC e o *Scratch*, preparando-se para desafios futuros no campo da tecnologia. O objetivo é promover o ensino de Lógica de Programação a partir do Pensamento Computacional em um material didático complementar de fácil linguagem e aderente ao contexto da EPT.

A avaliação do PE (Produto Educacional) deu-se a partir da aplicação dos seus fundamentos, durante a intervenção realizada com os estudantes, na qual os exercícios e conceitos apresentados ao longo da obra foram socializados com estes. A vantagem desse PE é trazer um material simples, de fácil compreensão, que apresenta as competências do Pensamento Computacional replicadas no contexto da Lógica de Programação. Desse modo, os estudantes podem ter uma percepção de como é possível implementar tais competências a partir de uma linguagem clara.

## **5 CONSIDERAÇÕES FINAIS**

Integrar o Pensamento Computacional ao progresso desse propósito é uma maneira de garantir a solidez do processo educativo, que busca promover a habilidade de aprender por meio do desenvolvimento de competências cognitivas fundamentais necessárias para lidar com os desafios da vida contemporânea.

Dessa forma, com o objetivo de melhorar o processo de ensino e aprendizagem da disciplina Lógica de Programação através da implementação de uma metodologia que utilizasse os pilares do Pensamento Computacional para os

discentes da Escola Estadual de Educação Profissional Irmã Ana Zélia da Fonseca, foi ministrado um minicurso de 16 horas/aula a 25 estudantes do curso técnico em Informática. No qual voltou-se para fundamentações teóricas e práticas das competências do Pensamento Computacional e como aplicá-las para resolução de problemas, em contrapartida da metodologia que já vinha sendo trabalhada, anteriormente.

Foi apresentado como livro texto e material de apoio para o minicurso, o produto educacional resultante deste trabalho (Apêndice O), intitulado: Um guia de boas práticas: Pensamento Computacional utilizando *Scratch*. Do outro lado do processo formativo, existem os docentes. E para três, de outras instituições de ensino, também foi ministrado um minicurso sobre as competências do Pensamento Computacional, para assim verificar também, pela visão destes, se seria viável a nova abordagem.

Com base na leitura dos dados quantitativos e qualitativos extraídos dos questionários e entrevistas com os discentes, ficou constatado que a metodologia utilizando Pensamento Computacional foi bem aceita, devido a todo apelo gráfico e intuitivo que a ferramenta *Scratch* traz em sua concepção. Porém, não é o suficiente para traduzir-se com bons resultados em termos práticos, pois foi observado pelo autor deste trabalho, que em algumas situações dos problemas apresentados as dificuldades persistem. Por exemplo, no problema 1, o cálculo de médias, alguns estudantes sentiram dificuldades bem básicas no seu desenvolvimento, na adaptação de uma metodologia para outra.

Os docentes não conheciam a prática e os fundamentos do Pensamento Computacional, porém consideraram uma metodologia bem aceita e propícia de ser trabalhada, não só nas disciplinas de Lógica de Programação, mas em todas as disciplinas do currículo dos cursos de tecnologia da Educação Profissional. Contudo, constataram a mesma necessidade dos estudantes.

Conclui-se, a partir das observações durante as práticas do minicurso aplicado aos discentes e docentes, e análise das respostas dos questionários com ambos os públicos, que no atual momento, não é possível a substituição total para uma metodologia, na qual empregue em sua totalidade o Pensamento Computacional. O ideal é mesclar entre ambas metodologias, combinando aspectos que utilizem exemplos estratégicos para ilustrar os conceitos teóricos, conectando a teoria à aplicação prática. Assim, introduzir gradualmente os princípios do

Pensamento Computacional, destacando a decomposição de problemas, a abstração de informações, o reconhecimento de padrões e a criação de algoritmos. Dessa maneira, mostrar como esses conceitos se inter-relacionam com os tópicos já abordados de maneira tradicional.

Por fim, a conclusão é que uma metodologia, a qual combine aspectos da Linguagem de Programação tradicional usando os fundamentos do PC, tem potencial para ser utilizada por docentes e discentes no processo de ensino aprendizagem de Lógica de Programação. E o produto educacional "Um guia de boas práticas: Pensamento Computacional utilizando *Scratch*" pode ser explorado como instrumento de auxílio. Assim, será possível desenvolver um ambiente gráfico (jogo, site ou aplicativo), combinado com o produto educacional resultante deste trabalho, e fornecer ao professor uma abordagem mais específica para trabalhar os pilares do PC; além do mais, buscar desenvolver um estudo mais aprofundado de como mensurar a complexidade algorítmica de problemas que envolvam o mesmo.

## REFERÊNCIAS

ALMEIDA, M. E. B.; VALENTE, J. A. Tecnologias e currículo: trajetórias convergentes ou divergentes? São Paulo: Paulus, 2011.

ALE, Moyosore; STURDEE, Miriam; RUBEGNI, Elisa. A systematic survey on embodied cognition: 11 years of research in child–computer interaction. **International Journal of Child-Computer Interaction**, v. 33, p. 100478, 2022.

ARAÚJO, Aline Cássia Silva; DE OLIVEIRA, Francisco Kelsen. Revisão Sistemática da Literatura sobre Tecnologias digitais de informação e comunicação de tradução do par linguístico Português Libras. **Revista Semiárido De Visu**, v. 9, n. 3, p. 286-299, 2021. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/305> . Acesso em: 18 set. de 2022

BARR, Valerie; STEPHENSON, Chris. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. **Acm Inroads**, v. 2, n. 1, p. 48-54, 2011.

BLIKSTEIN, Paulo. O pensamento computacional e a reinvenção do computador na educação. **Education & Courses**, v. 1, 2008.

BARDIN, Laurence. **Análisis de contenido**. Ediciones Akal, 1991.

BOMBASAR, James et al. Ferramentas para o ensino-aprendizagem do pensamento computacional: onde está Alan Turing?. In: **Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie)**. 2015. p. 81.

Bogdan, R., & Biklen, S. (1994). *Investigação qualitativa em educação: uma introdução à teoria e aos métodos*. Porto: Porto Editora.

CAMPOS, Fabrício Vieira; DE SOUZA, Paulo Henrique. POSSIBILIDADES DE USO DO SCRATCH NO DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL.

In: **Anais do CIET: EnPED: 2020-(Congresso Internacional de Educação e Tecnologias| Encontro de Pesquisadores em Educação a Distância)**. 2020.

CAVAZIN, Jucélia. Um Estudo de Caso envolvendo Pensamento Computacional e o uso do Scratch. 2020

CHEN, Xieling et al. Application and theory gaps during the rise of artificial intelligence in education. **Computers and Education: Artificial Intelligence**, v. 1, p. 100002, 2020.

COSTA, M. G. N. da .; OLIVEIRA, F. K. de. A prática docente de professores não licenciados no Ensino Médio Integrado à Educação Profissional e Tecnológica. **Revista Semiárido De Visu**, [S. l.], v. 8, n. 3, p. 512–523, 2020. DOI: 10.31416/rsdv.v8i3.36. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/36>. Acesso em: 01 jan. 2024.

DA SILVA VIEIRA, Sebastião; SABBATINI, Marcelo. Pensamento computacional através do Scratch numa perspectiva Maker. **REVISTA INTERSABERES**, v. 16, n. 37, p. 43-63, 2021.

DE OLIVEIRA, Milena et al. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. In: **Anais do XXII Workshop sobre Educação em Computação**. SBC, 2014. p. 239-248.

De Paula, B. H., Valente, J. A. e Burn, A. O uso de jogos digitais para o desenvolvimento do currículo para a Educação Computacional na Inglaterra. *Currículo sem Fronteiras*, v. 14, n. 3, p. 46-71.

Deitos, Roberto Antonio, Angela Mara de Barros Lara, and Isaura Monica Souza Zanardini. "Política de educação profissional no Brasil: aspectos socioeconômicos e ideológicos para a implantação do Pronatec." *Educação & Sociedade* 36 (2015): 985-1001.



ELIEZER, Pacheco, org., *Perspectivas da educação profissional técnica de nível médio. Proposta diretrizes curriculares nacionais* (São Paulo: Moderna, 2012).

Disponível em: <<http://reitoria.ifpr.edu.br/wp-content/uploads/2014/06/Perspectivas-da-EPT.pdf>> Acesso em: 12. set. 2022

APPOLINÁRIO, Fábio. Dicionário de metodologia científica: um guia para a produção do conhecimento científico. In: **Dicionário de metodologia científica: um guia para a produção do conhecimento científico**. 2012. p. 300-300

FURTADO, Vasco. *Tecnologia e Gestão da Informação na Segurança Pública*. Rio de Janeiro: Editora Garamond, 2002.

FLYNN, Rachel M. et al. A narrative review of methods used to examine digital gaming impacts on learning and cognition during middle childhood. **International Journal of Child-Computer Interaction**, v. 30, p. 100325, 2021.

Gaudêncio Frigotto, Maria Ciavatta. “Educação básica no Brasil na década de 1990: subordinação ativa e consentida à lógica do mercado”. *Educ. Soc.*, Campinas, vol. 24, n. 82, p. 93-130, abril 2003. Disponível em : <<http://www.scielo.br/pdf/es/v24n82/a05v24n82.pdf>> Acesso em: 08. set. 2022.

GOBBI, R.; SILVEIRA, S. R.; BERTOLINI, C.; BIGOLIN, N. M.; PARREIRA, F. J.; CUNHA, G. B.; MACEDO, R. T. Formação Docente: desenvolvimento do pensamento computacional nos anos finais do ensino fundamental. **Anais da 35.a JAI – Jornada Acadêmica Integrada da UFSM**. Santa Maria: UFSM, 2020.

GUGGEMOS, Josef; MOSER, Luca; SEUFERT, Sabine. Learners don't know best: Shedding light on the phenomenon of the K-12 MOOC in the context of information literacy. **Computers & Education**, v. 188, p. 104552, 2022.

JING, Sifeng et al. A learner model integrating cognitive and metacognitive and its application on scratch programming projects. **IFAC-PapersOnLine**, v. 53, n. 5, p. 644-649, 2020.

JÚNIOR, Paulo Antonio Pasqual; DE OLIVEIRA, Simone. Pensamento computacional: uma proposta de oficina para a formação de professores. **RENOTE**,

v. 17, n. 1, p. 62-71, 2019. Disponível em: <  
<https://seer.ufrgs.br/renote/article/view/95707/0>> Acesso em: 15 de ago. de 2022.

Johnson, R. B., & Onwuegbuzie, A. J. (2004). Mixed methods research: A research paradigm whose time has come. *Educational researcher*, 33(7), 14-26.

LAUPICHLER, Matthias Carl et al. Artificial intelligence literacy in higher and adult education: A scoping literature review. **Computers and Education: Artificial Intelligence**, p. 100101, 2022.

KETELE, J.; ROEGIERS, X. *Méthologie du recueil d'informations: fondements de méthodes d'observation de questionnaires, d'interviews et d'étude de documents*. 2. ed. Bruxelles: De Boeck Universisté, 1993.

KRUGEL, Diego Ricardo. *Uso do Scratch para o desenvolvimento do Pensamento Computacional*. 2021.

MADEIRA, C. **Introdução ao Pensamento Computacional com Scratch**. 2017. Disponível em: [http://ceur-ws.org/Vol-1877/CtrlE2017\\_MC\\_4.pdf](http://ceur-ws.org/Vol-1877/CtrlE2017_MC_4.pdf) Acesso em: 08. jul. 2020.

MARINHO, Anna Raquel da S. et al. Relato de Experiência Vivenciada no PIBID sobre a Utilização da Computação Desplugada, a Hora do Código e do Scratch no Ensino Médio. In: **Anais do XXVI Workshop sobre Educação em Computação**. SBC, 2018.

MASSA, Nayara Poliana et al. Mapeamento do Pensamento Computacional por meio da ferramenta Scratch no contexto educacional brasileiro: análise de publicações do Congresso Brasileiro de Informática na Educação entre 2012 e 2017. 2019.

MADEIRA, Charles. **Introdução ao Pensamento Computacional com Scratch**. In: **Anais do II Congresso sobre Tecnologias na Educação (CTRL+ E-2017)**. 2017. p. 725-730.

MARQUES, Kelly Cristina Mucio; CAMACHO, Reinaldo Rodrigues; ALCANTARA, Caio Cesar Violin de. Assessment of the Methodological Rigor of Case Studies in the Field of Management Accounting Published in Journals in Brazil. **Revista Contabilidade & Finanças**, [s.l.], v. 26, n. 67, p.27-42, abr. 2015. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/1808-057x201500280>. Disponível em: <[http://www.scielo.br/pdf/rcf/v26n67/pt\\_1519-7077-rcf-26-67-00027.pdf](http://www.scielo.br/pdf/rcf/v26n67/pt_1519-7077-rcf-26-67-00027.pdf)>. Acesso em: 09 fev. 2024.

MOHER, David et al. Preferred reporting items for systematic review and meta-analysis protocols (PRISMA-P) 2015 statement. **Systematic reviews**, v. 4, n. 1, p. 1-9, 2015.

MARTÍNEZ-COMESAÑA, Miguel et al. Impact of artificial intelligence on assessment methods in primary and secondary education: systematic literature review. **Revista de Psicodidáctica (English ed.)**, 2023.

MORETTI, Vinícius Fernandes. O pensamento computacional no ensino básico: potencialidades de desenvolvimento com o uso do Scratch. 2019

MOREIRA, João Padilha. O SCRATCH PODE MELHORAR A APRENDIZAGEM DA LÓGICA DE PROGRAMAÇÃO?. **REVISTA ACADÊMICA ALCIDES MAYA**, v. 2, n. 1, p. 1-82, 2020.

MOHAMMADI, Zahra; VALERO, Oscar. A new contribution to the fixed point theory in partial quasi-metric spaces and its applications to asymptotic complexity analysis of algorithms. **Topology and its Applications**, v. 203, p. 42-56, 2016.

NUNES, D. J. Ciência da Computação na Educação Básica. **ADUFRGS - Sindical**, 6. jun. 2011. Disponível em: <http://adufrgs.org.br/artigos/ciencia-da-computacao-na-eduacao-basica>. Acesso em: 21 set. de 2022

OLIVEIRA, F. K.; GOMES, A. S. Revisão sistemática da literatura. **Métodos e pesquisas em Educação**, v. 1, p. 164, 2016.

PAULA; B. H.; VALENTE; J. A.; BURN, A. O uso de jogos digitais para o desenvolvimento do currículo para a educação computacional na Inglaterra. *Currículo sem Fronteiras*, v. 14, n. 3, p. 46-71, set/dez, 2014. Disponível em: <<http://www.curriculosemfronteiras.org/vol14iss3articles/paula-valenteburn.htm>> Acesso em: 21 set. de 2022.

PAULINO JÚNIOR, J. W.; OLIVEIRA, F. K. Pensamento computacional com ênfase no ensino de Lógica de programação: revisão sistemática de literatura: Computational thinking with emphasis on teaching programming logic: systematic literature review. *Revista Semiárido De Visu*, [S. l.], v. 10, n. 3, 2022. DOI: 10.31416/rsdv.v10i3.396. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/396>. Acesso em: 26 jul. 2023.

PAULINO JÚNIOR, J. W.; OLIVEIRA, F. K. de. Pensamento Computacional como metodologia para o ensino de Lógica de Programação. *Revista Semiárido De Visu*, [S. l.], v. 12, n. 1, p. 539–550, 2024. DOI: 10.31416/rsdv.v12i1.394. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/394>. Acesso em: 7 mar. 2024.

PAPERT, Seymour. **Teaching children to be mathematicians us. teaching about mathematics**. memo, 1980.

PAPERT, Seymour. *The children's machine: Rethinking school in the age of the computer*. **New York**, 1993.

PIMENTA, Diogo de Farias et al. *Desenvolvendo o raciocínio lógico e o pensamento computacional na educação básica: uma experiência com tecnologias educacionais e lógica de programação*. 2019.

POLONI, Leonardo; DO SACRAMENTO SOARES, Eliana Maria; WEBBER, Carine G. Pensamento computacional no ensino médio: práticas mediadoras utilizando a linguagem Scratch. **RENOTE**, v. 17, n. 3, p. 508-517, 2019.

RAMOS, Marise Nogueira. *A pedagogia das competências*. São Paulo: Cortez, 2002.

SILVEIRA, S. R. et al. **Formação Docente: Desenvolvimento do pensamento computacional nos anos finais do ensino fundamental. Projeto de Extensão.** Frederico Westphalen: UFSM, 2019.

RAMOS, Fellipe; DA SILVA TEIXEIRA, Lilian. Significação da aprendizagem através do pensamento computacional no ensino médio: uma experiência com scratch. In: **Anais do Workshop de Informática na Escola.** 2015. p. 217-226.

RADIANTI, Jaziar et al. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. **Computers & Education**, v. 147, p. 103778, 2020.

RATTRAY, Janice; JONES, Martyn C. Essential elements of questionnaire design and development. **Journal of clinical nursing**, v. 16, n. 2, p. 234-243, 2007.

X RODRIGUES, Renan Kodama. **Estudo do uso da linguagem de blocos scratch no ensino do pensamento computacional.** 2019. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

ROCHA, E. dos S.; OLIVEIRA, F. K. de. Critérios para divisão de recursos financeiros e orçamentários na execução da Política de assistência estudantil em instituições da rede federal de educação:: uma revisão sistemática de literatura. *Revista Semiárido De Visu*, [S. l.], v. 11, n. 3, p. 682–698, 2023. DOI: 10.31416/rsdv.v11i3.781. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/781>. Acesso em: 1 jan. 2024.

SILVA, G. H. da; OLIVEIRA, F. K. de. Mapeamento Sistemático de Literatura sobre Pensamento Matemático-Computacional. **Revista Semiárido De Visu**, [S. l.], v. 11, n. 3, p. 637–648, 2023. DOI: 10.31416/rsdv.v11i3.788. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/788>. Acesso em: 1 jan. 2024.

SILVA, G. H.; OLIVEIRA, F. K. Pensamento Computacional e STEAM numa perspectiva interdisciplinar com a matemática e informática: Uma revisão sistemática de literatura: Computational Thinking and STEAM in an interdisciplinary perspective

with mathematics and informatics: A systematic literature review. **Revista Semiárido De Visu**, v. 10, n. 3, 2022. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/399>. Acesso em: 01 jan. 2024.

SILVA, G. G.; OLIVEIRA, F. K. de. Material didático utilizado na Educação Profissional de Jovens e Adultos: Uma revisão sistemática da literatura. **Revista Semiárido De Visu**, [S. l.], v. 9, n. 3, p. 335–343, 2021. DOI: 10.31416/rsdv.v9i3.308. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/308>. Acesso em: 01 jan. 2024.

SILVEIRA, Sidnei Renato et al. FORMAÇÃO DOCENTE E O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL NOS ANOS FINAIS DO ENSINO FUNDAMENTAL. **Educação Básica Revista**, v. 6, n. 2, p. 55-70, 2020.

SILVA, T. R. *et al* (2015). **Ensino-aprendizagem de programação**: uma revisão sistemática da literatura. RBIE, 23(01):182. Disponível em <[www.br-ie.org/pub/index.php/rbie/article/view/2838](http://www.br-ie.org/pub/index.php/rbie/article/view/2838)>. Acesso em 15 de Maio de 2023.

SOUZA, D. M.; BATISTA, M. H. S.; BARBOSA, E. F. (2016). **Problemas e dificuldades no ensino de programação**: Um mapeamento sistemático. Revista Brasileira de Informática na Educação, 24(1):39. Disponível em <<http://www.br-ie.org/pub/index.php/rbie/article/view/3317>>. Acesso em 10 Out. 2023.

SUN, Liping et al. A systematic literature review of teacher scaffolding in game-based learning in primary education. **Educational Research Review**, p. 100546, 2023.

SCHOEFFEL, Pablo et al. Uma experiência no ensino de pensamento computacional para alunos do ensino fundamental. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2015. p. 1474.

TRIDAPALLI, Joan Giancesini; MATTOS, Mauro Marcelo. PENSAMENTO COMPUTACIONAL E GAMIFICATION: RELATO DE UM EXPERIMENTO NA PLATAFORMA FURBOT.

VALENTE, José Armando. Liberando a mente: computadores na educação especial. **Campinas: Unicamp**, 1991.

VALENTE, José Armando et al. Informática na educação no Brasil: análise e contextualização histórica. **O computador na sociedade do conhecimento. Campinas: UNICAMP/NIED**, p. 1-13, 1999.

VON WANGENHEIM, Christiane Gresse; NUNES, Vinícius Rodrigues; DOS SANTOS, Giovane Daniel. Ensino de computação com scratch no ensino fundamental—um estudo de caso. **Revista Brasileira de Informática na Educação**, v. 22, n. 03, p. 115, 2014.

VIEIRA, A. M. D. P.; SOUZA JUNIOR, A. A Educação Profissional no Brasil. *Revista Interacções*, Santarém, n.40, p.152-169, 2016.

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

ZANETTI, Humberto; BORGES, Marcos; RICARTE, Ivan. Pensamento computacional no ensino de programação: Uma revisão sistemática da literatura brasileira. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2016. p. 21.

1 - Utilizando a metodologia de pensamento computacional, você é estimulado e desafiado a buscar aprendizado a partir do conteúdo que está aprendendo ?

- a) Concordo plenamente
- b) Concordo
- c) Neutro
- d) Discordo
- e) Discordo fortemente

2- Os objetivos (conhecimentos, habilidades e atitudes) da disciplina de lógica do programação foram alcançados?

- a) Concordo plenamente
- b) Concordo
- c) Neutro
- d) Discordo
- e) Discordo fortemente

3- Como você acha que pode aprender mais sobre esses tópicos?

- a) Metodologia PC
- b) Metodologia Tradicional

4- Quer utilizar a metodologia PC em outras disciplinas do curso técnico?

- a) sim
- b) não

5- Como você acha que o conteúdo técnico deve ser ensinado em sala de aula?

- a) Palestra
- b) PC
- c) Pesquisa
- d) Jogos de aprendizagem
- e) Experiências

6- Você acredita que a metodologia PC teve um efeito positivo na sua nota no curso de programação lógica?

- a) Concordo plenamente
- b) Concordo
- c) Neutro
- d) Discordo



- e) Discordo fortemente
- 7- Você gostaria que as disciplinas do curso técnico utilizassem a metodologia PC?
- a) Concordo plenamente
  - b) Concordo
  - c) Neutro
  - d) Discordo
  - e) Discordo fortemente
- 8- Qual a sua validação quanto a utilização da metodologia de PC na disciplina de Lógica de Programação?
- a) Péssimo
  - b) Ruim
  - c) Regular
  - d) Bom
  - e) Ótimo
- 9- Qual a sua opinião sobre seu desempenho na disciplina de lógica de programação utilizando a metodologia de PC?
- a) Péssimo
  - b) Ruim
  - c) Regular
  - d) Bom
  - e) Ótimo
- 10- Enumere de 1 a 5 acerca dos pilares do PC que contribuíram para a sua aprendizagem
- a) Decomposição
  - b) Reconhecimento de Padrões
  - c) Abstração
  - d) Algoritmo

## **APÊNDICE B – ANÁLISE DE METODOLOGIA**

1- Nas aulas de lógica de programação, por meio da metodologia tradicional, quais dos itens abaixo (em uma escala de 0 a 5) você consegue perceber:

- a) Coleta informações adequadas e selecionar informações relevantes para o contexto da atividade desenvolvidas (coleta de dados) (    )
- b) Dar sentido aos dados, encontra padrões e obtém conclusões a partir da observação de dados (análise de dados) (    )
- c) Organiza e descreve dados em representações apropriadas, como gráficos, palavras, frases, imagens, tabelas,etc, (representação de dados) (    )
- d) Divide tarefas grandes em partes menores e mais fáceis de gerenciar e resolver (decomposição) (    )
- e) Planeja e organiza sequências de passos para resolver um problema ou alcançar algum objetivo (algoritmo)
- f) Reduz a complexidade para criar ideias e encontrar características básicas e criar modelos (abstração) (    )

## **APÊNDICE C - ENTREVISTA SEMIESTRUTURADA COM ESPECIALISTAS**

Entrevistados: docentes que atuam no EMI ao curso técnico em Informática com

formação na área de Ciência da Computação ou áreas correlatas.

1- O que você entende por Pensamento Computacional?

2-Quais as principais dificuldades encontradas pelos estudantes ao estudar Lógica de Programação?

3-Você acredita que a metodologia de Pensamento Computacional pode vir a contribuir com o desenvolvimento do raciocínio lógico para a resolução de problemas computacionais?

4-Você acredita no Pensamento Computacional como alternativa ao ensino tradicional de Lógica de Programação?

5-Na sua opinião, o que faz o Pensamento Computacional ser um tipo de metodologia multidisciplinar?

## **APÊNDICE D - PROBLEMAS ABORDADOS NA INTERVENÇÃO**

### **Problema 1 - Sistema para cálculo de notas**

O sistema de cálculo de média é uma ferramenta fundamental na análise do desempenho em diversas áreas, desde a educação até o mundo corporativo. Esse

sistema visa simplificar o processo de avaliação, proporcionando uma visão objetiva e quantitativa do rendimento de um indivíduo ou de um grupo.

Funcionamento: O sistema opera de maneira intuitiva, permitindo a inserção de valores referentes a diferentes avaliações ou medições. Seja em notas escolares, avaliações de desempenho profissional ou em qualquer contexto que exija uma média, o sistema proporciona uma estrutura flexível para a inclusão de dados.

Principais características:

#### 1 - **Entrada de Dados:**

- O usuário insere os valores correspondentes aos resultados das avaliações.
- O sistema aceita tanto valores numéricos como escalas específicas, dependendo do contexto.

#### 2. **Ponderação:**

- Em alguns casos, certas avaliações podem ter pesos diferentes. O sistema permite a atribuição de pesos para cada valor, refletindo a importância relativa de cada componente na média final.

#### 3. **Cálculo Automático:**

- Uma vez que os dados são inseridos, o sistema realiza automaticamente os cálculos necessários para obter a média.
- Fórmulas matemáticas são aplicadas de acordo com a metodologia escolhida, seja a média aritmética simples, ponderada ou outra.

#### 4. **Exibição de Resultados:**

- O resultado final é apresentado de forma clara e acessível, proporcionando uma compreensão imediata do desempenho global.
- Gráficos e estatísticas adicionais podem ser gerados para uma análise mais aprofundada.

Ao final, foram feitos os seguintes questionamentos:

1. Qual estratégia sua equipe utilizou?
2. Qual outro problema você resolveria dessa mesma maneira?
- 3.

### **Problema 2 - Sistema de Calculadora de Índice de Massa Corporal**

A Calculadora de Índice de Massa Corporal (IMC) é um sistema projetado para calcular o IMC de uma pessoa com base nas informações fornecidas sobre peso e

altura. O IMC é uma medida útil para avaliar se uma pessoa está dentro de uma faixa de peso saudável de acordo com sua altura.

### **Funcionalidades:**

#### **Registro de Informações:**

Os usuários podem inserir seu peso e altura no sistema.

#### **Cálculo Automático do IMC:**

Com base nas informações fornecidas, o sistema realiza automaticamente o cálculo do IMC usando a fórmula padrão:  $IMC = \text{Peso (kg)} / (\text{Altura (m)})^2$ .

- cálculo do IMC usando a fórmula padrão:  $IMC = \text{Peso (kg)} / (\text{Altura (m)})^2$ .
- **Classificação do IMC:**
  - Após o cálculo, o sistema classifica o IMC do usuário em categorias, como abaixo do peso, peso normal, sobrepeso, obesidade leve, obesidade moderada e obesidade grave.

### **Benefícios:**

- **Consciência da Saúde:** Ajuda os usuários a entenderem sua condição de peso e a tomar medidas para melhorar a saúde.
- **Acompanhamento Pessoal:** Permite que os usuários acompanhem seu progresso ao longo do tempo, incentivando a manutenção de um estilo de vida saudável.
- **Educação sobre Saúde:** Fornece informações educativas sobre a importância do IMC e suas implicações para a saúde.

A Calculadora de IMC é uma ferramenta prática e educativa para indivíduos que desejam monitorar e melhorar sua saúde através do controle do peso. Ao oferecer informações claras e conselhos personalizados, o sistema visa incentivar escolhas de estilo de vida saudáveis.

Ao final, foram feitos os seguintes questionamentos:

4. Qual estratégia sua equipe utilizou?
5. Qual outro problema você resolveria desta mesma maneira?

### **Problema 3 - Sistema de cadastro de livros**

O Sistema de Cadastro de Livros é uma solução intuitiva e eficiente projetada para simplificar o processo de gestão do acervo em bibliotecas. Com foco na organização, acessibilidade e praticidade, esse sistema oferece funcionalidades específicas para catalogar, atualizar e disponibilizar informações sobre cada livro de forma rápida e eficaz.

#### **Principais Funcionalidades:**

##### **1. Cadastro Detalhado:**

- Permite o registro detalhado de cada livro, incluindo informações como título, autor, editora, ano de publicação, número de páginas, ISBN, resumo e capa.
- Facilita a busca rápida e precisa por qualquer critério.

##### **2. Classificação por Categorias:**

- Categoriza os livros em diferentes gêneros, temas ou assuntos, simplificando a organização física da biblioteca.
- Possibilita a aplicação de múltiplas etiquetas para facilitar a busca por diferentes critérios.

#### **Vantagens:**

##### **1. Eficiência na Gestão:**

- Agiliza o processo de cadastro e atualização de informações, economizando tempo e recursos operacionais.

##### **2. Facilidade de Busca:**

- Proporciona uma busca fácil e rápida por meio de diferentes critérios, promovendo a eficácia na localização de livros.

##### **3. Otimização do Espaço Físico:**

- Contribui para a organização física da biblioteca ao permitir uma categorização eficiente dos livros

O Sistema de Cadastro de Livros é uma ferramenta essencial para bibliotecas modernas que buscam otimizar a gestão de seu acervo, proporcionando uma administração eficiente e garantindo o acesso facilitado e organizado à vasta gama de conhecimento contido em seus livros.

Ao final, foram feitos os seguintes questionamentos:

6. Qual estratégia sua equipe utilizou?
7. Qual outro problema você resolveria desta mesma maneira?

#### **Problema 4**

O Sistema de Computação de Votos para o Grêmio Estudantil é uma plataforma inovadora desenvolvida para modernizar o processo eleitoral, proporcionando transparência, segurança e eficiência na escolha dos representantes estudantis. Com foco na facilitação do exercício democrático, este sistema oferece funcionalidades avançadas para garantir um processo de votação eficaz e confiável.

- 1. Transparência e Credibilidade:**
  - Torna o processo eleitoral mais transparente, com resultados disponíveis em tempo real e garantia de integridade dos votos.
- 2. Aumento da Participação:**
  - Facilita a participação dos estudantes, permitindo que votem de qualquer lugar, promovendo uma maior representatividade.
- 3. Redução de Custos e Impacto Ambiental:**
  - Elimina a necessidade de papel e reduz custos associados à organização de eleições presenciais, ao mesmo tempo em que minimiza o impacto ambiental.
- 4. Segurança Eleitoral:**
  - Implementa protocolos avançados de segurança cibernética para proteger contra tentativas de manipulação ou interferência no processo eleitoral.
- 5. Facilidade na Administração:**
  - Simplifica o trabalho dos organizadores das eleições, oferecendo

ferramentas intuitivas e relatórios automáticos.

O Sistema de Computação de Votos para o Grêmio Estudantil é uma ferramenta indispensável para instituições de ensino que buscam promover eleições estudantis transparentes, eficientes e alinhadas aos princípios democráticos, proporcionando uma experiência eleitoral moderna e acessível aos estudantes.

Ao final, foram feitos os seguintes questionamentos:

8. Qual estratégia sua equipe utilizou?
9. Qual outro problema você resolveria dessa mesma maneira?

## **APÊNDICE E - SEQUÊNCIA DIDÁTICA**





**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO SERTÃO**  
**PERNAMBUCANO**  
**PROFEPT - PROGRAMA DE PÓS-GRADUAÇÃO EM EDUCAÇÃO PROFISSIONAL E**  
**TECNOLÓGICA**  
**MESTRADO PROFISSIONAL EM EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**

**INFORMAÇÕES DO MINISTRANTE/TURMA**

<b>Série/Turma</b>	docentes que atuam no EMI ao curso técnico em Informática com formação na área de Ciência da Computação ou áreas correlatas.
<b>Ministrante</b>	José Walter Paulino Júnior

**DESCRIÇÃO DA AULA**

<b>Assunto(s)</b>	
<b>Objetivos</b>	Apresentar o Pensamento Computacional e como ele pode estimular o ensino da lógica de programação para a resolução de problemas.
<b>Conteúdos</b>	Apresentar a competência da BNCC que trata do pensamento computacional; Apresentar as etapas do pensamento computacional; Contextualizar a importância de sua aplicação; Apresentar o Scratch como alternativa ao ensino de Lógica de Programação utilizando o pensamento computacional.
<b>Duração</b>	90 minutos (2 aulas)
<b>Procedimentos metodológicos</b>	Aula expositiva e dialogadas, com utilização de ferramentas práticas
<b>Recursos didáticos</b>	Google Meet; Smartphone.

<b>Avaliação</b>	Experimentação das ferramentas digitais
<b>Bibliografia</b>	<p>GOMES, T. A.; RODRIGUES, C. K. <b>A evolução das tendências da educação matemática e o enfoque da história da matemática no ensino</b>. Revista de Educação, Ciências e Matemática, v.4 n.3 set/dez 2014.</p> <p>LIBÂNEO, J. C. <b>Os métodos de ensino</b>. São Paulo: Cortez, 1994. P. 149-176</p> <p>PRENSKY, Marc. <b>Digital natives, digital immigrants. On the Horizon</b>, v.9, n.5. MCB University Press. Out. 2005. Disponível em: &lt;  <a href="http://marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part%201.pdf">http://marcprensky.com/writing/Prensky%20-%20Digital%20Natives,%20Digital%20Immigrants%20-%20Part 1.pdf</a>&gt;. Acesso em:02 set.2022</p>

## APÊNDICE F - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

### TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

O (A) Sr(a) está sendo convidado(a) a participar da pesquisa intitulada “aplicação e avaliação da abordagem PC no componente curricular de lógica de

programação” no curso técnico de informática na Escola Estadual de Educação Profissional Irmã Ana Zélia da Fonseca na Cidade de Milagres – Ceará, sob a responsabilidade do professor e pesquisador José Walter Paulino Júnior. Nesta pesquisa está sendo pesquisado como melhorar as notas dos discentes com dificuldades de aprendizagem e melhorar a motivação e engajamento na referida disciplina. Para esta pesquisa adotaremos os seguintes procedimentos: aplicar questionário aos discentes ao final da disciplina de lógica de programação do semestre de 2023 para avaliar as suas percepções com a utilização da metodologia de Pensamento Computacional e por fim analisar as percepções do docente das disciplinas que foram anotadas no diário de bordo. Na sua participação, o Sr. (a) não terá nenhum custo, nem receberá qualquer vantagem financeira. O (A) Sr. (a) terá o esclarecimento sobre o estudo em qualquer aspecto que desejar e estará livre para participar ou recusar-se a participar. Até o momento da divulgação dos resultados, você também é livre para solicitar a retirada dos seus dados da pesquisa. A sua participação é voluntária e a recusa em participar não acarretará qualquer penalidade ou modificação na forma em que o Sr. (a) é atendido (a) pelo pesquisador, que tratará a sua identidade com padrões profissionais de sigilo. Os resultados da pesquisa estarão à sua disposição quando finalizada. O (A) Sr. (a) não será identificado (a) em nenhuma publicação que possa resultar. Os pesquisadores tratarão a sua identidade com padrões profissionais de sigilo, atendendo a legislação brasileira, utilizando as informações somente para fins acadêmicos e científicos. Nome do pesquisador responsável: José Walter Paulino Júnior. Fone: (88) 9 98118497 Email: jose.walter@aluno.ifsertao-pe.edu.br Curso: Mestrado Profissional em Educação Profissional e Tecnológica Instituição: Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano - IFSertãoPE.

#### **APÊNDICE G - Matriz Curricular**

<b>Disciplinas</b>	<b>1º Série</b>	<b>2º Série</b>	<b>3º Série</b>
<b>Informática Básica</b>	<b>X</b>		
<b>Lógica de Programação</b>	<b>X</b>		
<b>Arquitetura e Manutenção de Computadores</b>	<b>X</b>		
<b>HTML/CSS</b>	<b>X</b>		

<b>Planejamento de Carreira</b>	<b>X</b>		
<b>Sistemas Operacionais</b>		<b>X</b>	
<b>Programação Orientada a Objetos</b>		<b>X</b>	
<b>Programação WEB</b>		<b>X</b>	
<b>Noções de Robótica</b>		<b>X</b>	
<b>Gestão de Startups I</b>		<b>X</b>	
<b>Redes de Computadores</b>		<b>X</b>	
<b>Design Gráfico</b>		<b>X</b>	
<b>Gerenciador de Conteúdo</b>		<b>X</b>	
<b>Banco de Dados</b>		<b>X</b>	
<b>Gestão de Startups II</b>		<b>X</b>	
<b>Laboratório de Software</b>			<b>X</b>
<b>Laboratório de Hardware</b>			<b>X</b>
<b>Projeto Integrador</b>			<b>X</b>
<b>Gestão de Startups III</b>			<b>X</b>
<b>Estágio Curricular</b>			<b>X</b>

## **APÊNDICE H - CARTA DE ANUÊNCIA**

Escola Estadual de Educação Profissional Irmã Ana Zélia da Fonseca

Endereço: R. Raimundo Tavares da Cruz, sn - Eucaliptos, Milagres - CE, 63250-000,

Telefone: (88) 3553-1811

## **CARTA DE ANUÊNCIA**

Declaramos, para os devidos fins, que aceitaremos (o) a pesquisador (a) JOSÉ WALTER PAULINO JÚNIOR a desenvolver o seu projeto de pesquisa “**Lógica de Programação no Ensino Médio Integrado: Aplicação e Avaliação do Pensamento Computacional no Curso Técnico em Informática**” no período de 21/11 a 20/12. O projeto está sob a coordenação/orientação de FRANCISCO KELSEN DE OLIVEIRA e seu objetivo é o de investigar as notas dos discentes com dificuldades de aprendizagem de Lógica de Programação e melhorar a motivação e o engajamento na referida disciplina por meio da utilização de recursos visuais. Para esta pesquisa adotaremos os seguintes procedimentos: aplicar questionário aos discentes durante as aulas da disciplina de Lógica de Programação do segundo semestre de 2023 para avaliar as suas percepções com a utilização da metodologia de Pensamento Computacional e por fim, analisar também as percepções do docente das disciplinas que foram registradas em diário de bordo nesta Instituição Escolar.

A aceitação está condicionada ao cumprimento do (a) pesquisador (a) aos requisitos das Resoluções 466/12 e 516/16 do CNS/MS e suas complementares, comprometendo-se a utilizar os dados e materiais coletados exclusivamente para os fins da pesquisa.

Local, em \_\_\_\_/\_\_\_\_/\_\_\_\_.

---

Assinatura

Wergila de Souza Tavares

Coordenadora Escolar

## **APÊNDICE I - TERMO DE ASSENTIMENTO LIVRE E ESCLARECIDO PARA MENORES DE 18 ANOS**

### **ASSENTIMENTO LIVRE E ESCLARECIDO PARA MENORES DE 18 ANOS**

O que é assentimento?

O assentimento significa que você concorda em participar de uma pesquisa,

na qual serão respeitados seus direitos e você receberá todas as informações necessárias para compreender a importância de sua participação.

Convidamos você para participar, como voluntário (a), da pesquisa: **LÓGICA DE PROGRAMAÇÃO NO ENSINO MÉDIO INTEGRADO: APLICAÇÃO E AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL NO CURSO TÉCNICO EM INFORMÁTICA** que está sob a responsabilidade José Walter Paulino Júnior

Também participam desta pesquisa: Francisco Kelsen de Oliveira (orientador do estudo), e-mail:

### **INFORMAÇÕES SOBRE A PESQUISA:**

#### **Participação na pesquisa:**

O objetivo deste estudo é analisar a aplicação das habilidades do Pensamento Computacional como alternativa metodológica ao ensino tradicional no processo de aprendizagem de Lógica de Programação no curso técnico em Informática Integrado ao Ensino Médio da EEEP Irmã Ana Zélia da Fonseca. Para isso será realizada uma intervenção em sala de aula onde os participantes irão assistir um curso sobre Pensamento Computacional ministrado por este pesquisador. A intervenção será realizada nas disciplinas de Programação Web, Programação Orientada a Objetos e Banco de Dados, tendo sua duração de aproximadamente 3 semanas. Seguido da aplicação de exercícios sobre algoritmos utilizando como estratégia de resolução a metodologia de Pensamento Computacional e por fim a aplicação de 02 questionários via formulários do Google para avaliar o impacto e a percepção dos estudantes sobre a intervenção que foi realizada. Serão coletadas informações dos estudantes a respeito da metodologia e grau de dificuldade das soluções. Não haverá nenhuma outra forma de envolvimento ou comprometimento neste estudo. Em caso de gravação, filmagem ou fotos, a realização desses procedimentos será previamente avisada.

#### **Local da pesquisa:**

A pesquisa será realizada presencialmente na EEEP Irmã Ana Zélia da Fonseca, localizada na Rua Professora Maria de Fátima Tavares Leite, S/N no Bairro Eucaliptos, Milagres-CE, a durante as aulas das disciplinas de Programação Web, Programação Orientada a Objetos e Banco de Dados em turmas de primeiro e

segundo ano de nível médio da modalidade integrada ao técnico em Informática.

### **Benefícios e riscos decorrentes da Participação na pesquisa:**

A pesquisa é considerada segura, mas é possível que ocorram riscos mínimos previsíveis. Os riscos previsíveis nesta pesquisa podem envolver aspectos psicológicos, com a exposição a situações de confronto no qual possam encontrar dificuldades em resolver problemas que necessitam de uma base matemática, mobilização de emoções, sentimentos, experiências de decisão, frustração e conflitos de percepção que podem gerar um nível de estresse baixo.

É previsto que a duração do risco seja transitória e de nível mínimo, haja vista que o desconforto psicológico que pode ser causado pela intervenção na pesquisa, são equiparados ao observado/relatado em situações reais do cotidiano, cabendo frisar o cuidado e atenção maior por se tratar de uma população vulnerável ao sofrimento psíquico e em uma fase crítica do desenvolvimento biopsicossocial. Fica também garantida a indenização em casos de danos, comprovadamente decorrentes da participação na pesquisa, conforme decisão judicial ou extrajudicial.

Os benefícios esperados para os participantes devem ser transitórios e também permanentes, sendo estimados impactos positivos em termos educacionais, como o desenvolvimento da habilidade de resolução de problemas, autonomia, aumento da motivação, interesse, o que deve impactar os participantes a curto, médio e longo prazo.

Benefícios estes que proporcionam o desenvolvimento de habilidades essenciais para os profissionais do século XXI, como trabalho em grupo, pensamento crítico e reflexivo, resolução de problemas e comunicação. Além disso, espera-se que o produto educacional proveniente deste estudo contribua para aprimorar o processo de ensino de lógica de programação através do Pensamento Computacional no curso técnico em Informática da EEEP Irmã Ana Zélia da Fonseca e na Educação Profissional e Tecnológica como um todo.

### **Autonomia e sigilo do participante da pesquisa:**

Sua participação é voluntária. Você não terá nenhum prejuízo se não quiser participar. Você poderá se retirar desta pesquisa a qualquer momento, bastando

para isso entrar em contato com o pesquisador responsável. Seus dados serão manuseados somente pelo pesquisador e pelo orientador e não será permitido o acesso a outras pessoas. Os resultados deste trabalho poderão ser apresentados em encontros ou revistas científicas, entretanto, ele mostrará apenas os resultados obtidos como um todo, sem revelar seu nome ou qualquer informação que esteja relacionada com sua privacidade.

### **ASSENTIMENTO DA PARTICIPAÇÃO DO SUJEITO COMO VOLUNTÁRIO (A)**

As informações desta pesquisa serão confidenciais e serão divulgadas apenas em eventos ou publicações científicas, não havendo identificação dos voluntários, a não ser entre os responsáveis pelo estudo, sendo assegurado o sigilo sobre a participação dos voluntários, a menos que seja autorizado pelo participante da pesquisa.

Os dados coletados nesta pesquisa serão apenas as respostas do tipo texto, por meio da aplicação dos formulários e ficarão armazenados em pasta de arquivo em drive, sob a responsabilidade do pesquisador e do orientador.

O (A) voluntário (a) não pagará nada para participar desta pesquisa. Se houver necessidade, as despesas para a participação serão assumidas ou ressarcidas pelos pesquisadores. Fica também garantida a indenização em casos de danos, comprovadamente decorrentes da participação na pesquisa, conforme decisão judicial ou extrajudicial.

Caso tenha alguma dúvida, pergunte à pessoa que está lhe entrevistando, seus pais e/ou seu responsável legal para que esteja bem esclarecido (a) sobre sua participação na pesquisa.

Após ser esclarecido (a) sobre as informações a seguir, no caso de aceitar em fazer parte do estudo, assine este documento que será rubricado e assinado também por seus pais ou seu responsável legal, que está em duas vias, uma delas é sua e a outra é do pesquisador responsável.

Se você não quiser participar da pesquisa, não será prejudicado (a) de forma alguma e tem o direito de desistir da participação na pesquisa em qualquer momento.

Em caso de dúvidas relacionadas aos aspectos éticos deste estudo, você poderá consultar o Comitê de Ética em Pesquisa com Seres Humanos do IF



SERTÃO-PE, Reitoria – Rua Aristarco Lopes, 240, Centro, CEP 56.302-100, Petrolina-PE, Telefone: (87) 2101-2350 / Ramal 2364, <http://www.ifsertao-pe.edu.br/index.php/comite-de-etica-em-pesquisa>, [cep@ifsertao-pe.edu.br](mailto:cep@ifsertao-pe.edu.br); ou poderá consultar a Comissão nacional de Ética em Pesquisa, Telefone (61)3315-5878, [conep.cep@saude.gov.br](mailto:conep.cep@saude.gov.br).

O Comitê de Ética em Pesquisa (CEP) é um colegiado interdisciplinar e independente, que deve existir nas instituições que realizam pesquisas envolvendo seres humanos no Brasil, criado para defender os interesses dos sujeitos da pesquisa em sua integridade e dignidade, objetivando contribuir no desenvolvimento da pesquisa dentro de padrões éticos. O Comitê de Ética em Pesquisa (CEP) é responsável pela avaliação e acompanhamento dos aspectos éticos de todas as pesquisas envolvendo seres humanos. Por isso, o cep estará a disposição caso você deseje maiores informações.

---

Assinatura do (a)Pesquisador (a)

---

Assinatura do (a)Voluntário (a)

---

Assinatura do (a) Responsável Legal ou Pais

### **ASSENTIMENTO DA PARTICIPAÇÃO DO SUJEITO COMO VOLUNTÁRIO (A)**

Eu,

---

portador(a) do documento de Identidade: \_\_\_\_\_ CPF

\_\_\_\_\_ (se já tiver documento), fui informado(a) dos objetivos da pesquisa de maneira clara/ detalhada e esclareci minhas dúvidas.

Sei que a qualquer momento poderei solicitar novas informações, e o meu responsável legal poderá modificar a decisão de minha participação se assim o

desejar, mesmo já tendo assinado o consentimento/ assentimento.

Declaro que concordo em participar dessa pesquisa. Recebi uma cópia deste termo de assentimento e me foi dada a oportunidade de ler e esclarecer as minhas dúvidas.

Local e data \_\_\_\_\_ ,

\_\_\_\_\_.

---

Assinatura do (da) Participante/ Voluntário (a)

---

Assinatura do (a) Responsável Legal ou Pais

## **APÊNDICE J - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA MAIORES DE 18 ANOS**

### **TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA MAIORES DE 18 ANOS**

Convidamos o(a) Sr.(a), docente de Informática para participar como voluntário(a) da pesquisa **LÓGICA DE PROGRAMAÇÃO NO ENSINO MÉDIO**

**INTEGRADO: APLICAÇÃO E AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL NO CURSO TÉCNICO EM INFORMÁTICA**, que está sob a responsabilidade do pesquisador José Walter Paulino Júnior, tendo como instituição proponente o Instituto Federal de Educação, Ciência e Tecnologia do Sertão Pernambucano – campus Salgueiro, localizado na BR 232, Km 508, sentido Recife, Zona Rural – Salgueiro – PE – Brasil. CEP: 56.000-000/Telefone: (87) 3421.0050, e está sob a orientação do professor Dr. Francisco Kelsen de Oliveira.

Ao ler este documento, caso haja alguma dúvida, pergunte à pessoa que está lhe entrevistando, para que o/a senhor/a esteja bem esclarecido(a) sobre tudo que está respondendo. Após ser esclarecido(a) sobre as informações a seguir, caso aceite em fazer parte do estudo, rubrique as folhas e assine ao final deste documento, que está em duas vias. Uma delas é sua e a outra é do pesquisador responsável. Em caso de recusa o(a) Sr.(a) não será penalizado(a) de forma alguma. Também garantimos que o(a) Senhor(a) tem o direito de retirar o consentimento da sua participação em qualquer fase da pesquisa, sem qualquer penalidade.

### **INFORMAÇÕES SOBRE A PESQUISA:**

#### **Participação na pesquisa:**

O objetivo deste estudo é analisar a aplicação das habilidades inerentes ao Pensamento Computacional como recurso metodológico no processo de aprendizagem de Lógica de Programação no curso Técnico em Informática Integrado ao Ensino Médio da EEEP Irmã Ana Zélia da Fonseca. Para isso será realizada uma intervenção onde os participantes irão assistir um minicurso sobre Pensamento Computacional ministrado por este pesquisador. A intervenção será realizada aos docentes que ministram as disciplinas de Programação Web, Programação Orientada a Objetos e Banco de Dados, tendo sua duração de aproximadamente 120 minutos divididos em dois encontros, cada um de 60 minutos através da plataforma Google Hangouts. Seguido da aplicação de 01 questionário via formulários do Google para avaliar o impacto e a percepção sobre os fundamentos do Pensamento Computacional. Serão coletadas informações dos docentes a respeito da metodologia e viabilidade de sua implementação. Não haverá nenhuma outra forma de envolvimento ou comprometimento neste estudo. Em caso

de gravação, filmagem ou fotos, a realização desses procedimentos será previamente avisada.

**Local da pesquisa:**

A pesquisa será realizada de forma virtual por meio de videoconferência utilizando o aplicativo Google Hangouts e a aplicação do questionário via Formulários do Google.

**Benefícios e riscos decorrentes da Participação na pesquisa:**

O minicurso será ofertado via videotransmissão podendo ocorrer riscos relacionados a atrasos de conexão que podem levar por exemplo perda de informações que os docentes possam julgar importantes numa possível tomada de decisão em relação a metodologia proposta e as habilidades do Pensamento Computacional como alternativa ao ensino de Lógica de Programação. Outros riscos previsíveis nesta pesquisa envolvem aspectos psicológicos, como cansaço ou aborrecimento ao responder o questionário que pode gerar um nível de estresse baixo ou moderado, sendo que sua participação é voluntária, e você não participará do minicurso, caso assim deseje.

Estima-se que a duração dos riscos seja transitória e de nível mínimo a moderado, haja visto que os riscos que podem ser causados pela intervenção na pesquisa, são equiparados ao observado/relatado em situações reais do cotidiano.

Fica também garantida a indenização em casos de danos, comprovadamente decorrentes da participação na pesquisa, conforme decisão judicial ou extrajudicial

Os benefícios esperados para os participantes devem ser transitórios e também permanentes, sendo estimados impactos positivos em termos educacionais, como a adoção das habilidades e metodologias do Pensamento Computacional em sua prática didática para o ensino de Lógica de Programação, bem como a utilização dos conceitos aprendidos para a resolução de problemas do seu contexto. Ademais, espera-se que a pesquisa possa produzir conhecimentos generalizáveis que permita também uma maior compreensão sobre a metodologia de Pensamento Computacional fortalecendo sua prática didática na Educação Profissional e Tecnológica, referente ao Ensino Médio Integrado.

**Autonomia e sigilo do participante da pesquisa:**

Sua participação é voluntária. O(a) senhor(a) não terá nenhum prejuízo se não quiser participar. O(a) senhor(a) poderá se retirar desta pesquisa a qualquer momento, bastando para isso entrar em contato com o pesquisador responsável. Seus dados serão manuseados somente pelo pesquisador e pelo orientador e não será permitido o acesso a outras pessoas. Os resultados deste trabalho poderão ser apresentados em encontros ou revistas científicas, entretanto, ele mostrará apenas os resultados obtidos como um todo, sem revelar seu nome ou qualquer informação que esteja relacionada com sua privacidade.

Os dados coletados nesta pesquisa serão apenas as respostas do tipo texto, por meio da aplicação dos formulários e ficarão armazenados em pasta de arquivo em drive, sob a responsabilidade do pesquisador e do orientador.

O(a) senhor(a) não pagará nada para participar desta pesquisa. Se houver necessidade, as despesas para a sua participação serão assumidas pelos pesquisadores (ressarcimento de transporte e alimentação).

Em caso de dúvidas relacionadas aos aspectos éticos deste estudo, você poderá consultar o Comitê de Ética em Pesquisa Envolvendo Seres Humanos do IF SERTÃO-PE no endereço: Reitoria – Rua Aristarco Lopes, 240, Centro, CEP 56.302-100, Petrolina-PE, Telefone: (87) 2101-2350 / Ramal 2364, <http://www.ifsertao-pe.edu.br/index.php/comite-de-etica-em-pesquisa>, [cep@ifsertao-pe.edu.br](mailto:cep@ifsertao-pe.edu.br); ou poderá consultar a Comissão nacional de Ética em Pesquisa, Telefone (61)3315-5878, [conep.cep@saude.gov.br](mailto:conep.cep@saude.gov.br).

O Comitê de Ética em Pesquisa (CEP) é um colegiado interdisciplinar e independente, que deve existir nas instituições que realizam pesquisas envolvendo seres humanos no Brasil, criado para defender os interesses dos sujeitos da pesquisa em sua integridade e dignidade para contribuir no desenvolvimento da pesquisa dentro de padrões éticos. O CEP é responsável pela avaliação e acompanhamento dos aspectos éticos de todas as pesquisas envolvendo seres humanos.

---

(assinatura do pesquisador)

**CONSENTIMENTO DA PARTICIPAÇÃO DA PESSOA COMO VOLUNTÁRIO (A)**

Eu, \_\_\_\_\_, CPF \_\_\_\_\_, abaixo assinado, após a leitura deste documento e de ter tido a oportunidade de conversar e ter esclarecido as minhas dúvidas com o pesquisador responsável, concordo em participar do estudo **LÓGICA DE PROGRAMAÇÃO NO ENSINO MÉDIO INTEGRADO: APLICAÇÃO E AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL NO CURSO TÉCNICO EM INFORMÁTICA** como voluntário (a). Fui devidamente informado (a) e esclarecido (a) pelo(a) pesquisador (a) sobre a pesquisa, os procedimentos nela envolvidos, assim como os possíveis riscos e benefícios decorrentes de minha participação. Foi-me garantido que posso retirar o meu consentimento a qualquer momento, sem que isto leve a qualquer penalidade (ou interrupção de meu acompanhamento/ assistência/tratamento).

Local e data \_\_\_\_\_

Assinatura do participante: \_\_\_\_\_

**Presenciamos a solicitação de consentimento, esclarecimentos sobre a pesquisa**

**e o aceite do voluntário em participar.** (02 testemunhas não ligadas à equipe de pesquisadores):

NOME:	NOME:
ASSINATURA:	ASSINATURA:

**APÊNDICE L - TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA PAIS OU RESPONSÁVEIS**

**TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO PARA PAIS OU RESPONSÁVEIS**

Convidamos o (a) Sr.(a) para permitir que a pessoa, a qual esteja sob sua responsabilidade, participe como voluntário (a), da pesquisa **LÓGICA DE PROGRAMAÇÃO NO ENSINO MÉDIO INTEGRADO: APLICAÇÃO E AVALIAÇÃO**

**DO PENSAMENTO COMPUTACIONAL NO CURSO TÉCNICO EM INFORMÁTICA,** que está sob a responsabilidade do pesquisador José Walter Paulino Júnior e está sob a orientação de Francisco Kelsen de Oliveira.

Este Termo de Consentimento pode conter informações que o/a senhor/a não entenda. Caso haja alguma dúvida, pergunte à pessoa que está lhe entregando o termo para que o/a senhor/a esteja bem esclarecido (a) sobre essa pesquisa. Após ser esclarecido (a) sobre as informações a seguir, no caso de permitir a participação na pesquisa, rubrique as folhas e assine ao final deste documento, que está em duas vias. Uma delas é sua e a outra é do pesquisador responsável. Em caso de recusa o/a senhor/a não será penalizado (a) de forma alguma. O (a) Senhor (a) tem o direito de retirar sua permissão sobre participação na pesquisa referente à pessoa que está sob sua responsabilidade em qualquer tempo, sem qualquer penalidade.

### **INFORMAÇÕES SOBRE A PESQUISA:**

#### **Participação na pesquisa:**

O objetivo deste estudo é analisar a aplicação das habilidades do Pensamento Computacional como alternativa metodológica ao ensino tradicional no processo de aprendizagem de Lógica de Programação no curso técnico em Informática Integrado ao Ensino Médio da EEEP Irmã Ana Zélia da Fonseca. Para isso será realizada uma intervenção em sala de aula onde os participantes irão assistir um curso sobre Pensamento Computacional ministrado por este pesquisador. A intervenção será realizada na disciplina de Programação Web, tendo sua duração de aproximadamente 3 semanas. Seguido da aplicação de exercícios sobre algoritmos utilizando como estratégia de resolução a metodologia de Pensamento Computacional e por fim a aplicação de 02 questionários via formulários do Google para avaliar o impacto e a percepção dos estudantes sobre a intervenção que foi realizada. Serão coletadas informações dos estudantes a respeito da metodologia e grau de dificuldade das soluções. Não haverá nenhuma outra forma de envolvimento ou comprometimento neste estudo. Em caso de gravação, filmagem ou fotos, a realização desses procedimentos será previamente avisada.

#### **Local da pesquisa:**

A pesquisa será realizada presencialmente na EEEP Irmã Ana Zélia da Fonseca, localizada na Rua Professora Maria de Fátima Tavares Leite, S/N no Bairro Eucaliptos, Milagres-CE, a durante as aulas das disciplinas de Programação Web, Programação Orientada a Objetos e Banco de Dados em turmas de primeiro e segundo ano de nível médio da modalidade integrada ao técnico em Informática.

**Benefícios e riscos decorrentes da Participação na pesquisa:**

A pesquisa é considerada segura, mas é possível que ocorram riscos mínimos previsíveis. Os riscos previsíveis nesta pesquisa podem envolver aspectos psicológicos, com a exposição a situações de confronto no qual possam encontrar dificuldades em resolver problemas que necessitam de uma base matemática, mobilização de emoções, sentimentos, experiências de decisão, frustração e conflitos de percepção que podem gerar um nível de estresse baixo. É previsto que a duração do risco seja transitória e de nível mínimo, haja vista que o desconforto psicológico que pode ser causado pela intervenção na pesquisa, são equiparados ao observado/relatado em situações reais do cotidiano, cabendo frisar o cuidado e atenção maior por se tratar de uma população vulnerável ao sofrimento psíquico e em uma fase crítica do desenvolvimento biopsicossocial.

Fica também garantida a indenização em casos de danos, comprovadamente decorrentes da participação na pesquisa, conforme decisão judicial ou extrajudicial.

Os benefícios esperados para os participantes devem ser transitórios e também permanentes, sendo estimados impactos positivos em termos educacionais, como o desenvolvimento da habilidade de resolução de problemas, autonomia, aumento da motivação, interesse, o que deve impactar os participantes a curto, médio e longo prazo.

Benefícios estes que proporcionam o desenvolvimento de habilidades essenciais para os profissionais do século XXI, como trabalho em grupo, pensamento crítico e reflexivo, resolução de problemas e comunicação. Além disso, espera-se que o produto educacional proveniente deste estudo contribua para aprimorar o processo de ensino de lógica de programação através do Pensamento Computacional no curso técnico em Informática da EEEP Irmã Ana Zélia da Fonseca e na Educação Profissional e Tecnológica como um todo.

**Autonomia e sigilo do participante da pesquisa:**



A participação da pessoa, a qual esteja sob sua responsabilidade, é voluntária. Em caso de recusa o(a) senhor(a) não será penalizado(a) de forma alguma. O(a) senhor poderá retirar sua permissão sobre participação na pesquisa referente à pessoa que está sob sua responsabilidade a qualquer momento desta pesquisa, bastando para isso entrar em contato com a pesquisador responsável. Os dados do participante serão manuseados somente pelo pesquisador e pelo orientador e não será permitido o acesso a outras pessoas. Os resultados deste trabalho poderão ser apresentados em encontros ou revistas científicas, entretanto, ele mostrará apenas os resultados obtidos como um todo, sem revelar o nome ou qualquer informação que esteja relacionada com a privacidade da pessoa que está sob sua responsabilidade.

Os dados coletados nesta pesquisa serão apenas as respostas do tipo texto, por meio da aplicação dos formulários e ficarão armazenados em pasta de arquivo em drive, sob a responsabilidade do pesquisador e do orientador. O(a) senhor(a) não pagará nada para permitir a participação nessa pesquisa. Se houver necessidade, as despesas para a participação da pessoa que está sob sua responsabilidade serão assumidas pelos pesquisadores (ressarcimento de transporte e alimentação).

Em caso de dúvidas relacionadas aos aspectos éticos deste estudo, você poderá consultar o Comitê de Ética em Pesquisa Envolvendo Seres Humanos do IF SERTÃO-PE no endereço: Reitoria – Rua Aristarco Lopes, 240, Centro, CEP 56.302-100, Petrolina-PE, Telefone: (87) 2101-2350, Ramal 2364, <http://www.ifsertao-pe.edu.br/index.php/comite-de-etica-em-pesquisa>, [cep@ifsertao-pe.edu.br](mailto:cep@ifsertao-pe.edu.br); ou poderá consultar a Comissão nacional de Ética em Pesquisa, Telefone (61)3315-5878, [conep.cep@saude.gov.br](mailto:conep.cep@saude.gov.br).

O Comitê de Ética em Pesquisa (CEP) é um colegiado interdisciplinar e independente, que deve existir nas instituições que realizam pesquisas envolvendo seres humanos no Brasil, criado para defender os interesses dos sujeitos da pesquisa em sua integridade e dignidade para contribuir no desenvolvimento da pesquisa dentro de padrões éticos. O CEP é responsável pela avaliação e acompanhamento dos aspectos éticos de todas as pesquisas envolvendo seres humanos.

---

Assinatura do pesquisador (a)

**CONSENTIMENTO DA PARTICIPAÇÃO DA PESSOA COMO VOLUNTÁRIO (A)**

Eu, \_\_\_\_\_, CPF \_\_\_\_\_, abaixo assinado pelo meu representante legal, após a escuta da leitura deste documento e ter tido a oportunidade de conversar e esclarecido as minhas dúvidas com o pesquisador responsável, concordo em participar do estudo **LÓGICA DE PROGRAMAÇÃO NO ENSINO MÉDIO INTEGRADO: APLICAÇÃO E AVALIAÇÃO DO PENSAMENTO COMPUTACIONAL NO CURSO TÉCNICO EM INFORMÁTICA**, como voluntário (a). Fui devidamente informado (a) e esclarecido (a) pelo (a) pesquisador (a) sobre a pesquisa, os procedimentos nela envolvidos, assim como os possíveis riscos e benefícios decorrentes de minha participação. Foi-me garantido que posso retirar meu consentimento a qualquer momento, sem que isto leve a qualquer penalidade (ou interrupção de meu acompanhamento/assistência/tratamento).

A rogo de \_\_\_\_\_, que é (não alfabetizado/juridicamente incapaz/ deficiente visual), eu \_\_\_\_\_ assino o presente documento que autoriza a sua participação neste estudo.

Local e data \_\_\_\_\_

Assinatura do (da) responsável: \_\_\_\_\_

## Apêndice M - Revisão Sistemática de Literatura 01



This work is licensed under a  
Creative Commons Attribution 4.0 Unported License.

DOI: 10.31416/rsdv.v12i1.394

### **Pensamento Computacional como metodologia para o ensino de Lógica de Programação**

*Computational Thinking as a methodology for teaching Programming Logic*

PAULINO JÚNIOR, José Walter. Especialista em Redes de Computadores  
Instituto Federal de Educação, Ciências e Tecnologia do Sertão Pernambucano-Campus Salgueiro. Endereço da  
instituição - Salgueiro - Pernambuco - Brasil. CEP: 56.000-000 / Telefone: (88) 99811.8497/ E-mail:  
[jose.walter@aluno.ifsertao-pe.edu.br](mailto:jose.walter@aluno.ifsertao-pe.edu.br), [orcid.org/0000-0003-1587-9858](https://orcid.org/0000-0003-1587-9858)

**OLIVEIRA, Francisco Kelsen. Doutor em Ciência da Computação**

Instituto Federal de Educação, Ciências e Tecnologia do Sertão Pernambucano-Campus Salgueiro. Endereço da instituição - Salgueiro - Pernambuco - Brasil. CEP: 56.000-000 / Telefone: (85) 98817.8151 / E-mail: [francisco.oliveira@ifsertao-pe.edu.br](mailto:francisco.oliveira@ifsertao-pe.edu.br), [orcid.org/0000-0002-7382-3206](https://orcid.org/0000-0002-7382-3206)

## RESUMO

Este trabalho objetivou realizar uma Revisão Sistemática de Literatura (RSL) para conhecer as experiências de uso e aplicação do Pensamento Computacional em contexto internacional. Para tanto, foi realizada a RSL utilizando-se da metodologia PRISMA analisando os principais trabalhos que abordavam o tema dispostos em publicações disponíveis nas bases Science Direct, Scopus, SBC Open Lib e Google Scholar. Os resultados dos trabalhos mostraram que a ferramenta Scratch aliada a metodologias alternativas a exemplo da Inteligência Artificial e Realidade Virtual estão sendo difundidas para apresentar os conceitos de Pensamento Computacional. A utilização de tecnologias alternativas como Inteligência Artificial, Jogos Digitais e Realidade Virtual como incremento ao processo de ensino também aparece como um mecanismo que motiva para o desafio de recompensa, porém, é verificado que apesar da utilização de metodologias alternativas, a aplicação ainda é deveras superficial, necessitando de um aprofundamento. Foram analisados 16 artigos, com o objetivo de apresentar uma visão sobre a área de pesquisa e descrever quais as linhas e resultados dos pesquisadores a partir de trabalhos publicados no idioma inglês.

**Palavras-chave:** Lógica de Programação; Pensamento Computacional; Aprendizagem.

## ABSTRACT

This study aimed to carry out a Systematic Literature Review (SLR) to learn about the experiences of using and applying Computational Thinking in an international context. Therefore, the RSL was carried out using the PRISMA methodology, analyzing the main works that addressed the subject arranged in publications available in the Science Direct, Scopus, SBC Open Lib and Google Scholar databases. The results of the work showed that the Scratch tool combined with alternative methodologies such as Artificial Intelligence and Virtual Reality are being disseminated to present the concepts of Computational Thinking. The use of alternative technologies such as Artificial Intelligence, Digital Games and Virtual Reality as an increment to the teaching process also appears as a mechanism that motivates the reward challenge, however, it appears that despite the use of alternative methodologies, the application is still very difficult. superficial, in need of deepening. Sixteen articles were analyzed, with the aim of presenting an overview of the research area and describing the researchers' statements and results from works published in the English language.

**Keywords:** Programming Logic; Computational Thinking; Learning.

## Introdução

Conhecer sobre Computação nos dias atuais é tão fundamental quanto os saberes básicos da Matemática, Filosofia e Física, por exemplo, assim como pensar,

abstrair, contar e relacionar, estando presente em todos os aspectos da sociedade moderna, inclusive, na solução de problemas complexos. Os gargalos do ensino e aprendizagem de lógica de programação têm sido discutidos no contexto acadêmico e científico, tais como dificuldade em aprendizagem por parte dos alunos seguido de evasões. Técnicas e ferramentas vêm sendo propostas para minimizar as dificuldades de quem é iniciante no estudo.

O Pensamento Computacional, por outro lado, introduz uma nova abordagem para a área da Ciência do Pensamento e Ciência da Computação (PC), pois parte da ideia de que seja possível utilizar-se de conceitos da própria área da Computação na educação básica, aperfeiçoando o desenvolvimento de habilidades como a abstração, no qual ajuda crianças e adolescentes em seu processo de construção criativa e de resolução de problemas em todas as áreas da vida. Desse modo, pode-se considerar o PC como um aliado a futuros profissionais da sociedade digital.

Estudos que estabelecem relações com PC ganharam destaque com as pesquisas de Wing (2006) e de outros autores apresentados na próxima seção deste trabalho, que apresentam o PC como metodologia de impacto dentro do ensino de computação e com aplicações no ensino de Lógica de Programação.

É preciso ressaltar que pensar computacionalmente não é nenhum sinônimo de construir aplicações por meio da programação, no entanto Blikstein (2008) define que uma das etapas fundamentais do PC é saber programar um computador para a realização de tarefas cognitivas de maneira automatizada para que esse conhecimento seja um tipo de suporte ao raciocínio humano. De Paula, Valente e Burn (2014) afirmam que o PC é uma maneira de se pensar e de analisar uma determinada situação ou artefato, independente de tecnologia, porém defendendo a utilização de ideias teóricas que são tidas como a base podem ser aliadas a aspectos práticos da programação.

Diante do que foi exposto, este trabalho é uma complementação ao trabalho PAULINO JÚNIOR e OLIVEIRA (2022) que se limitava a realizar uma revisão de literaturas apenas em pesquisas nacionais se fazendo necessário e relevante desenvolver uma revisão sistemática na qual seja possível indicar as pesquisas estrangeiras no idioma inglês que avançam na utilização prática dos conceitos de PC voltadas ao ensino de programação.

Desse modo, o presente trabalho tem como seu objetivo principal, realizar uma Revisão Sistemática de Literatura (RSL) para verificar e levantar pesquisas relacionadas à aplicação do PC ao ensino de Lógica de Programação que abordem aspectos relacionados que envolvam ou estimulem o PC objetivando o ensino de programação de computadores ou direcionados a práticas educacionais que utilizem ferramentas e métodos que se utilizem de conceitos de programação.

Este trabalho detalha resultados obtidos através de uma Revisão Sistemática de Literatura (RSL) a partir de trabalhos com o idioma em inglês, apresentando seu processo de desenvolvimento, mostrando as regras e procedimentos adotados durante a pesquisa até chegar nos resultados finais. O presente referencial teórico abordará conceitos sobre a educação profissional e tecnológica, ensino de computação e pensamento computacional.

## **2 Referencial Teórico**

O presente referencial teórico abordará conceitos sobre a Educação Profissional e Tecnológica fazendo uma breve contextualização histórica em seguida

trabalhando alguns dos seus fundamentos a partir de visões de autores referências no campo de estudo. Em sequência os tópicos seguintes tratarão do ensino de computação fazendo uma associação com o pensamento computacional e seus principais estudiosos.

## 2.1 Ensino de Computação

A utilização da programação como instrumento pedagógico e didático favorece a aprendizagem ativa, uma vez que a interação estabelecida entre as ações dos estudantes e o feedback do programa favorece a dinâmica. Na perspectiva de Valente (1999) no qual trata da formação de professores para a área da informática na educação, objetiva preparar os participantes para o uso do computador em sala de aula, versando em atividades que usam o computador como mecanismo para que os alunos possam construir seu conhecimento. Almeida (2009) afirma que o aluno sintetiza os passos necessários para atingir determinado objetivo, sendo direcionado a transformar seus conhecimentos em procedimentos. O ensino dos princípios da computação proporciona o desenvolvimento do pensamento computacional, sendo uma habilidade fundamental para todos, não só de cientistas da computação. Os autores citados discutem e conceituam a tecnologia e propõe a ideia de que ela possa ser uma vertente aliada ao ensino de forma alternativa às metodologias tradicionais, dado o contexto em que estamos vivendo, onde as transformações são constantes e em tempo real se faz necessário que sua aplicação esteja bem fundamentada no ambiente escolar, porém, apenas utilizá-la como fim não é o suficiente para um bom desenvolvimento, pois precisa fazer sentido ao alunado. O ensino de computação as vezes é confundido com as aulas de Informática, que tem em seu propósito instruir o aluno quanto ao uso de softwares, restringindo o ensino de computação ao simples manuseio de aplicativos e equipamentos eletrônicos. Desse modo, a utilização do Pensamento Computacional na educação básica não visa apenas a empregabilidade e desenvolvimento econômico, mas também a construção de habilidades. A seção seguinte irá tratar do pensamento computacional a partir dos seus fundamentos, entre algumas citações, estarão as de Papert e Wing, considerados como pioneiros nos estudos do Pensamento Computacional.

## 2.2 Pensamento Computacional

O termo introduzido em Wing (2006) é um método para ensinar a solucionar problemas, conceber sistemas e compreender o comportamento humano, por meio de conceitos da Ciência da Computação ou, como colocado por Paula, Valente e Burn (2014), "É um modo de pensar e analisar de forma específica uma situação ou artefato, independente de tecnologia".

Assim, desenvolver habilidades em PC não significa necessariamente aprender a programar (ZANETTI; BORGES; RICARTE, 2016.) Wing (2006) afirma que o pensamento computacional (PC) deve ser uma habilidade inerente e básica a ser ensinada a crianças, assim como ler ou escrever, por exemplo. Alguns currículos, enquanto componente extracurricular já trazem em suas práticas o ensino de programação tanto no ensino fundamental quanto no ensino médio. Práticas como estas podem levar o aluno ao desenvolvimento de habilidades de abstração e conseqüentemente a resolução de problemas, habilidades muito importantes no atual contexto tecnológico que vivemos.

Desse modo, começar a utilizar o PC como metodologia de ensino poderá impactar positivamente na construção desses saberes. Nunes (2011) afirma ainda que o pensamento computacional pode ser entendido como o processo sistematizado das fases da resolução de problemas, podendo ser aplicado não só na Ciência da Computação, como também nas demais áreas. Já na perspectiva Lee (2014), o pensamento computacional atribui-se ao raciocínio lógico, incluindo características como: formulação de problemas, representações abstratas, organização e análise lógica de dados, identificação, análise e implementação de soluções para resolução de problemas.

## Material e métodos

A presente revisão bibliográfica foi realizada a partir da metodologia PRISMA, do inglês *Preferred Reporting Items for Systematic Re-views and Metaanalyses*, tendo como base os trabalhos de Moher D et al (2015). A metodologia objetiva ajudar autores a melhorarem a construção de revisões sistemáticas e meta-análises, também podendo ser utilizada como base para relatos de outras revisões.

Este trabalho seguiu os preceitos metodológicos dos trabalhos de Silva e Oliveira (2023; 2022; 2021), Araújo e Oliveira (2022; 2021), Paulino Júnior e Oliveira (2022), Rocha e Oliveira (2020), Costa e Oliveira (2020), e Oliveira e Gomes (2016). No processo de construção desta Revisão Sistemática de Literatura (RSL) foram especificadas as palavras-chaves "*computational thinking*", "*programming logic*", "*systematic literature review*", "*Scratch*" e "*experiments*" que posteriormente se tornaram as *strings* de buscas.

Após a escolha das principais palavras-chaves, foram elencadas a estrutura de pesquisa nas plataformas SBC Lib Online, Science Direct, Scopus e Google Acadêmico utilizando a ferramenta de revisão sistemática StArt no qual buscou-se apenas trabalhos no idioma inglês a partir de 2020 em anais de eventos e periódicos, com os resultados apresentados no quadro 1 abaixo:

**Quadro 1- Palavras-chaves**

Strings de busca	SBC Open Lib	Google	Scopus	Science Direct
"systematic literature review" + "computational thinking" + "experiments" + "programming logic"	10	32	2.417	23
"systematic literature review" + "computational thinking" + "experiments" + "Scratch"	10	353	2.393	12

Fonte: adaptada de ARAÚJO e OLIVEIRA (2021).

Finalizada as consultas com base nas palavras-chaves e combinações apresentadas foram localizados os trabalhos relacionados expostos no quadro 2 abaixo:

**Quadro 2 - Resultados de buscas por strings**

Palavra-chave	Google Acadêmico	SBC Open Lib	Scopus	Science Direct

Computational thinking	21.900	217	9.402	61.000
Programming logic	19.600	15	6.809	65.000
Scratch	69.900	26	121	61.000

Fonte: Pesquisa direta.

### Critérios de Inclusão e Exclusão

O critério de inclusão definido a partir da capacidade do mesmo em solucionar pelo menos uma das questões da pesquisa, após esse primeiro critério foram definidos outros critérios como mostra o quadro 3 abaixo. Os critérios de exclusão definem diretrizes para excluir aqueles que não se enquadram totalmente no contexto da pesquisa.

#### Quadro 3- Critérios de Inclusão e Exclusão

Critérios	D	Descrição
Inclusão	C1	Artigos resumidos ou completos publicados em periódicos científicos das bases listadas.
	C2	Apresentam experiências positivas e negativas no ensino de Scratch
	C3	Artigos que abordam sobre o uso com finalidade educacional da ferramenta Scratch no ensino de lógica/linguagem de programação.
Exclusão	E1	Artigos duplicados ou semelhantes.
	E2	Artigos apenas com caráter publicitário ou de marketing publicados em magazines.
	E3	Apresentação de slides e Livros
	E4	Artigos que aparentemente não contribuem para a pesquisa vigente.

Fonte: Pesquisa direta.

O objetivo da RSL é verificar e avaliar contribuições de estudos publicados e disponibilizados nas bases de dados SBC Lib Online, Science Direct, Scopus e Google Acadêmico, para tanto, apresentam suas contribuições no ensino de programação de computadores e pensamento computacional. Desse modo, pretende-se responder a seguinte questão base para esta pesquisa: "No contexto do ensino de programação utilizando o Pensamento Computacional, o que se tem produzido nas pesquisas estrangeiras no idioma inglês?" Com base nessa questão norteadora, outras três questões de pesquisa mais bem definidas e específicas foram levantadas na quadro 4 abaixo:

#### Quadro 4- Questões da pesquisa.

ID	Perguntas	Motivação
P01	Quais vantagens ou desvantagens são observadas pelos usuários no uso educacional das ferramentas e metodologias?	Entender o ambiente e contexto em que a ferramenta é utilizada para o ensino de LP bem como conhecer as lições aprendidas a partir das experiências que não surtiram o devido efeito.



P02	Conhecer como o Pensamento Computacional pode colaborar com a educação?	Em meio a um universo de experiências exitosas, retirar aquelas que alcançaram resultados modestos ou insatisfatórios de acordo com os objetivos elencados.
P03	Como a ferramenta Scratch pode colaborar com o ensino de Lógica de Programação a partir dos conceitos de Pensamento Computacional?	Identificar as colaborações e práticas com o uso de Scratch aplicadas ao ensino de LP com a utilização de PC

Fonte: Pesquisa direta.

Os artigos selecionados estão de acordo com atendimento aos critérios de inclusão, a partir da solução das perguntas definidas por meio da leitura do resumo e do título do trabalho. A busca pelos artigos englobou as bases de dados citadas, sendo selecionados trabalhos publicados em eventos tais como simpósio e congressos, a exemplo do Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Educação em Informática (WEI) dentre outros a partir de 2005. Não foram selecionados artigos que apenas realizaram uma revisão da literatura, sendo considerados aqueles que descrevem e/ou aplicam práticas pedagógicas e a utilização de ferramentas educacionais. O quadro 5 abaixo apresenta e organiza os artigos selecionados por um identificador único (ID), título e o autor.

#### Quadro 5- Artigos catalogados

ID	Autor	Contribuições
A01	MARTÍNEZ-COMESAÑA, Miguel et al (2023)	Os autores buscam analisar de forma detalhada dos efeitos da inteligência artificial (IA) nos processos de avaliação educacional em níveis primário e secundário, fornecendo uma visão abrangente das mudanças que a inteligência artificial está trazendo para os métodos de avaliação na educação básica, apresentando aos leitores uma compreensão mais aprofundada do tópico e suas implicações para o sistema educacional.
A02	JING, Sifeng et al (2020)	O estudo propõe um modelo que leva em consideração não apenas os processos de pensamento e aprendizado do aluno (aspectos cognitivos), mas também sua capacidade de monitorar, avaliar e regular seu próprio aprendizado (aspectos metacognitivos) por meio da utilização do Scratch. A obra visa fornecer uma abordagem abrangente e eficaz para o ensino de programação com Scratch, considerando tanto os aspectos cognitivos quanto os metacognitivos do aprendizado do aluno. Permitindo que os educadores aprimorem suas estratégias de ensino e os estudantes desenvolvam habilidades de autorregulação, resultando em um processo de aprendizagem mais eficiente e eficaz.
A03	FLYNN, Rachel M. et al (2021)	Os autores analisam diversas pesquisas anteriores que investigaram como os jogos digitais afetam o desenvolvimento cognitivo e a capacidade de aprendizagem em crianças na faixa etária da infância média. Essa revisão narrativa visa fornecer uma visão abrangente das abordagens de pesquisa utilizadas na área, destacando as tendências e questões importantes para orientar futuros estudos sobre o assunto. Ao reunir e analisar as pesquisas

		existentes, os autores pretendem contribuir para o desenvolvimento de uma base sólida de conhecimentos sobre os efeitos dos jogos digitais no processo de aprendizagem e no desenvolvimento cognitivo das crianças durante a infância média.
A04	SUN, Liping et al (2023)	O estudo examina diversas fontes acadêmicas, incluindo artigos científicos e publicações relevantes, para analisar como os professores oferecem suporte e orientação aos alunos durante a utilização de jogos como ferramentas educacionais no ambiente da educação primária. Ao reunir e analisar as evidências existentes, a obra busca oferecer insights importantes para educadores interessados em integrar o <i>game-based learning</i> em suas práticas de ensino, destacando a importância do papel do professor na facilitação do processo de aprendizagem nesse contexto específico.
A05	ALE; STURDEE; RUBEGNI (2022).	O estudo examina uma ampla variedade de fontes, incluindo artigos científicos e publicações relevantes, para investigar como a cognição incorporada é aplicada no design e desenvolvimento de interfaces e tecnologias de computação voltadas para crianças.
A06	CHEN, Xieling et al (2020).	O estudo investiga a evolução do uso da inteligência artificial no campo da educação e destaca as áreas onde ainda há lacunas significativas tanto na aplicação prática quanto na base teórica. A obra oferece uma avaliação crítica do estágio atual da inteligência artificial na educação, identificando tanto as oportunidades quanto os desafios que precisam ser abordados para promover seu uso efetivo e benéfico no contexto educacional. Ao explorar as lacunas na aplicação e teoria, os autores buscam estimular a pesquisa e a inovação no campo, visando uma melhor integração e aproveitamento da inteligência artificial como uma ferramenta poderosa de aprimoramento da educação.
A07	SHARMA; PAPAVLASO POULOU; GIANNAKOS (2022)	É um estudo que explora as expressões faciais das crianças durante a codificação colaborativa e compara o desempenho objetivo com o desempenho subjetivo. A obra investiga as reações emocionais e comportamentais das crianças enquanto participam de atividades de codificação em colaboração, e como essas expressões se relacionam com as avaliações objetivas de desempenho, feitas por observadores externos, e as avaliações subjetivas, relatadas pelas próprias crianças.
A08	GEROSA, Anaclara et al (2021).	O estudo investiga as habilidades cognitivas e o pensamento computacional em crianças de 5 anos, buscando evidências de associações entre essas habilidades e a capacidade de sequenciamento e comparação de números simbólicos.
A09	RADIANTI, Jaziar et al (2020)	A obra examina como a realidade virtual imersiva está sendo aplicada no ensino superior. Ao analisar uma variedade de estudos, a revisão sistemática busca fornecer insights sobre os elementos de design utilizados nessas aplicações e as lições aprendidas com suas implementações. Além disso, o estudo oferece sugestões para futuras pesquisas a fim de aprimorar a utilização da realidade virtual imersiva como uma ferramenta efetiva no ensino superior.
A10	ZHAO; LLORENTE;	A obra oferece uma visão abrangente da compreensão e aplicação da competência digital no campo do ensino superior. Ao analisar

	GÓMEZ (2021).	uma variedade de estudos, a revisão sistemática visa identificar tendências, desafios e oportunidades para promover a integração eficaz da competência digital na pesquisa e prática pedagógica em instituições de ensino superior.
A11	IVANOVIĆ, Mirjana et al (2019)	a obra examina as experiências e perspectivas do uso de tecnologia no ensino superior na Sérvia. Ao investigar as percepções dos envolvidos, o estudo busca fornecer insights sobre como a tecnologia pode influenciar o ensino e aprendizagem nesse contexto específico, destacando oportunidades e desafios para o aprimoramento da educação superior com o uso das tecnologias educacionais.
A12	ZHARYLGAS SOVA, Peruza et al (2021)	Esse trabalho apresentou a experiência de um projeto implementado em uma turma de matemática, com atividades usando o Scratch e abordando os conceitos do Pensamento Computacional. Foi realizado um estudo de caso na disciplina de Matemática do 9º ano do Ensino Fundamental, onde foi possível observar que os alunos construíram uma relação entre os conhecimentos prévios da disciplina e os que foram apresentados por meio do Scratch, tornando assim o aprendizado mais significativo.
A13	ZARESTKY, Jill et al (2022)	Explora como a escrita reflexiva pode apoiar a metacognição e a autorregulação em estudantes de pós-graduação em Ciência e Engenharia Computacional. investiga como a prática da escrita reflexiva pode ajudar os estudantes a desenvolverem habilidades metacognitivas, que envolvem a capacidade de pensar sobre seu próprio processo de aprendizagem, e aprimorar a autorregulação, que diz respeito à capacidade de controlar e ajustar suas estratégias de aprendizagem.
A14	GUGGEMOS ; MOSER; SEUFERT (2022).	aborda o fenômeno dos MOOCs (Cursos Online Abertos e Massivos) para estudantes do ensino fundamental e médio (K-12) no contexto da alfabetização informacional. investiga o uso de MOOCs no ensino fundamental e médio, analisando sua influência na alfabetização informacional dos alunos. Ao explorar as percepções e resultados, o estudo busca compreender o papel dos MOOCs nesse contexto educacional específico, fornecendo insights importantes para educadores e formuladores de políticas em relação ao uso efetivo desses cursos para aprimorar a alfabetização informacional dos alunos no ensino K-12.
A15	JING, Sifeng et al (2020)	O estudo propõe um modelo que considera não apenas os processos cognitivos de aprendizagem dos alunos, mas também sua capacidade de monitorar, avaliar e regular seu próprio aprendizado (metacognição). Esse modelo é aplicado em projetos de programação na plataforma Scratch, onde os alunos podem aprender a programar e criar projetos interativos.
A16	Gobbi e Silveira (2020).	Os resultados obtidos neste trabalho mostraram que independentemente de quais mecanismo tecnológicos disponíveis na escola, se faz necessário realizar atividades que estimulem o desenvolvimento de habilidades relacionadas ao pensamento computacional. Além disso, a inserção do pensamento computacional contribui para melhoria no desempenho dos alunos

	na resolução de problemas matemáticos.
--	--

Fonte: Pesquisa direta.

## Resultados e discussão

Como foi apresentado na seção anterior, o número de pesquisas que abordam o tema pensamento computacional é deveras crescente. O ensino de programação tem crescido bastante nos últimos tempos, dado avanço da tecnologia e do digital cada vez mais presente no nosso cotidiano. Os trabalhos foram categorizados por práticas, a fim de se descobrir as categorias de aplicação que estão sendo utilizadas. Foram definidas cinco categorias, tendo como objetivo propor um modelo para futuros pesquisadores que queiram se relacionar com outras metodologias e práticas dentro do contexto de computação. Após a leitura dos trabalhos foi identificado as seguintes categorias: Linguagens de Programação (LP), Jogos Digitais (JD) e Computação Desplugada (CD). No quadro 6 a seguir, é apresentada uma breve descrição sobre cada categoria.

### Quadro 6- Categorização

Categoria	Descrição
Linguagens de Programação Associadas ao Scratch (LD)	Utilização de linguagens de programação como prática de ensino, exemplo: Python, Java, C, etc.
Tecnologias Associadas (CD)	Métodos que utilizam a tecnologia como mecanismo de ensino: realidade virtual, realidade aumentada, scratch e inteligência artificial.
Jogos Digitais (JD)	Utilização de jogos digitais como mecanismo de ensino de computação.

Fonte: Pesquisa direta.

De modo a relacionar os trabalhos com as categorias selecionadas, foi elaborado um esquema que representa a associação entre ambas no quadro 7 abaixo:

### Quadro 7- Práticas relacionadas

Práticas	Trabalhos (ID)
Linguagens de Programação (LD)	A15, A03
Tecnologias Associadas ao Pensamento Computacional (CD)	A01, A02, A05, A06, A07, A08, A09, A10, A11, A13, A14, A15, A16
Jogos Digitais (JD)	A03, A04

Fonte: Pesquisa direta.

Após o levantamento, foi observado uma diversidade nos estudos. Não há um padrão, 13 trabalhos foram voltados para a utilização de tecnologias associadas ao Pensamento Computacional na educação, dois trabalhos se utilizam de conceitos de Lógica de Programação com Scratch e dois trabalhos abordaram Jogos Digitais como alternativa. Pela análise dos dados, observou-se uma tendência de ensino no qual o

Pensamento Computacional é amplamente utilizado a partir de outras metodologias/tecnologias.

Todos os trabalhos em seus resultados apontaram para uma aprendizagem significativa por meio dos conceitos básicos de computação (especialmente associados ao Pensamento Computacional através de metodologias alternativas) de forma divertida, impactando diretamente no desenvolvimento do interesse e motivação dos alunos. A seguir será apresentado a relação entre as questões de pesquisa e as abordagens de alguns trabalhos selecionados, os mais relevantes

**P01: Quais vantagens ou desvantagens são observadas pelos usuários no uso educacional das ferramentas e metodologias associadas?**

Esta questão buscou verificar se os trabalhos em análise entregam vantagens reais e práticas em sua aplicação. Após o levantamento foi possível perceber que a maioria dos estudos apresentam resultados positivos em relação ao seu contexto de aprendizagem. Em [A01], no contexto da educação primária e secundária, os métodos de avaliação são cruciais para medir o desempenho dos estudantes e melhorar a qualidade do ensino. A inteligência artificial tem sido cada vez mais incorporada nesses processos, e a obra explora como essa tecnologia está impactando a maneira como os alunos são avaliados.

Em A02 o estudo constatou que ao utilizar o ambiente de programação Scratch como cenário de aplicação, os autores mostram como o modelo pode ser implementado e como ele ajuda a melhorar a eficácia do ensino de programação.

Em A04 os autores apresentam a partir de uma revisão de literatura a perspectiva de que os jogos digitais são bem aplicados e entregam melhores resultados quando os participantes, professores e alunos já estão familiarizados com outros tipos de jogos, sendo um critério importante para o desenvolvimento do conteúdo. Nos respectivos trabalhos A01, A02, A05, A06, A07, A08, A09, A10, A11, A13, A14, A15, A16 em que tratam de metodologias alternativas utilizando o Pensamento Computacional observou-se que apesar de serem bem difundidas e aplicadas, ainda falta conhecimento de técnicas eficientes de como aplicar corretamente, a exemplo do trabalho A06 que aborda uma revisão de literatura sobre a utilização da Inteligência Artificial na educação, no qual foram estudados 45 artigos e constatado que a metodologia de Pensamento Computacional é bem aplicada, porém, observou-se que poucos trabalhos se aprofundam no tema, demonstrando um nível raso, e os que se aprofundam apresentam poucas referências, evidenciando assim uma necessidade de mais produções sobre o tema.

**P02: Conhecer como o Pensamento Computacional pode colaborar com o ensino de Lógica de Programação?**

Os trabalhos A03, A04 e A15 analisam e propõem modelos de ensino a partir de aspectos cognitivos e metacognitivos, aplicados em projetos de programação com Scratch e Pensamento Computacional. O estudo A015 propõe um modelo que considera não apenas os processos cognitivos de aprendizagem dos alunos, mas também sua capacidade de monitorar, avaliar e regular seu próprio aprendizado (metacognição). Esse modelo é aplicado em projetos de programação na plataforma Scratch, onde os alunos podem aprender a programar e criar projetos interativos. Ao utilizar esse modelo, os alunos são incentivados a desenvolver habilidades de autorregulação e aprimorar seu processo de aprendizagem,

resultando em um melhor desempenho e compreensão dos conceitos de programação na plataforma Scratch.

Apresentando aqui um viés que analisa também. Abordagem visual e intuitiva, blocos de código estruturados, pensamento sequencial, desenvolvimento do raciocínio lógico, fomento à criatividade, aprendizagem colaborativa e feedback instantâneo são características que estão presentes em todos os trabalhos, mesmo aqueles que analisam a perspectiva e encontraram lacunas, como em A06, no qual constatou-se que apesar das metodologias associadas ao Pensamento Computacional como a Inteligência Artificial estarem amplamente em utilização, ainda falta conhecer seus fundamentos e aplicá-las de modo mais específico e aprofundado.

### **P03: Como as metodologias associadas ao Scratch podem colaborar com o ensino de Lógica de Programação a partir dos conceitos de Pensamento Computacional?**

As metodologias associadas ao Scratch a exemplo dos presentes nos artigos analisados que são: Inteligência Artificial, Realidade Virtual e Jogos Digitais, além de oferecem uma abordagem visual, intuitiva e lúdica para o ensino de Lógica de Programação também relaciona a aspectos cognitivos e comportamentais, a partir do estudo dos trabalhos propostos. Observou-se também que existe uma preocupação quanto a metacognição, que é a habilidade de refletir sobre o próprio pensamento e aprendizado, permitindo aos indivíduos serem conscientes de suas habilidades cognitivas, monitorarem seu desempenho e tomarem decisões estratégicas para otimizar sua aprendizagem e resolução de problemas, o que se alinha perfeitamente aos conceitos e objetivos do Pensamento Computacional.

O trabalho A07 por exemplo, explora as expressões faciais das crianças durante a codificação colaborativa e compara o desempenho objetivo com o desempenho subjetivo, investigando as reações emocionais e comportamentais das crianças enquanto participam de atividades de programação em colaboração, e como essas expressões se relacionam com as avaliações objetivas de desempenho, feitas por observadores externos, e as avaliações subjetivas, relatadas pelas próprias crianças se utiliza ao permitir que os alunos experimentem, colaborem e desenvolvam projetos criativos, o Scratch promove o pensamento lógico e sequencial, o raciocínio abstrato e a resolução de problemas, tornando-se uma ferramenta eficaz no desenvolvimento de habilidades fundamentais para a programação e para a vida cotidiana.

### **Considerações finais**

De acordo com os resultados observados percebe-se que a quantidade de trabalhos relacionados ao Pensamento Computacional como alternativa de ensino não só de programação, mas também de tecnologias é deveras cresce e a adoção tem sido rápida. Muitos trabalhos além de abordarem o Pensamento Computacional em suas práticas, também aliam a metodologias alternativas a exemplo da Inteligência Artificial e Realidade Virtual como formas de ensino, contexto este, não verificado no trabalho de Paulino Júnior e Oliveira (2022) referente a revisão de literatura a partir de trabalhos nacionais.

Observou-se também uma preocupação dos autores com relação não apenas a forma como os conteúdos são trabalhados, selecionados e aplicados, mas também em aspectos cognitivos e comportamentais do público-alvo buscando evidências de associações entre essas habilidades e a capacidade de sequenciamento e comparação de números simbólicos, por exemplo. A presente RSL utilizou-se de um processo de seleção através de critérios de inclusão e exclusão, objetivando a coerência e fidelidade à temática proposta.

Os critérios de inclusão e exclusão foram cautelosamente discutidos e projetados para inviabilizar o risco de exclusão de trabalhos relevantes, a busca foi feita apenas em trabalhos no idioma em inglês. Uma limitação encontrada no trabalho anterior Paulino Júnior e Oliveira (2022) foi restringir apenas o Scratch como ferramenta de ensino de lógica de programação e pensamento computacional, desse modo, o presente trabalho verificou a aplicabilidade de outras estratégias tais como a utilização de Realidade Virtual, Inteligência Artificial e Jogos Digitais.

## Referências

ALE, Moyosore; STURDEE, Miriam; RUBEGNI, Elisa. A systematic survey on embodied cognition: 11 years of research in child-computer interaction. **International Journal of Child-Computer Interaction**, v. 33, p. 100478, 2022.

ARAÚJO, A. C. S.; OLIVEIRA, F. K. Análise comparativa das ferramentas de tradução português Libras: Comparative analysis of Portuguese/BSL translation technological tools. **Revista Semiárido De Visu**, [S. l.], v. 10, n. 3, 2022. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/397>. Acesso em: 01 jan. 2024.

ARAÚJO, A. C. S.; OLIVEIRA, F. K. de. Revisão Sistemática da Literatura sobre Tecnologias digitais de informação e comunicação de tradução do par linguístico Português Libras. **Revista Semiárido De Visu**, [S. l.], v. 9, n. 3, p. 286-299, 2021. DOI: 10.31416/rsdv.v9i3.305. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/305>. Acesso em: 01 jan. 2024.

CHEN, Xieling et al. Application and theory gaps during the rise of artificial intelligence in education. **Computers and Education: Artificial Intelligence**, v. 1, p. 100002, 2020.

COSTA, M. G. N. da .; OLIVEIRA, F. K. de. A prática docente de professores não licenciados no Ensino Médio Integrado à Educação Profissional e Tecnológica. **Revista Semiárido De Visu**, [S. l.], v. 8, n. 3, p. 512-523, 2020. DOI: 10.31416/rsdv.v8i3.36. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/36>. Acesso em: 01 jan. 2024.

FLYNN, Rachel M. et al. A narrative review of methods used to examine digital gaming impacts on learning and cognition during middle childhood. **International Journal of Child-Computer Interaction**, v. 30, p. 100325, 2021.

GEROSA, Anaclara et al. A simple reorganization of childcare centre with positive effects on self-regulation and play behaviour. 2021.

GUGGEMOS, Josef; MOSER, Luca; SEUFERT, Sabine. Learners don't know best: Shedding light on the phenomenon of the K-12 MOOC in the context of information literacy. **Computers & Education**, v. 188, p. 104552, 2022.

GOBBI, Rodrigo. Utilização do Scratch como ferramenta para apoiar o desenvolvimento do pensamento computacional. 2020.

JING, Sifeng et al. A learner model integrating cognitive and metacognitive and its application on scratch programming projects. **IFAC-PapersOnLine**, v. 53, n. 5, p. 644-649, 2020.

LAUPICHLER, Matthias Carl et al. Artificial intelligence literacy in higher and adult education: A scoping literature review. **Computers and Education: Artificial Intelligence**, p. 100101, 2022.

MARTÍNEZ-COMESAÑA, Miguel et al. Impact of artificial intelligence on assessment methods in primary and secondary education: systematic literature review. **Revista de Psicodidáctica (English ed.)**, 2023.

MOHAMMADI, Zahra; VALERO, Oscar. A new contribution to the fixed point theory in partial quasi-metric spaces and its applications to asymptotic complexity analysis of algorithms. **Topology and its Applications**, v. 203, p. 42-56, 2016.

OLIVEIRA, F. K.; GOMES, A. S. Revisão sistemática da literatura. **Métodos e pesquisas em Educação**, v. 1, p. 164, 2016.

PAULINO JÚNIOR, J. W.; OLIVEIRA, F. K. Pensamento computacional com ênfase no ensino de Lógica de programação: revisão sistemática de literatura: Computational thinking with emphasis on teaching programming logic: systematic literature review. **Revista Semiárido De Visu**, [S. l.], v. 10, n. 3, 2022. DOI: 10.31416/rsdv.v10i3.396. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/396>. Acesso em: 26 jul. 2023.

PAPERT, Seymour. **Teaching children to be mathematicians us. teaching about mathematics**. memo, 1980.

RADIANTI, Jaziar et al. A systematic review of immersive virtual reality applications for higher education: Design elements, lessons learned, and research agenda. **Computers & Education**, v. 147, p. 103778, 2020.

ROCHA, E. dos S.; OLIVEIRA, F. K. de. Critérios para divisão de recursos financeiros e orçamentários na execução da Política de assistência estudantil em instituições da rede federal de educação:: uma revisão sistemática de literatura. **Revista Semiárido De Visu**, [S. l.], v. 11, n. 3, p. 682-698, 2023. DOI: 10.31416/rsdv.v11i3.781. Disponível em:



<https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/781>.  
Acesso em: 1 jan. 2024.

SAVIĆ, Miloš; IVANOVIĆ, Mirjana; JAIN, Lakhmi C. **Complex networks in software, knowledge, and social systems**. Cham: Springer International Publishing, 2019.

SHARMA, Kshitij; PAPAVLASOPOULOU, Sofia; GIANNAKOS, Michail. Children's facial expressions during collaborative coding: Objective versus subjective performances. *International Journal of Child-Computer Interaction*, v. 34, p. 100536, 2022.

SILVA, G. H. da; OLIVEIRA, F. K. de. Mapeamento Sistemático de Literatura sobre Pensamento Matemático-Computacional. *Revista Semiárido De Visu*, [S. l.], v. 11, n. 3, p. 637-648, 2023. DOI: 10.31416/rsdv.v11i3.788. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/788>. Acesso em: 1 jan. 2024.

SILVA, G. H.; OLIVEIRA, F. K. Pensamento Computacional e STEAM numa perspectiva interdisciplinar com a matemática e informática: Uma revisão sistemática de literatura: Computational Thinking and STEAM in an interdisciplinary perspective with mathematics and informatics: A systematic literature review. *Revista Semiárido De Visu*, v. 10, n. 3, 2022. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/399>. Acesso em: 01 jan. 2024.

SILVA, G. G.; OLIVEIRA, F. K. de. Material didático utilizado na Educação Profissional de Jovens e Adultos: Uma revisão sistemática da literatura. *Revista Semiárido De Visu*, [S. l.], v. 9, n. 3, p. 335-343, 2021. DOI: 10.31416/rsdv.v9i3.308. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/308>. Acesso em: 01 jan. 2024.

SUN, Liping et al. A systematic literature review of teacher scaffolding in game-based learning in primary education. *Educational Research Review*, p. 100546, 2023.

ZARESTKY, Jill et al. Reflective writing supports metacognition and self-regulation in graduate computational science and engineering. *Computers and Education Open*, v. 3, p. 100085, 2022.

ZHAO, Yu; LLORENTE, Ana María Pinto; GÓMEZ, María Cruz Sánchez. Digital competence in higher education research: A systematic literature review. *Computers & Education*, v. 168, p. 104212, 2021.

ZHARYLGASSOVA, Peruza et al. RETRACTED: Psychological and pedagogical foundations of practice-oriented learning of future STEAM teachers. 2021.

## APÊNDICE N - Revisão Sistemática de Literatura 02

### Pensamento computacional com ênfase no ensino de Lógica de programação: revisão sistemática de literatura

## *Computational Thinking with an emphasis on the teaching of Programming Logic: a Systematic Literature Review*

**PAULINO JÚNIOR, José Walter.** Especialista MBA em Gestão de Projetos  
Instituto Federal de Educação, Ciências e Tecnologia do Sertão Pernambucano-Campus Salgueiro.  
Endereço da instituição - Salgueiro - Pernambuco - Brasil. CEP: 56.000-000 / Telefone: (88)  
99811.8497 / E-mail: jose.walter@aluno.ifsertao-pe.edu.br.orcid.org/0000-0003-1587-9858

**OLIVEIRA, Francisco Kelsen.** Doutor em Ciência da Computação  
Instituto Federal de Educação, Ciências e Tecnologia do Sertão Pernambucano-Campus Salgueiro.  
Endereço da instituição - Salgueiro - Pernambuco - Brasil. CEP: 56.000-000 / Telefone: (85) 98817.8151  
/ E-mail: francisco.oliveira@ifsertao-pe.edu.br.orcid.org/0000-0002-7382-3206

### RESUMO

É importante a adoção de metodologias dinâmicas de ensino, entretanto a forma como o ensino é ministrado pelas escolas brasileiras em sua considerável parcela, ainda se baseia em métodos tradicionais e engessados. Nesse contexto, a programação de computadores é relatada como uma disciplina complexa e de difícil associação por parte dos discentes, apontando o método tradicional como o responsável. Diante do exposto, este trabalho tem por objetivo realizar uma Revisão Sistemática de Literatura (RSL) para verificar e levantar pesquisas relacionadas a aplicação do PC ao ensino de Lógica de Programação, delimitando o território nacional. Para tanto, foi realizada a RSL utilizando-se da metodologia PRISMA analisando os principais trabalhos que abordavam o tema dispostos em publicações do Simpósio Brasileiro de Informática na Educação (SBIE) e o Workshop de Educação em Informática (WEI) entre os anos de 2005 e 2022. Os resultados dos trabalhos mostraram que a ferramenta Scratch é bastante difundida no meio acadêmico e que apresenta, em sua maioria, resultados positivos. A utilização de jogos digitais como incremento ao processo de ensino também aparece como um mecanismo que motiva para o desafio de recompensa, e a computação desplugada apresenta-se bastante enviesada com foco na proposta de ensinar as bases da computação sem a necessidade do uso dos computadores, proporcionando assim um ensino de baixo custo. Foram analisados 20 artigos, com o objetivo de apresentar uma visão crítica da área de pesquisa e descrever quais as linhas de pesquisa e os resultados dos pesquisadores do contexto nacional.

**Palavras-chave:** lógica de programação; pensamento computacional; aprendizagem.

### ABSTRACT

It is important to adopt dynamic teaching methodologies, however the way in which teaching is provided by Brazilian schools in its considerable portion is still based on traditional and plastered methods. In this context, computer programming is reported as a complex discipline and difficult to associate by students, pointing to the traditional method as responsible. In view of the above, this work aims to carry out a Systematic Literature Review (RSL) to verify and survey research related to the application of the PC to the teaching of Programming Logic, delimiting the national territory. For that, the RSL was carried out using the PRISMA methodology, analyzing the main works that addressed the theme arranged in publications of the Brazilian Symposium on Informatics in Education (SBIE) and the Workshop on Informatics Education (WEI) between 2005 and 2005. 2022. The results of the work showed that the Scratch tool is quite widespread in the academic environment and that it presents mostly positive results. The use of digital games as an increment to the teaching process also appears as a mechanism that motivates the challenge of reward, and unplugged computing presents itself quite biased with a focus on the proposal to teach the basics of computing without the need to use computers. , thus providing a low-cost education. Twenty articles were analyzed, with the objective of presenting a critical view of the research area and describing the lines of research and the results of researchers in the national context.

**Keywords:** programming logic; computational thinking; learning.

### Introdução

Desafios do ensino e aprendizagem de lógica de programação têm sido bastante discutidos dentro do meio acadêmico e científico. Metodologias e ferramentas vêm sendo propostas para minimizar as dificuldades de quem é iniciante no estudo. O Pensamento Computacional, por outro lado, introduz uma nova abordagem para a área da Ciência do Pensamento e Ciência da Computação, pois parte da ideia de que seja possível utilizar-se de conceitos da própria área da Computação na educação básica, aperfeiçoando o desenvolvimento de habilidades como a abstração, no qual ajuda crianças e adolescentes em seu processo de construção criativa e de resolução de problemas em todas as áreas da vida. Desse modo, pode-se considerar o PC como um aliado a futuros profissionais da sociedade digital.

Estudos que estabelecem relações com o pensamento computacional vêm ganhando notório espaço, tanto no Brasil quanto no contexto mundial. Desde a criação do termo "Pensamento Computacional" pela pesquisadora e cientista da Microsoft, Jeannette Wing em 2006, diversos pesquisadores veem o PC como mecanismo de grande importância dentro do ensino de programação.

É preciso ressaltar que pensar computacionalmente não é nenhum sinônimo de construir aplicações por meio da programação, no entanto Blikstein (2008) define que uma das etapas fundamentais do PC é saber programar um computador para a realização de tarefas cognitivas de maneira automatizada para que esse conhecimento seja um tipo de suporte ao raciocínio humano. De Paula, Valente e Burn (2014) afirmam que o PC é uma maneira de se pensar e de analisar uma determinada situação ou artefato, independente de tecnologia, porém defendendo a utilização de ideias teóricas que são tidas como a base podem ser aliadas a aspectos práticos da programação.

Diante do que foi exposto, se faz necessário e relevante uma revisão sistemática na qual seja possível indicar as pesquisas brasileiras avançam na utilização prática dos conceitos de PC no ensino de programação, como também, prover um instrumento norteador para futuras pesquisas na área.

Desse modo, o presente trabalho tem como seu objetivo principal, realizar uma Revisão Sistemática de Literatura (RSL) para verificar e levantar pesquisas relacionadas à aplicação do PC ao ensino de Lógica de Programação, delimitando o território nacional. Desse modo, foram utilizados estudos que estão relacionados a práticas que envolvam ou estimulem o PC objetivando o ensino de programação de computadores ou direcionados a práticas educacionais que utilizem ferramentas e métodos que se utilizem de conceitos de programação. Este trabalho detalha resultados obtidos através de uma Revisão Sistemática de Literatura (RSL), apresentando seu processo de desenvolvimento, mostrando as regras e procedimentos adotados durante a pesquisa até chegar nos resultados finais.

## **Referencial teórico**

O presente referencial teórico abordará conceitos sobre a educação profissional e tecnológica, ensino de computação e pensamento computacional.

## Educação profissional e tecnológica

No início da década de 90 acontece um marco na educação brasileira, a promulgação da Lei de Diretrizes e Bases - LDB em 1996 aprovada durante o Governo de Fernando Henrique Cardoso, implementa uma educação voltada para a cidadania e meios para progredir no trabalho (FRIGOTTO & CIAVATTA, 2003; COELHO, 2014). A educação profissional técnica de nível médio, em 1997, tem uma alteração significativa nas suas bases com a promulgação do Decreto no 2.208 de 17 de abril de 1997, que versava sobre a separação do ensino técnico de nível médio e o ensino regular, sendo prioritariamente ofertada na forma concomitante e subsequente e, assim, o ensino médio obtém um caráter propedêutico e dual (MENDES, 2011; PACHECO, 2012).

Afirma Ramos (2002) que as transformações não aconteceram apenas no formato de oferta da educação profissional, houve também uma reformulação curricular, pautada em competências para traçar o perfil do futuro profissional com metodologias apontadas em resolução de problemas e projetos. Desse modo, a formação técnica deve proporcionar ao aluno a integração também com sua formação básica escolar de ensino médio.

Os cursos técnicos são formação de nível médio com objetivo de ingressar os discentes de uma forma mais rápida no mercado (DEITOS; LARA; DE BARROS, 2016) Ao contrário das graduações que têm uma visão mais ampla de conhecimento fornecendo uma formação mais qualificada para exercer atividades tais como pesquisadores e cientistas, e dar continuidade aos estudos, como por exemplo mestrado e doutorado (SEVERINO, 2008). Por esse motivo, o objetivo dos cursos técnicos é fornecer conhecimentos mais práticos em relação às graduações objetivando uma inserção mais rápida no mercado de trabalho. Nesse sentido a computação quando trabalhada através da Educação Profissional e Tecnológica (EPT) como meio e fim, possibilita ao aluno a construção do seu próprio conhecimento e o desenvolvimento de sua capacidade resolutiva de problemas.

### Ensino de computação.

A utilização da programação como instrumento pedagógico e didático favorece a aprendizagem ativa, uma vez que a interação estabelecida entre as ações dos estudantes e o feedback do programa favorece a dinâmica. Na perspectiva de Valente (1999) no qual trata da formação de professores para a área da informática na educação, objetiva preparar os participantes para o uso do computador em sala de aula, versando em atividades que usam o computador como mecanismo para que os alunos possam construir seu conhecimento.

Almeida (2009) afirma que o aluno sintetiza os passos necessários para atingir determinado objetivo, sendo direcionado a transformar seus conhecimentos em procedimentos. O ensino dos princípios da computação proporciona o desenvolvimento do pensamento computacional, sendo uma habilidade fundamental para todos, não só de cientistas da computação.

Os autores citados discutem e conceituam a tecnologia e propõe a

ideia de que ela possa ser uma vertente aliada ao ensino de forma alternativa às metodologias tradicionais, dado o contexto em que estamos vivendo, onde as transformações são constantes e em tempo real, se faz necessário que sua aplicação esteja bem fundamentada no ambiente escolar, porém, apenas utilizá-la como fim não é o suficiente para um bom desenvolvimento, pois precisa fazer sentido ao alunado. O ensino de computação as vezes é confundido com as aulas de Informática, que tem em seu propósito instruir o aluno quanto ao uso de softwares, restringindo o ensino de computação ao simples manuseio de aplicativos e equipamentos eletrônicos.

Desse modo, a utilização do Pensamento Computacional na educação básica não visa apenas a empregabilidade e desenvolvimento econômico, mas também a construção de habilidades. A seção seguinte irá tratar do pensamento computacional a partir dos seus fundamentos, entre algumas citações, estarão as de Papert e Wing, considerados como pioneiros nos estudos do Pensamento Computacional.

### **Pensamento computacional**

O termo introduzido em Wing (2006) é um método para ensinar a solucionar problemas, conceber sistemas e compreender o comportamento humano, por meio de conceitos da Ciência da Computação ou, como colocado por Paula, Valente e Burn (2014), "É um modo de pensar e analisar de forma específica uma situação ou artefato, independente de tecnologia". Assim, desenvolver habilidades em PC não significa necessariamente aprender a programar (ZANETTI; BORGES; RICARTE, 2016.)

Wing (2006) afirma que o pensamento computacional (PC) deve ser uma habilidade inerente e básica a ser ensinada a crianças, assim como ler ou escrever, por exemplo. Alguns currículos, enquanto componente extracurricular já trazem em suas práticas o ensino de programação tanto no ensino fundamental quanto no ensino médio. Práticas como estas podem levar o aluno ao desenvolvimento de habilidades de abstração e conseqüentemente a resolução de problemas, habilidades muito importantes no atual contexto tecnológico que vivemos. Desse modo, começar a utilizar o PC como metodologia de ensino poderá impactar positivamente na construção desses saberes.

Nunes (2011) afirma ainda que o pensamento computacional pode ser entendido como o processo sistematizado das fases da resolução de problemas, podendo ser aplicado não só na Ciência da Computação, como também nas demais áreas. Já na perspectiva Lee (2014), o pensamento computacional atribui-se ao raciocínio lógico, incluindo características como: formulação de problemas, representações abstratas, organização e análise lógica de dados, identificação, análise e implementação de soluções para resolução de problemas.

### **Material e métodos**

A presente revisão bibliográfica foi realizada a partir de vertentes da metodologia PRISMA *Preferred Reporting Items for Systematic Re-views and*

*Metaanalyses*, tendo como base os trabalhos de Moher D et al (2015). A metodologia objetiva ajudar autores a melhorarem a construção de revisões sistemáticas e meta-análises, também podendo ser utilizada como base para relatos de outras revisões.

No processo de construção desta Revisão Sistemática de Literatura (RSL) foram especificadas as palavras-chaves Pensamento Computacional, Lógica de Programação, Linguagem de Programação, Ensino de Computação e Scratch que posteriormente se tornaram as strings de buscas.

Após a escolha das principais palavras-chaves, foram elencadas a estrutura de pesquisa nas plataformas SBC Lib Online e Google Acadêmico utilizando a ferramenta de revisão sistemática StArt no qual buscou-se apenas trabalhos no idioma português em anais de eventos e periódicos, com os resultados apresentados no quadro 1 abaixo:

Quadro 1- Palavras-chaves

Strings	SBC Open Lib	Google	StArt
"Pensamento Computacional" + "Lógica de Programação" + "Ensino de Computação" + "Scratch"	307 resultados	319	Google: 331 SBC: 278
"Pensamento Computacional" + "Linguagem de Programação" + "Ensino de Computação" + "Scratch"	404 resultados	417	Google: 364 SBC: 417

Fonte: adaptada de ARAÚJO e OLIVEIRA (2021)

Finalizada as consultas com base nas palavras-chaves e combinações apresentadas foram localizados os trabalhos relacionados expostos no quadro 2 abaixo:

Quadro 2 - Resultados de buscas por strings

Palavra-chave	Google Acadêmico	SBC Open Lib
Pensamento Computacional	2.060 trabalhos	271 trabalhos
Linguagem de Programação	4.640 trabalhos	144 trabalhos
Lógica de Programação	4.170 trabalhos	104 trabalhos
Ensino de Computação	4.240 trabalhos	1868 trabalhos

Scratch	20.700 trabalhos	129 trabalhos
---------	------------------	---------------

### Critérios de Inclusão e Exclusão

O critério de inclusão definido a partir da capacidade do mesmo em solucionar pelo menos uma das questões da pesquisa, após esse primeiro critério foram definidos outros critérios como mostra o quadro 3 abaixo. Os critérios de exclusão definem diretrizes para excluir aqueles que não se enquadram totalmente no contexto da pesquisa.

Quadro 3 - Critérios de Inclusão e Exclusão

Critérios		Descrição
Inclusão	C1	Artigos resumidos ou completos publicados em periódicos científicos das bases listadas.
	C2	Apresentam experiências positivas e negativas de ensino com cratch
	C3	Artigos que abordam sobre o uso com finalidade educacional da ferramenta Scratch no ensino de lógica/linguagem de programação.
Exclusão	E1	Artigos duplicados ou semelhantes.
	E2	Artigos apenas com caráter publicitário ou de marketing publicados em magazines.
	E3	Apresentação de slides.
	E4	Artigos que aparentemente não contribuem para a pesquisa vigente.

Fonte: Pesquisa direta.

O objetivo da RSL é verificar e avaliar contribuições de estudos publicados e disponibilizados nas bases de dados do Google Acadêmico e da Biblioteca da Sociedade Brasileira de Computação (SBC), para tanto, apresentam contribuições no ensino de programação de computadores e pensamento computacional. Desse modo, pretende-se responder a seguinte questão base para esta pesquisa: "No contexto do ensino de programação utilizando o Pensamento Computacional, o que se tem produzido nas pesquisas nacionais?" Com base nessa questão norteadora, outras três questões de pesquisa mais bem definidas e específicas foram levantadas no quadro 4 abaixo:

quadro 4 - Questões da pesquisa

ID	Perguntas	Motivação
P 01	Quais vantagens ou desvantagens são observadas pelos usuários no uso educacional da ferramenta ?	Entender o ambiente e contexto em que a ferramenta é utilizada para o ensino de LP bem como conhecer as lições aprendidas a partir das experiências que não surtiram o devido efeito.
P 02	Conhecer como o Pensamento Computacional pode colaborar com o ensino de Lógica de Programação ?	Em meio a um universo de experiências exitosas, retirar aquelas que alcançaram resultados modestos ou insatisfatórios de acordo com os objetivos elencados.
P 03	Como a ferramenta Scratch pode colaborar com o ensino de Lógica de Programação a partir dos conceitos de Pensamento Computacional?	Identificar as colaborações e práticas com o uso de scratch aplicadas ao ensino de LP com a utilização de PC

Fonte: Pesquisa direta.

Os artigos selecionados estão de acordo com atendimento aos critérios de inclusão, a partir da solução das perguntas definidas por meio da leitura do resumo e do título do trabalho. A busca pelos artigos englobou as bases de dados citadas, sendo selecionados trabalhos publicados em eventos tais como simpósio e congressos, a exemplo do Simpósio Brasileiro de Informática na Educação (SBIE), Workshop de Educação em Informática (WEI) dentre outros a partir de 2005. Não foram selecionados artigos que apenas realizaram uma revisão da literatura, sendo considerados aqueles que descrevem e/ou aplicam práticas pedagógicas e a utilização de ferramentas educacionais. O quadro 5 abaixo apresenta e organiza os artigos selecionados por um identificador único (ID), título e o autor.

Quadro 5 - Artigos catalogados

ID	Autor	Contribuições
----	-------	---------------



A01	Campos e Souza (2020).	O autor busca identificar habilidades de autonomia e criatividade em sua intervenção no qual pode ser considerada exitosa. A partir do desenvolvimento dos projetos é verificado que os resultados foram
		satisfatórios. Tanto no que se refere a interdisciplinaridade quanto a problemática inicial. Com a utilização do Scratch também foi possível relacionar com as disciplinas do curso técnico avaliado.
A02	POLONI; SOARES; WEBBER, (2019).	Ao se utilizar o ambiente scratch, as possibilidades de aprendizagem são expandidas, porém, o professor também deve atuar como mediador, desenvolvendo estratégias e intervenções para auxiliar os estudantes.
A03	Tridapalli e Mattos (2019).	Propõe uma plataforma para apoiar o ensino de Lógica de Programação a partir de conceitos do Pensamento Computacional no qual é bem sucedida quando implementada.
A04	Vieira e Sabatini (2021).	Os autores nos apresentam que a partir da intervenção da ferramenta scratch, os alunos desenvolveram interesse e curiosidade na resolução de problemas, desse modo, fazendo com que o estudante esteja participativo. O uso do scratch acerca do pensamento computacional é se mostrou fundamental para que a escola possa criar espaços de desenvolvimento, despertando no aluno o interesse em "criar" a partir do "fazer".

A05	Bombasar et al. (2015).	O presente trabalho realizou uma revisão sistemática de literatura entre os anos de 2006 até 2015 na tentativa de encontrar ferramentas para o ensino do pensamento computacional baseado em teorias e formalismo de Alan Turing, porém, nenhum trabalho apresentou uma ferramenta coerente com a proposta.
A06	Amorim (2015).	Educadores precisam se apropriar de técnicas do pensamento computacional para que desse modo, possam implementar melhorias em suas metodologias de ensino, o autor

		em seu trabalho visa identificar o nível de pensamento computacional de futuros professores no qual os resultados apontaram que os mesmos possuem um nível básico, demonstrando a real necessidade de inserção dessa dimensão na formação de professores.
A07	Moretti (2019).	Ao pensar na utilização da tecnologia em sala de aula, a pesquisa em questão faz o leitor refletir maneiras de resolução de problemas, investigando os próprios recursos do programa sendo uma pesquisa reflexiva com potencial para gerar novas perguntas.
A08	MASSA (2019).	O trabalho se trata de uma revisão de literatura no qual é feito um mapeamento de trabalhos publicados entre 2012 e 2017 no qual é identificado que o aluno é o construtor do seu próprio conhecimento desenvolvendo o trabalho cooperativo por meio do uso do pensamento computacional e das tecnologias de construção.

A09	Ferreira et al. (2018).	<p>Os autores utilizaram atividades lúdicas através da adoção das abordagens de computação desplugada que visa estimular o pensamento computacional sem o uso do computador e ensino de programação utilizando o ambiente Scratch. Os resultados obtidos mostram que independente dos recursos tecnológicos disponíveis na escola, é necessário realizar atividades que estimulem o desenvolvimento de habilidades relacionadas com o pensamento computacional. Como também, há indícios de que a inserção do pensamento computacional pode contribuir para melhoria no desempenho dos alunos na resolução de problemas matemáticos.</p>
A10	MADEIRA (2017 ).	<p>Este trabalho uniu as metodologias da aprendizagem baseada em resolução de problemas e em jogos digitais com técnicas de programação visual a fim de estimular o desenvolvimento do pensamento computacional, visando o uso do computador como um instrumento de aumento do poder cognitivo e operacional e humano. Tendo como resultados demonstrar a simplicidade e a viabilidade do emprego das metodologias de aprendizagem baseadas em resolução de problemas e em jogos digitais quando associadas a programação visual.</p>

A11	Pires et al. (2019).	O trabalho em questão explorou conceitos de lógica de programação utilizando abordagens como Computação desplugada, programação com Scratch e Python e Robótica através de gamificação. Os resultados evidenciaram que as atividades com maior nível de ludicidade foram mais bem recebidas pelos alunos.
A12	Krugel e Macedo (2021).	Esse trabalho apresentou a experiência de um projeto implementado em uma turma de matemática, com atividades usando o Scratch e abordando os conceitos do Pensamento Computacional. Foi realizado um estudo de caso na disciplina de Matemática do 9º ano do Ensino Fundamental, onde foi possível observar que os alunos construíram uma relação entre os conhecimentos prévios da disciplina e os que foram apresentados por meio do Scratch, tornando assim o aprendizado mais significativo.
A13	Rodrigues (2019).	Os resultados se mostraram
		favoráveis e que o desempenho dos participantes apresentaram-se como satisfatórios tanto nas atividades das semanas quanto nos conceitos apresentados, enquanto também, permaneciam instigados durante as fases do experimento. Desse modo, pode-se este indicar um forte indício positivo da influência da programação em blocos com Scratch, no ensino/aprendizado dos conceitos de programação e da Engenharia de Software.

A14	Wangenheim; Nunes; Santos, (2014).	A presente pesquisa sugere uma unidade instrucional para o ensino de computação no Ensino Fundamental de forma interdisciplinar, seguindo as diretrizes de currículo para o ensino de computação K-12 usando SCRATCH. A unidade instrucional foi implementada e posta a avaliação em uma turma do primeiro ano de uma escola em Florianópolis/SC. Os resultados apontaram que a unidade instrucional e o uso do SCRATCH possibilitaram a aprendizagem de conceitos básicos de computação (especificamente da programação) de forma dinâmica e divertida e despertaram o interesse e motivação dos alunos para esta área de conhecimento.
A15	Pimenta (2019).	A introdução de noções de algoritmos e programação aproxima as crianças da área da Computação e o desenvolvimento de suas tecnologias, dessa forma o que era desconhecido passou a fazer parte do conhecimento de cada uma delas. Para os licenciandos a experiência foi de grande importância para sua formação docente, pois tornou-se uma excelente oportunidade para unirem teoria e prática, além de adquirirem novos conhecimentos. Ao
		final, percebe-se o potencial e a importância das tecnologias no desenvolvimento humano, pois contribuem com sua capacidade de raciocinar e de realizar tarefas.

A16	Gobbi e Silveira (2020).	Os resultados obtidos neste trabalho mostraram que independente de quais mecanismo tecnológicos disponíveis na escola, se faz necessário realizar atividades que estimulem o desenvolvimento de habilidades relacionadas ao pensamento computacional. Além disso, a inserção do pensamento computacional contribui para melhoria no desempenho dos alunos na resolução de problemas matemáticos.
A17	Marinho (2018).	Para que fosse possível entender de algoritmos, o Scratch foi utilizado como uma ferramenta de auxílio pedagógico, assim como também a metodologia de computação desplugada. Os resultados alcançados mostram que é possível incluir a temática no cotidiano escolar de maneira que desperte o interesse dos alunos da educação básica.
A18	Cavazin e Silveira. (2020 ).	O presente trabalho verifica que a programação Scratch proporciona aos alunos um ambiente motivador, e resultados positivos no sentido em que eles possam desenvolver habilidades para a resolução de problemas. O autor relatava resultados satisfatórios no processo de ensino, no qual os alunos se mostraram atenciosos às atividades, colaborando com os colegas no desenvolvimento de ideias nos projetos, algumas dificuldades quanto ao uso das ferramentas do Scratch surgiram, mas também

		demonstraram aptidão em algumas delas. Percebe-se que o Scratch é uma ferramenta que impulsiona no aspecto motivacional e desperta interesse no processo de construção do conhecimento.
A19	SCHOEFFEL, Pablo et al (2015)	Os resultados alcançados neste trabalho mostraram que independente dos recursos tecnológicos disponíveis na escola, é necessário a realização de atividades que estimulem o desenvolvimento de habilidades relacionadas com o pensamento computacional. Além disso, a implementação do pensamento computacional contribui para melhoria no desempenho dos alunos na resolução de problemas matemáticos.
A20	Ramos e Teixeira (2015 ).	O presente artigo analisou os resultados observados em um minicurso intitulado “Scratch e Desenvolvimento de Jogos”. Concluindo que os público alvo foi capazes de interagir e construir objetos digitais num ambiente de linguagem de programação de forma exitosa, apresentando especificidades particulares a sua condição de estudantes da educação.

Fonte: Pesquisa direta.

## Resultados e discussão

Como foi apresentado na seção anterior, o número de pesquisas que abordam o tema pensamento computacional é deveras crescente. O ensino de programação tem crescido bastante nos últimos tempos, dado avanço da tecnologia e do digital cada vez mais presente no nosso cotidiano. Os trabalhos foram categorizados por práticas, a fim de se descobrir as categorias de aplicação que estão sendo utilizadas. Foram definidas cinco categorias, tendo como objetivo propor um modelo para futuros pesquisadores que queiram se relacionar com outras metodologias e práticas dentro do contexto de computação. Após a leitura dos trabalhos foi identificado as seguintes categorias: Linguagens de Programação

LP), Jogos Digitais (JD) e Computação Desplugada (CD). No quadro 6 a seguir, é apresentada uma breve descrição sobre cada categoria.

**Quadro 6 - Categorização**

Categoria	Descrição
Linguagens de Programação (LD)	Utilização de linguagens de programação como prática de ensino, exemplo: Python, Java, C, etc.
Computação Desplugada (CD)	Métodos que visam o ensino de computação sem necessariamente utilizar o computador.
Jogos Digitais (JD)	Utilização de jogos digitais como mecanismo de ensino de computação.

Fonte: Pesquisa direta.

De modo a relacionar os trabalhos com as categorias selecionadas, foi elaborado um esquema que representa a associação entre ambas no quadro 7 abaixo:

**Quadro 7- Práticas relacionadas**

Prática	Trabalho (ID)
Linguagens de Programação (LD)	A10, A02, A03, A12, A07, A05, A13, A15
Computação Desplugada (CD)	A01, A19, A10, A16, A06, A07, A08, A11
Jogos Digitais (JD)	A14, A05, A03, A09, A18, A09, A19, A20, A04

Fonte: Pesquisa direta.

Após o levantamento, foi observado uma diversidade nos estudos. Não há um padrão, 14 trabalhos foram voltados para o ensino médio/técnico, 3 para o ensino fundamental e 3 para o ensino superior. Pela análise dos dados, observou-se uma tendência de ensino, no qual os próprios alunos possuem capacidade de abstração adequada à proposta.

Todos os trabalhos em seus resultados apontaram para uma aprendizagem significativa por meio dos conceitos básicos de computação (especialmente programação) de forma divertida, impactando diretamente no desenvolvimento do interesse e motivação dos alunos. A seguir será apresentado a relação entre as questões de pesquisa e as abordagens de alguns trabalhos selecionados, os mais relevantes.

**P01: Quais vantagens ou desvantagens são observadas pelos usuários no uso educacional da ferramenta?**



Esta questão buscou verificar se os trabalhos em análise entregam vantagens reais e práticas em sua aplicação. Após o levantamento foi possível perceber que a maioria dos estudos apresentam resultados positivos em relação ao seu contexto de aprendizagem. Em [A01], o autor busca identificar habilidades de autonomia e criatividade em sua intervenção, no qual pode ser considerada exitosa. A partir do desenvolvimento dos projetos é verificado que os resultados foram satisfatórios. Tanto no que se refere a interdisciplinaridade quanto a problemática inicial. Com a utilização do Scratch também foi possível relacionar com as disciplinas do curso técnico avaliado.

[A06] concluiu que os educadores precisam se apropriar de técnicas do pensamento computacional para que desse modo, possam implementar melhorias em suas metodologias de ensino, o autor em seu trabalho visa identificar o nível de pensamento computacional de futuros professores no qual os resultados, após a utilização da ferramenta Scratch, apontaram que os mesmos possuem um nível básico, demonstrando a real necessidade de inserção dessa dimensão na formação de professores.

[A12] apresentou a experiência de um projeto implementado em uma turma de matemática, com atividades usando o Scratch e abordando os conceitos do Pensamento Computacional. Foi realizado um estudo de caso na disciplina de Matemática do 9º ano do Ensino Fundamental, onde foi possível observar que os alunos construíram uma relação entre os conhecimentos prévios da disciplina e os que foram apresentados por meio do Scratch, tornando assim o aprendizado mais significativo.

[A13] apresentou seus resultados no qual se mostraram favoráveis e que o desempenho dos participantes apresentaram-se como satisfatórios tanto nas atividades das semanas quanto nos conceitos apresentados, enquanto também, permaneciam instigados durante as fases do experimento. Desse modo, pode-se este indicar um forte indício positivo da influência da programação em blocos com Scratch, no ensino/aprendizado dos conceitos de programação e da Engenharia de Software. [A14] sugeriu uma unidade instrucional para o ensino de computação no Ensino Fundamental de forma interdisciplinar, seguindo as diretrizes de currículo para o ensino de computação K-12 usando SCRATCH. A unidade instrucional foi implementada e posta a avaliação em uma turma do primeiro ano de uma escola em Florianópolis/SC. Os resultados apontaram que a unidade instrucional e o uso do SCRATCH possibilitaram a aprendizagem de conceitos básicos de computação (especificamente da programação) de forma dinâmica e divertida e despertaram o interesse e motivação dos alunos para esta área de conhecimento.

[A15] enfatiza que uma metodologia voltada a jogos e dinâmica, desperta maior interesse por parte dos alunos, oportunizando o desenvolvimento de novas habilidades que lhes serão úteis nas diversas disciplinas curriculares. A introdução de noções de algoritmos e programação aproxima as crianças da área da computação e o desenvolvimento de suas tecnologias, dessa forma o que era desconhecido passou a fazer parte do conhecimento de cada uma delas. Para os licenciandos a experiência foi de grande importância para sua formação docente, pois tornou-se uma excelente oportunidade para unirem teoria e prática, além de adquirirem novos conhecimentos. Ao final, percebe-se o potencial e a importância

das tecnologias no desenvolvimento humano, pois contribuem com sua capacidade de raciocinar e de realizar tarefas.

[A16] em seus resultados obtidos, mostraram que independente de quais mecanismo tecnológico disponíveis na escola, se faz necessário realizar atividades que estimulem o desenvolvimento de habilidades relacionadas ao pensamento computacional. Além disso, a inserção do pensamento computacional contribui para melhoria no desempenho dos alunos na resolução de problemas matemáticos. [A20] analisa os resultados observados em um minicurso intitulado “Scratch e Desenvolvimento de Jogos”. Concluindo que os público alvo foi capazes de interagir e construir objetos digitais num ambiente de linguagem de programação de forma exitosa, apresentando especificidades particulares a sua condição de estudantes da educação.

### **P02: Conhecer como o Pensamento Computacional pode colaborar com o ensino de Lógica de Programação ?**

O trabalho [A07] desenvolve a ideia da necessidade de pensar na utilização da tecnologia em sala de aula, a referida pesquisa faz o leitor refletir maneiras de resolução de problemas, investigando os próprios recursos do programa, sendo uma pesquisa reflexiva com potencial para gerar novas perguntas. Com base na questão norteadora, [A03] propõe uma plataforma para apoiar o ensino de Lógica de Programação a partir de conceitos do Pensamento Computacional no qual é bem sucedida quando implementada.

[A10] uniu as metodologias da aprendizagem baseada em resolução de problemas e em jogos digitais com técnicas de programação visual a fim de estimular o desenvolvimento do pensamento computacional, visando o uso do computador como um instrumento de aumento do poder cognitivo e operacional e humano. Tendo como resultados apresentar a simplicidade e a viabilidade do emprego das metodologias de aprendizagem baseadas em resolução de problemas e em jogos digitais quando associadas a programação visual.

### **P03: Como a ferramenta Scratch pode colaborar com o ensino de Lógica de Programação a partir dos conceitos de Pensamento Computacional?**

A programação Scratch é apresentada aos alunos como um ambiente motivador e de resultados positivos na maioria dos trabalhos, no sentido em que eles possam desenvolver habilidades enquanto realizam as tarefas. No trabalho [A02] verificou-se que o ambiente, entrega possibilidades de aprendizagem que logo são expandidas, porém, o professor também deve atuar como um elo, propondo metodologias e intervenções para auxiliar os estudantes. Em [A04] é apresentado que por meio da intervenção da ferramenta scratch, os alunos desenvolveram interesse e curiosidade na resolução de problemas, desse modo, fazendo com que o estudante esteja

participativo.

O uso do Scratch acerca do pensamento computacional se mostrou fundamental para que a escola possa criar espaços de desenvolvimento, despertando no aluno o interesse em "criar" a partir do "fazer". [A09] os autores utilizaram atividades lúdicas através da adoção das abordagens de computação

desplugada visando estimular o pensamento computacional sem o uso do computador e ensino de programação utilizando o ambiente Scratch. Os resultados obtidos mostram que independente dos recursos tecnológicos disponíveis na escola, é necessário realizar atividades que estimulem o desenvolvimento de habilidades relacionadas com o pensamento computacional. Como também, há indícios de que a inserção do pensamento computacional pode contribuir para melhoria no desempenho dos alunos na resolução de problemas matemáticos. [A11] explorou conceitos de lógica de programação utilizando abordagens como Computação desplugada, programação com Scratch e Python e Robótica através de gamificação. Os resultados evidenciaram que as atividades com maior nível de ludicidade foram mais bem recebidas pelos alunos.

[A17] apresentou o Scratch como uma ferramenta de auxílio pedagógico, assim como também a metodologia de computação desplugada. Os resultados alcançados mostram que é possível incluir a temática no cotidiano escolar de maneira que desperte o interesse dos alunos da educação básica. [A18] verifica que a programação Scratch proporciona aos alunos um ambiente motivador, e resultados positivos no sentido em que eles possam desenvolver habilidades para a resolução de problemas. O autor relatava resultados satisfatórios no processo de ensino, no qual os alunos se mostraram atenciosos às atividades, colaborando com os colegas no desenvolvimento de ideias nos projetos, algumas dificuldades quanto ao uso das ferramentas do Scratch surgiram, mas também demonstraram aptidão em algumas delas. Percebe-se que o Scratch é uma ferramenta que impulsiona no aspecto motivacional e desperta interesse no processo de construção do conhecimento.

## Conclusões

De acordo com os resultados observados percebe-se que existe um tipo de tendência de pesquisas relacionadas ao ensino médio integrado ao técnico e no ensino fundamental, pois o ensino de programação nesses cenários pode ser contemplado com a utilização de diversas ferramentas que vão de encontro aos interesses do aluno, tais como jogos digitais e robótica educacional, por exemplo. Observou-se também que a maioria das pesquisas são recentes, tendo uma considerável parcela a partir de 2015 em diante.

A presente RSL utilizou-se de um processo de seleção através de critérios de inclusão e exclusão, objetivando a coerência e fidelidade à temática proposta. Os critérios de inclusão e exclusão foram cautelosamente discutidos

e projetados para inviabilizar o risco de exclusão de trabalhos relevantes, porém a busca foi feita apenas em bases de dados brasileiras, logo é possível que algum trabalho relevante tenha sua ausência no mapeamento, outro fator é a subjetividade de algumas avaliações de resultados, impossibilitando muitas vezes uma análise imparcial no que se trata sobre conflitos em interpretações próprias.

Outra limitação encontrada foi restringir apenas o Scratch como ferramenta de ensino de lógica de programação e pensamento computacional, desse modo, sugere-se o estudo do PC com a utilização também de outros aplicativos. Desse modo, esta pesquisa está concentrada em um estudo teórico de uma RSL referente

a utilização do Pensamento Computacional no Ensino de Lógica de Programação no contexto nacional com a finalidade de apresentar estudos referentes a aplicações práticas do Scratch como ferramenta de ensino, acreditando que seja possível contribuir e incentivar a produção acadêmica do referido tema. Para trabalhos futuros, pretende-se desenvolver um material de apoio, tipo de produto educacional aplicável a um ambiente de ensino de lógica de programação apoiado em práticas do Pensamento Computacional.

## Referências

ALMEIDA, M. E. B.; VALENTE, J. A. Tecnologias e currículo: trajetórias convergentes ou divergentes? São Paulo: Paulus, 2011.

ARAÚJO, Aline Cássia Silva; DE OLIVEIRA, Francisco Kelsen. Revisão Sistemática da Literatura sobre Tecnologias digitais de informação e comunicação de tradução do par linguístico Português Libras. *Revista Semiárido De Visu*, v. 9, n. 3, p. 286-299, 2021. Disponível em: <https://semiaridodevisu.ifsertao-pe.edu.br/index.php/rsdv/article/view/305>. Acesso em: 18 set. de 2022

BARR, Valerie; STEPHENSON, Chris. Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community?. *Acm Inroads*, v. 2, n. 1, p. 48-54, 2011.

BLIKSTEIN, Paulo. O pensamento computacional e a reinvenção do computador na educação. *Education & Courses*, v. 1, 2008.

BOMBASAR, James et al. Ferramentas para o ensino-aprendizagem do pensamento computacional: onde está Alan Turing?. In: *Brazilian symposium on computers in education (simpósio brasileiro de informática na educação-sbie)*. 2015. p. 81.

CAMPOS, Fabrício Vieira; DE SOUZA, Paulo Henrique. POSSIBILIDADES DE USO DO SCRATCH NO DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL. In: *Anais do CIET: EnPED: 2020-(Congresso Internacional de Educação e Tecnologias |*

Encontro de Pesquisadores em Educação a Distância). 2020.

CAVAZIN, Jucélia. Um Estudo de Caso envolvendo Pensamento Computacional e o uso do Scratch. 2020

DA SILVA VIEIRA, Sebastião; SABBATINI, Marcelo. Pensamento computacional através do Scratch numa perspectiva Maker. **REVISTA INTERSABERES**, v. 16, n. 37, p. 43-63, 2021.

DE OLIVEIRA, Milena et al. Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência. In: **Anais do XXII Workshop sobre Educação em Computação**. SBC, 2014. p. 239-248.

De Paula, B. H., Valente, J. A. e Burn, A. O uso de jogos digitais para o desenvolvimento do currículo para a Educação Computacional na Inglaterra. *Currículo sem Fronteiras*, v. 14, n. 3, p. 46-71.

Deitos, Roberto Antonio, Angela Mara de Barros Lara, and Isaura Monica Souza Zanardini. "Política de educação profissional no Brasil: aspectos socioeconômicos e ideológicos para a implantação do Pronatec." *Educação & Sociedade* 36 (2015): 985- 1001.

ELIEZER, Pacheco, org., *Perspectivas da educação profissional técnica de nível médio. Proposta diretrizes curriculares nacionais* (São Paulo: Moderna, 2012). Disponível em: <<http://reitoria.ifpr.edu.br/wp-content/uploads/2014/06/Perspectivas-da-EPT.pdf>> Acesso em: 12. set. 2022  
FURTADO, Vasco. *Tecnologia e Gestão da Informação na Segurança Pública*. Rio de Janeiro: Editora Garamond, 2002.

Gaudêncio Frigotto, Maria Ciavatta. "Educação básica no Brasil na década de 1990: subordinação ativa e consentida à lógica do mercado". *Educ. Soc.*, Campinas, vol. 24, n. 82, p. 93-130, abril 2003. Disponível em : <<http://www.scielo.br/pdf/es/v24n82/a05v24n82.pdf>> Acesso em: 08. set. 2022.

GOBBI, R.; SILVEIRA, S. R.; BERTOLINI, C.; BIGOLIN, N. M.; PARREIRA, F. J.; CUNHA, G. B.; MACEDO, R. T. Formação Docente: desenvolvimento do pensamento computacional nos anos finais do ensino fundamental. **Anais da 35.a JAI - Jornada Acadêmica Integrada da UFSM**. Santa Maria: UFSM, 2020.

JÚNIOR, Paulo Antonio Pasqual; DE OLIVEIRA, Simone. Pensamento computacional: uma proposta de oficina para a formação de professores. **RENOTE**, v. 17, n. 1, p. 62-71, 2019. Disponível em: <<https://seer.ufrgs.br/renote/article/view/95707/0>> Acesso em: 15 de ago. de 2022.

KRUGEL, Diego Ricardo. *Uso do Scratch para o desenvolvimento do Pensamento Computacional*. 2021.

MADEIRA, C. **Introdução ao Pensamento Computacional com Scratch**. 2017. Disponível em: [http://ceur-ws.org/Vol-1877/CtrlE2017\\_MC\\_4.pdf](http://ceur-ws.org/Vol-1877/CtrlE2017_MC_4.pdf) Acesso em: 08. jul. 20202.

MARINHO, Anna Raquel da S. et al. Relato de Experiência Vivenciada no PIBID sobre a Utilização da Computação Desplugada, a Hora do Código e do Scratch no Ensino Médio. In: **Anais do XXVI Workshop sobre Educação em Computação**. SBC, 2018.

MASSA, Nayara Poliana et al. Mapeamento do Pensamento Computacional por meio da ferramenta Scratch no contexto educacional brasileiro: análise de publicações do Congresso Brasileiro de Informática na Educação entre 2012 e 2017. 2019.

MADEIRA, Charles. Introdução ao Pensamento Computacional com Scratch. In: **Anais do II Congresso sobre Tecnologias na Educação (CTRL+ E-2017)**. 2017. p. 725-730.

MOHER, David et al. Preferred reporting items for systematic review and meta- analysis protocols (PRISMA-P) 2015 statement. **Systematic reviews**, v. 4, n. 1, p. 1-9, 2015.

MORETTI, Vinícius Fernandes. O pensamento computacional no ensino básico: potencialidades de desenvolvimento com o uso do Scratch. 2019

NUNES, D. J. Ciência da Computação na Educação Básica. **ADUFRGS - Sindical**, 6. jun. 2011. Disponível em: <http://adufrgs.org.br/artigos/ciencia-da-computacao-na-educacao-basica>. Acesso em: 21 set. de 2022

PAULA; B. H.; VALENTE; J. A.; BURN, A. O uso de jogos digitais para o desenvolvimento do currículo para a educação computacional na Inglaterra. **Currículo sem Fronteiras**, v. 14, n. 3, p. 46-71, set/dez, 2014. Disponível em: <<http://www.curriculosemfronteiras.org/vol14iss3articles/paula-valenteburn.htm>> Acesso em: 21 set. de 2022.

PAPERT, Seymour. **Teaching children to be mathematicians us. teaching about mathematics**. memo, 1980.

PIMENTA, Diogo de Farias et al. Desenvolvendo o raciocínio lógico e o pensamento computacional na educação básica: uma experiência com tecnologias educacionais e lógica de programação. 2019.

POLONI, Leonardo; DO SACRAMENTO SOARES, Eliana Maria; WEBBER, Carine G. Pensamento computacional no ensino médio: práticas mediadoras utilizando a linguagem Scratch. **RENOTE**, v. 17, n. 3, p. 508-517, 2019.

RAMOS, Marise Nogueira. **A pedagogia das competências**. São Paulo: Cortez, 2002.

SILVEIRA, S. R. et al. **Formação Docente: Desenvolvimento do pensamento**

computacional nos anos finais do ensino fundamental. Projeto de Extensão. Frederico Westphalen: UFSM, 2019.

RAMOS, Fellipe; DA SILVA TEIXEIRA, Lilian. Significação da aprendizagem através do pensamento computacional no ensino médio: uma experiência com scratch. In: **Anais do Workshop de Informática na Escola**. 2015. p. 217-226.

RODRIGUES, Renan Kodama. **Estudo do uso da linguagem de blocos scratch no ensino do pensamento computacional**. 2019. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

SILVEIRA, Sidnei Renato et al. FORMAÇÃO DOCENTE E O DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL NOS ANOS FINAIS DO ENSINO FUNDAMENTAL. **Educação Básica Revista**, v. 6, n. 2, p. 55-70, 2020.

SCHOEFFEL, Pablo et al. Uma experiência no ensino de pensamento computacional para alunos do ensino fundamental. In: **Anais dos Workshops do Congresso Brasileiro de Informática na Educação**. 2015. p. 1474.

TRIDAPALLI, Joan Giancesini; MATTOS, Mauro Marcelo. PENSAMENTO COMPUTACIONAL E GAMIFICATION: RELATO DE UM EXPERIMENTO NA PLATAFORMA FURBOT.

VALENTE, José Armando. **Liberando a mente: computadores na educação especial**. Campinas: Unicamp, 1991.

VALENTE, José Armando et al. Informática na educação no Brasil: análise e contextualização histórica. **O computador na sociedade do conhecimento**. Campinas: UNICAMP/NIED, p. 1-13, 1999.

VON WANGENHEIM, Christiane Gresse; NUNES, Vinícius Rodrigues; DOS SANTOS, Giovane Daniel. Ensino de computação com scratch no ensino fundamental-um estudo de caso. **Revista Brasileira de Informática na Educação**, v. 22, n. 03, p. 115, 2014.

VIEIRA, A. M. D. P.; SOUZA JUNIOR, A. A Educação Profissional no Brasil. **Revista Interações**, Santarém, n.40, p.152-169, 2016.

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

ZANETTI, Humberto; BORGES, Marcos; RICARTE, Ivan. Pensamento computacional no ensino de programação: Uma revisão sistemática da literatura brasileira. In: **Brazilian Symposium on Computers in Education (Simpósio Brasileiro de Informática na Educação-SBIE)**. 2016. p. 21.

## **APÊNDICE O - um guia de boas práticas: Pensamento Computacional utilizando *Scratch* .**

### **Capítulo 1: Introdução ao Pensamento Computacional e Lógica de Programação**

#### **1.1 O que é o Pensamento Computacional?**

O pensamento computacional (PC) é uma habilidade essencial no mundo atual, pois a tecnologia está cada vez mais presente em nossas vidas. De acordo com Wing (2006), o PC envolve a capacidade de resolver problemas de forma estruturada, utilizando conceitos como decomposição, reconhecimento de padrões, abstração e algoritmos. Afirma Papert (1986) que o PC é uma habilidade básica para todos, não porque todos vão trabalhar como programadores, mas porque pensamento computacional é uma forma de pensar que é valiosa em todos os aspectos da vida cotidiana. Liukas (2015), colaboradora do programa de estudos de Informática da Finlândia, adota uma abordagem bastante compreensível, descrevendo como 'refletir sobre os desafios de maneira que um computador possa resolver'. Liukas também acrescenta: 'O Pensamento Computacional é realizado por seres humanos e não por máquinas. Ele engloba o raciocínio lógico, a capacidade de identificação de padrões, a formulação de algoritmos, a decomposição e a abstração de um problema.' De acordo com as definições dos autores, podemos entender o Pensamento Computacional como uma habilidade que utiliza fundamentos da Ciência da Computação para resolver problemas do cotidiano através principalmente da abstração.

Wing (2006) Afirma que a decomposição é a habilidade de dividir um problema complexo em partes menores e mais gerenciáveis. Por exemplo, se você precisa planejar uma festa, pode decompor o problema em tarefas como escolher um tema, fazer uma lista de convidados, organizar a decoração e preparar o cardápio. Ao dividir o problema dessa forma, fica mais fácil lidar com cada parte separadamente. O reconhecimento de padrões envolve identificar semelhanças entre diferentes problemas ou situações. Por exemplo, se você já sabe como resolver um determinado tipo de problema matemático, pode aplicar os



mesmos princípios para resolver outros problemas semelhantes. Ainda de acordo com Wing (2006), abstração é a capacidade de simplificar um problema complexo removendo detalhes desnecessários. Por exemplo, ao criar um programa para calcular a média das notas dos alunos em uma turma, você não precisa se preocupar com as notas individuais de cada aluno - basta saber quantos alunos há na turma e qual é a soma total das notas. Os algoritmos são sequências ordenadas de passos que levam à solução de um problema. Eles são como receitas que indicam exatamente o que fazer em cada etapa do processo. Por exemplo, para fazer um bolo, você segue uma série de passos específicos, como misturar os ingredientes, colocar a massa no forno e esperar o tempo necessário para assar. O pensamento computacional não se limita apenas à programação de computadores. Ele pode ser aplicado em diversas áreas da vida pessoal e profissional. Um médico, por exemplo, pode usar o pensamento computacional para diagnosticar doenças, analisando sintomas e identificando padrões. Um arquiteto pode usar o pensamento computacional para projetar edifícios, decompondo o problema em diferentes etapas e criando algoritmos para cada uma delas.

## **1.1 Importância do Pensamento Computacional na Resolução de Problemas**

O pensamento computacional é fundamental na resolução de problemas porque nos ajuda a abordar os desafios de forma estruturada e eficiente. De acordo com Wing (2006) nos permite analisar problemas complexos, identificar padrões e criar soluções inovadoras. Uma das principais vantagens do pensamento computacional é a capacidade de decompor problemas complexos em partes menores e mais gerenciáveis. Isso facilita a compreensão do problema como um todo e permite que trabalhemos em cada parte separadamente. Ao dividir um problema em partes menores, podemos resolver cada uma delas de forma mais eficiente e integrá-las posteriormente para obter a solução final. Um médico, por exemplo, pode usar o pensamento computacional para diagnosticar doenças, analisando sintomas e identificando padrões. Um arquiteto pode usar o pensamento computacional para projetar edifícios, decompondo o problema em diferentes etapas e criando algoritmos para cada uma delas.

## 1.2 Importância do Pensamento Computacional na Resolução de Problemas

O pensamento computacional é fundamental na resolução de problemas porque nos ajuda a abordar os desafios de forma estruturada e eficiente. De acordo com Wing (2006) nos permite analisar problemas complexos, identificar padrões e criar soluções inovadoras. Uma das principais vantagens do pensamento computacional é a capacidade de decompor problemas complexos em partes menores e mais gerenciáveis. Isso facilita a compreensão do problema como um todo e permite que trabalhemos em cada parte separadamente. Ao dividir um problema em partes menores, podemos resolver cada uma delas de forma mais eficiente e integrá-las posteriormente para obter a solução final. Além disso, o pensamento computacional nos ajuda a reconhecer padrões em diferentes situações. Isso significa que podemos aplicar soluções que funcionaram anteriormente para resolver problemas semelhantes no futuro. Essa capacidade de transferir conhecimento entre diferentes contextos é extremamente valiosa na resolução de problemas complexos. Outra vantagem do pensamento computacional é a capacidade de abstrair problemas complexos, removendo detalhes desnecessários da solução do problema principal. Isso nos permite focar nos aspectos mais importantes do problema e simplificar a solução. Ao remover detalhes irrelevantes, podemos criar soluções mais eficientes e fáceis de entender. Um exemplo de problema que pode ser resolvido utilizando a metodologia de pensamento computacional é a organização e classificação de uma grande quantidade de dados. Isso é particularmente relevante em áreas como análise de dados, gerenciamento de informações, e até mesmo em situações cotidianas, como organizar uma coleção de músicas em uma biblioteca digital.

Vamos considerar um cenário específico:

**Problema:** Você tem uma enorme coleção de músicas em formato digital (MP3) armazenada em uma pasta e deseja organizá-las de maneira eficiente por gênero musical, artista e álbum.

**Solução com Pensamento Computacional:**

1. **Coleta de Dados:** Primeiro, você precisa coletar informações sobre cada música. Isso pode incluir detalhes como o título da música, o nome do artista, o nome do álbum e o gênero musical. Você pode fazer isso manualmente ou

escrever um script para extrair essas informações dos metadados dos arquivos MP3.

2. **Armazenamento de Dados:** Em seguida, você pode criar uma estrutura de dados adequada para armazenar todas essas informações. Isso pode ser uma planilha, um banco de dados ou mesmo uma estrutura de dados personalizada, dependendo da complexidade do seu problema.
3. **Classificação e Organização:** Utilizando algoritmos de ordenação e agrupamento, você pode classificar as músicas por gênero musical, artista e álbum. Isso envolve a escrita de código para comparar e organizar os dados com base em critérios específicos.
4. **Acesso e Recuperação de Dados:** Para tornar sua coleção de músicas acessível, você pode desenvolver um sistema de busca que permite encontrar músicas com base em palavras-chave, artistas ou gêneros. Isso requer a implementação de algoritmos de busca.
5. **Interface do Usuário:** Para tornar o sistema amigável, você pode criar uma interface do usuário (GUI) que permita aos usuários interagir com sua coleção de músicas de maneira intuitiva. Isso pode envolver o desenvolvimento de aplicativos de desktop ou até mesmo aplicativos móveis.
6. **Manutenção e Atualização:** À medida que você adiciona novas músicas à sua coleção ou faz alterações nos metadados, seu sistema deve ser capaz de lidar com essas atualizações de forma eficiente.

Este é apenas um exemplo de como o pensamento computacional pode ser aplicado para resolver um problema do mundo real. Ele envolve a decomposição do problema em etapas menores, a definição de algoritmos para cada etapa e a implementação de código para automatizar o processo. O pensamento computacional é uma abordagem fundamental para resolver uma ampla variedade de problemas, desde organização de dados até tomada de decisões complexas e automação de tarefas. Também promove habilidades importantes, como resolução de problemas, criatividade, trabalho em equipe e pensamento crítico. Ao resolver problemas utilizando o pensamento computacional, somos desafiados a encontrar soluções inovadoras e pensar fora da caixa. Além disso, muitas vezes precisamos trabalhar em equipe para resolver problemas complexos, o que requer habilidades de comunicação e colaboração. A próxima seção trará conceitos de Lógica de

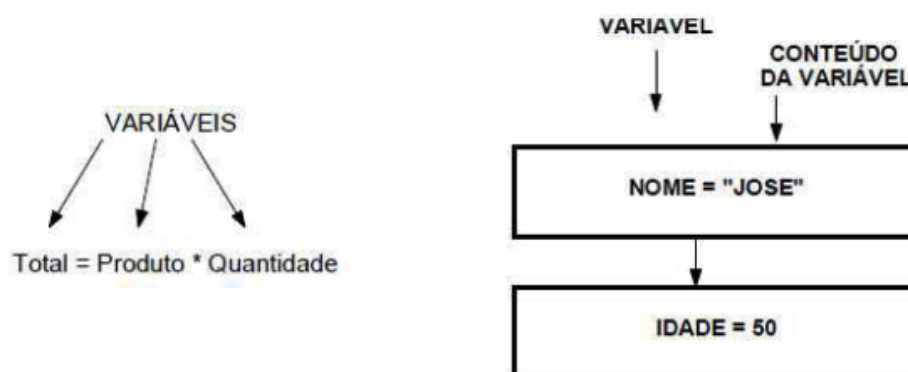
Programação e como esses conceitos podem contribuir para a implementação do Pensamento Computacional.

### 1.3 Conceitos Básicos da Lógica de Programação

A lógica de programação é a base para o desenvolvimento de programas de computador. Ela envolve a aplicação dos princípios do pensamento computacional na criação de algoritmos que resolvam problemas específicos, de acordo com Lopes e Garcia (2002) é a habilidade fundamental que capacita os programadores a desenvolverem algoritmos claros e eficientes, permitindo a resolução sistemática de problemas através de instruções sequenciais, decisões lógicas e iterações controladas. Afirma ainda Lopes e Garcia (2002) que algoritmos são os blocos fundamentais da computação, representando sequências de passos lógicos que solucionam problemas de forma eficiente, possibilitando a automação de tarefas e a tomada de decisões em sistemas computacionais. Sendo um algoritmo uma sequência ordenada de passos que leva à solução de um problema. Ele descreve exatamente o que fazer em cada etapa do processo, permitindo que um programa seja executado corretamente. Existem alguns conceitos básicos da lógica de programação que são essenciais para entender como os algoritmos funcionam, Lopes e Garcia (2002) afirmam que:

- ◆ Variáveis: são espaços na memória do computador onde podemos armazenar valores temporariamente. As variáveis podem ser utilizadas para guardar informações importantes durante a execução do programa.

Figura 1- Variáveis



Fonte: Pesquisa direta

- Estruturas condicionais: permitem que o programa tome decisões com base em condições específicas. Por exemplo, se uma determinada condição for verdadeira, o programa executará um conjunto de instruções; caso contrário, ele executará outro conjunto de instruções.

Código 01- Instruções algorítmicas

```
se media >=5 entao
escrever ("Aluno Aprovado")
fimse
```

Fonte:Pesquisa direta

Código 02- Instruções algorítmicas estrutura condicional

Em algoritmo ficaria  
assim:

```
SE MÉDIA >= 5.0 ENTÃO
ALUNO APROVADO
SENÃO
ALUNO REPROVADO
FIMSE
```

```
se ( media >= 5 ) entao
    escrever ( "Aluno Aprovado" )
senao
    escrever ( "Aluno Reprovado" )
fim se
```

Fonte:Pesquisa direta

- Estruturas de repetição: permitem que o programa execute um conjunto de instruções várias vezes. Isso é útil quando precisamos repetir uma tarefa várias vezes ou quando queremos processar uma grande quantidade de dados. Utilizamos os comandos Enquanto, Para, Repete e Faz.

Código 03-Instruções algorítmicas estrutura de repetição

## Portugol

```

inicio
  variavel inteiro nr
  nr <- 0
  enquanto nr <= 100 faz
    escrever nr , "\t"
    nr <- nr + 1
  fimenquanto
fim

```

Enquanto a variável inteira **nr** for menor ou igual a 100, escrever **nr** igual à **nr** mais um. Neste exemplo implementamos um contador que ira iniciar em 0 e vai ate 100.

Fonte:Pesquisa direta

- Funções: são blocos de código que podem ser reutilizados em diferentes partes do programa. Elas nos permitem organizar o código em partes menores e mais gerenciáveis, facilitando a manutenção e a compreensão do programa como um todo.

Código 04 - Funções

```

Procedimento Soma
variável v1,v2,soma: real
Inicio
Ler v1,v2
Soma<-v1+v2
Escrever soma
fim

```

Fonte:Pesquisa direta

Dominar esses conceitos básicos da lógica de programação é fundamental para criar programas eficientes e organizados. Eles nos ajudam a estruturar nossos algoritmos de forma clara e concisa, tornando-os mais fáceis de entender e manter.

Além dos conceitos básicos, existem muitos outros tópicos avançados na lógica de programação que podem ser explorados.

Alguns exemplos incluem estruturas de dados, recursividade, ordenação e busca. Cada um desses tópicos tem suas próprias aplicações práticas e pode ser estudado em maior profundidade para aprimorar ainda mais as habilidades em programação. A lógica de programação é a base para o desenvolvimento de programas de computador, envolvendo a criação de algoritmos que resolvam problemas específicos. Ela utiliza conceitos como variáveis, estruturas condicionais, estruturas de repetição e funções. A próxima seção abordará a ferramenta Scratch e como ela pode ser útil no contexto da aprendizagem de Lógica de Programação.

## Capítulo 2: A Plataforma Scratch

Esta seção abordará a plataforma Scratch, no qual é entendida como uma linguagem de programação visual e uma comunidade online que permite que pessoas de todas as idades criem projetos interativos, animações e jogos.

### 2.1 Apresentação do Scratch

O Scratch é uma plataforma educacional desenvolvida pelo MIT (Instituto de Tecnologia de Massachusetts) que permite que crianças e iniciantes aprendam programação de forma divertida e interativa. Ele utiliza uma interface gráfica baseada em blocos, o que torna a programação mais acessível e fácil de entender. Uma das principais características do Scratch é a sua simplicidade. De acordo com Bressan (2016) o Scratch é uma modalidade de software que se enquadra na categoria de ambientes de programação visual, especialmente projetados para crianças e iniciantes em programação. Este software permite a criação de projetos interativos, como jogos, histórias animadas e simulações, utilizando uma interface gráfica intuitiva baseada em blocos de código coloridos. Em vez de escrever linhas de código tradicionais, os usuários montam scripts arrastando e soltando blocos de função, facilitando o entendimento da lógica de programação e eliminando a necessidade de memorizar sintaxe complexa. O Scratch é amplamente utilizado em ambientes educacionais para promover o

pensamento computacional, a criatividade e o trabalho em equipe, enquanto introduz conceitos fundamentais de programação de uma maneira divertida e acessível.

Figura 2 - Logo Scratch



Fonte: Scratch.MIT

Ao utilizar o Scratch, os usuários podem criar projetos interativos, como jogos, animações e histórias interativas. Eles podem programar personagens para se moverem na tela, reproduzirem sons e responderem a eventos específicos. Essa abordagem visual estimula a criatividade e permite que os usuários experimentem diferentes ideias sem medo de cometer erros. Além disso, o Scratch possui uma comunidade online ativa onde os usuários podem compartilhar seus projetos, colaborar com outros membros e receber feedback sobre seu trabalho. Isso cria um ambiente colaborativo no qual os usuários podem aprender uns com os outros e se inspirar em projetos criativos feitos por pessoas ao redor do mundo.

O uso do Scratch também promove habilidades importantes além da programação. Ao criar projetos no Scratch, os usuários desenvolvem habilidades



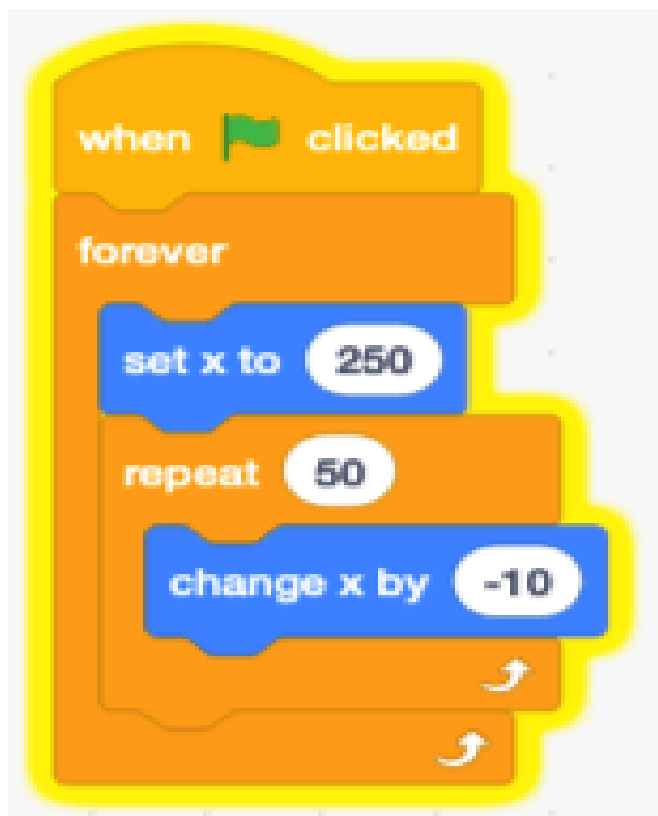
de resolução de problemas, pensamento crítico e trabalho em equipe. Eles aprendem a quebrar problemas complexos em partes menores, a identificar padrões e a criar soluções eficientes.

O Scratch também pode ser usado como uma ferramenta para ensinar conceitos de outras disciplinas, como matemática, ciências e arte. Por exemplo, os usuários podem criar simulações interativas para entender conceitos científicos complexos ou criar animações artísticas para expressar sua criatividade.

## 2.2 Utilizando Blocos Gráficos no Scratch

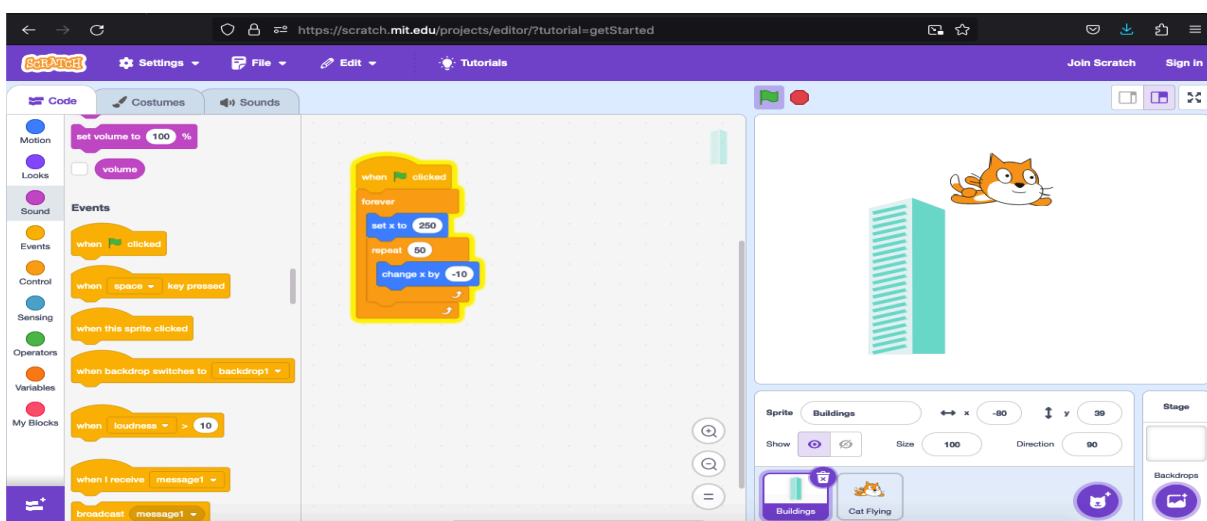
Uma das principais vantagens do Scratch é o uso de blocos gráficos para programar. Esses blocos são organizados em categorias e podem ser arrastados e encaixados para criar programas. Isso torna a programação mais visual e intuitiva, especialmente para iniciantes. Os blocos gráficos do Scratch são compostos por diferentes tipos de comandos, como movimento, aparência, som e controle. Cada bloco possui uma forma específica que indica sua função e uma cor que indica sua categoria. Por exemplo, os blocos de movimento têm uma forma retangular azul, enquanto os blocos de som têm uma forma oval rosa. O bloco apresentado a seguir, faz com que um objeto se mova para frente, repetindo o movimento quantas vezes o usuário informar, por exemplo.

Figura 3 - Conjunto de blocos



Fonte: Scratch.MIT

Figura 4- Blocos de blocos



Fonte: Scratch.MIT

Ao utilizar os blocos gráficos do Scratch, os usuários podem programar personagens para se moverem na tela, mudarem de aparência, reproduzirem sons e responderem a eventos específicos.

Por exemplo, eles podem usar um bloco "mover 10 passos" para fazer um personagem se mover 10 passos para frente ou um bloco "tocar som" para reproduzir um som quando um evento ocorrer. Os blocos gráficos também permitem que os usuários controlem o fluxo do programa usando estruturas condicionais (como "se" e "senão") e loops (como "repita" e "enquanto").

Isso permite que os usuários criem programas mais complexos e interativos. Além disso, o Scratch oferece uma ampla variedade de blocos gráficos para atender às necessidades dos usuários. Existem blocos para lidar com variáveis, listas, operações matemáticas, sensores e muito mais. Essa variedade de blocos permite que os usuários explorem diferentes conceitos de programação e criem projetos mais avançados.

## 2.2 Criando Programas Simples no Scratch

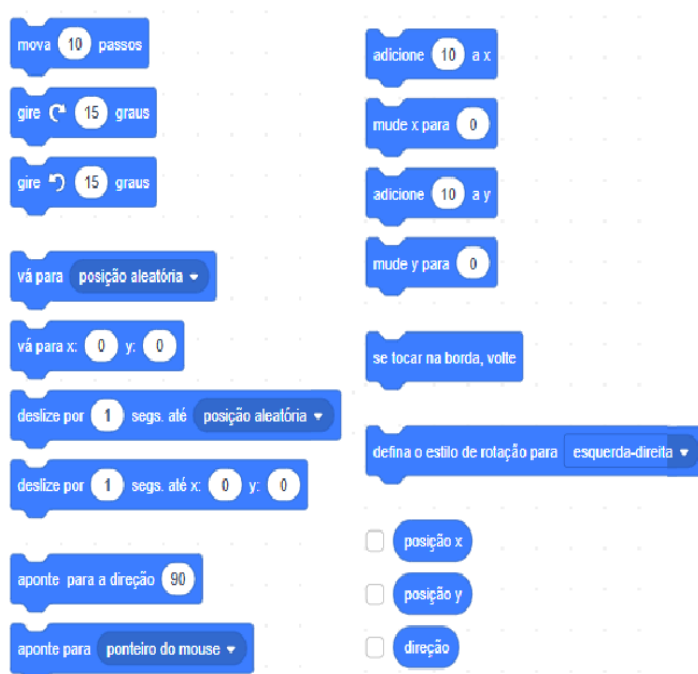
O Scratch é uma plataforma ideal para criar programas simples, especialmente para iniciantes. Com a ajuda dos blocos gráficos do Scratch, os usuários podem criar programas interativos em poucos minutos. Para começar a criar um programa no Scratch, os usuários precisam selecionar um personagem ou sprite para trabalhar. Eles podem escolher entre uma biblioteca de sprites pré-existentes ou criar seu próprio sprite personalizado importando imagens ou desenhando diretamente na plataforma.

Uma vez selecionado o *sprite*, os usuários podem começar a programá-lo arrastando e encaixando os blocos gráficos na área de programação. Por exemplo, eles podem usar um bloco "quando bandeira verde clicada" para iniciar o programa quando a bandeira verde for clicada ou um bloco "esperar 1 segundo" para adicionar um atraso antes de executar determinada ação.

Os usuários também podem usar blocos gráficos para controlar o movimento do sprite na tela. Por exemplo, eles podem usar um bloco "mover 10 passos" para fazer o sprite se mover 10 passos para frente ou um bloco "girar 15 graus" para

girar o sprite em um determinado ângulo.

Figura 5- Blocos de controle de movimento



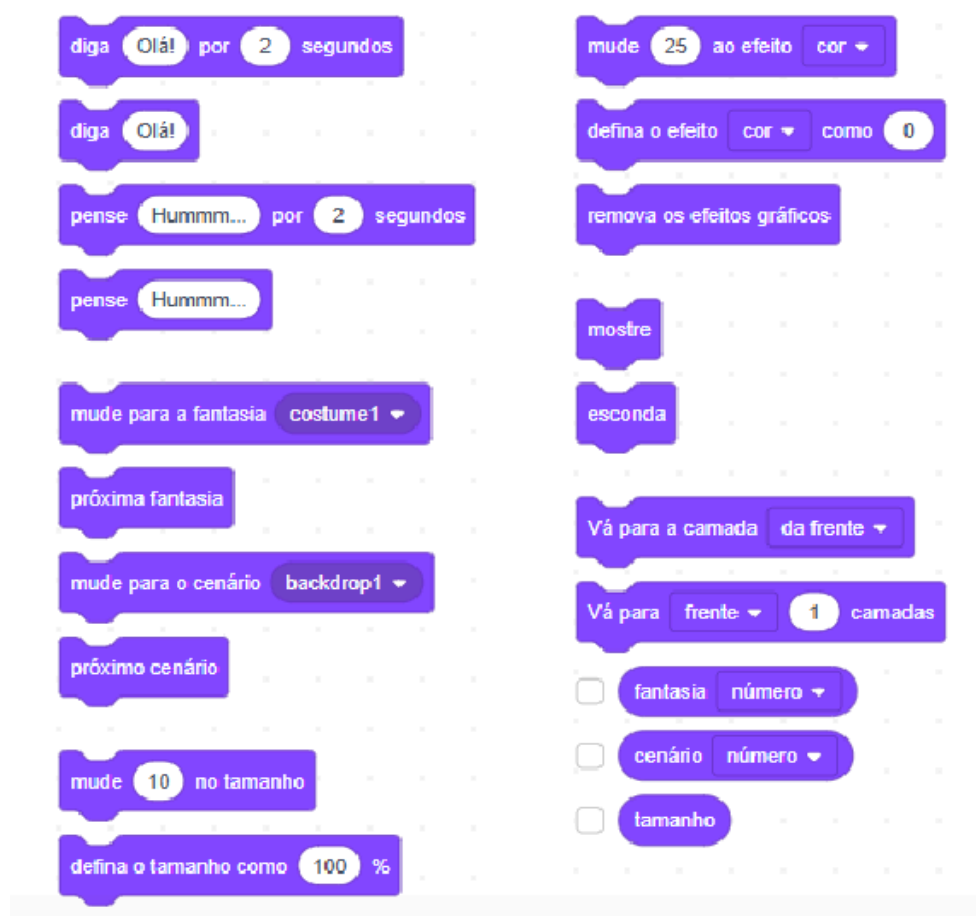
Fonte: Scratch.MIT

Além disso, os usuários podem usar blocos gráficos para alterar a aparência do sprite. Por exemplo, eles podem usar um bloco "mudar efeito cor em 25" para alterar a cor do sprite ou um bloco "trocar traje" para mudar a imagem do sprite.

Os programas criados no Scratch também podem responder a eventos específicos, como cliques de mouse ou pressionamentos de tecla. Os usuários podem usar blocos gráficos como "quando espaço pressionado" ou "quando sprite clicado" para adicionar interatividade aos seus projetos.

Ao criar programas simples no Scratch, os usuários são incentivados a experimentar diferentes combinações de blocos gráficos e testar suas ideias. Eles podem adicionar sons, criar animações e até mesmo compartilhar seus projetos com outras pessoas na comunidade online do Scratch.

Figura 6-Blocos de alteração de conteúdo



Fonte: Scratch.MIT

O Scratch é uma plataforma educacional poderosa que permite que crianças e iniciantes aprendam programação de forma divertida e interativa. Com o uso dos blocos gráficos os usuários podem criar programas simples e complexos, desenvolvendo habilidades importantes como resolução de problemas, pensamento crítico e trabalho em equipe. A seção a seguir trará conceitos mais aprofundados sobre o Scratch e como é possível implementar elementos da Lógica de Programação.

### Capítulo 3: Conceitos Básicos da Linguagem Scratch

No Scratch, os comandos básicos são fundamentais para criar interações e animações. Esta seção apresenta os principais comandos para a utilização da ferramenta.

### 3.1 Comandos e Blocos no Scratch

No Scratch, os comandos são as instruções que dizem ao programa o que fazer. Eles são representados por blocos gráficos coloridos que podem ser encaixados uns nos outros para criar um programa completo. Cada bloco representa uma ação específica, como mover um personagem ou reproduzir um som. Os blocos no Scratch são organizados em diferentes categorias, como Movimento, Aparência, Som e Controle. Cada categoria contém vários blocos relacionados às suas respectivas funcionalidades. Por exemplo, na categoria Movimento, você encontrará blocos para mover um objeto para frente ou para trás, girar ou saltar. Uma das vantagens do uso de blocos gráficos é que eles tornam a programação mais visual e intuitiva, especialmente para iniciantes. Em vez de escrever linhas de código complexas, os usuários podem simplesmente arrastar e soltar os blocos para criar programas interativos.

Além disso, os blocos no Scratch também possuem conectores que permitem controlar a ordem em que as ações são executadas. O usuário pode utilizar um bloco "se" para verificar uma condição antes de executar determinada ação. Isso permite criar programas com lógica condicional e controle de fluxo. Um exemplo prático do uso de comandos e blocos no Scratch é a criação de um jogo simples em que o jogador controla um personagem que deve desviar de obstáculos em movimento. Para isso, você pode usar blocos da categoria Movimento para controlar o movimento do personagem e detectar colisões com os obstáculos usando blocos da categoria Controle.

### 3.2 Variáveis e Operações Matemáticas no Scratch

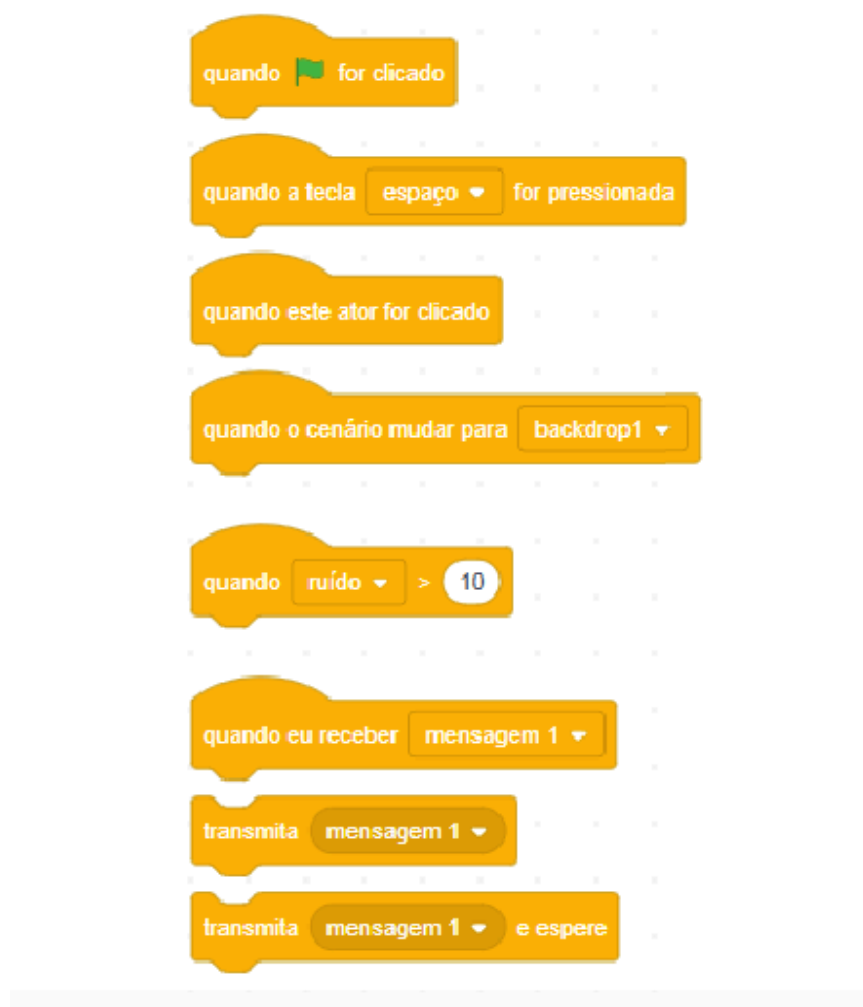
As variáveis são elementos essenciais na programação, pois permitem armazenar e manipular valores. No Scratch, você pode criar suas próprias variáveis e atribuir a elas diferentes tipos de dados, como números, texto ou booleanos. Para criar uma variável no Scratch, basta selecionar a categoria Variáveis e clicar em "Criar uma variável". Em seguida, você pode dar um nome à variável e escolher o tipo de dado que ela irá armazenar. Uma vez criada a variável, você pode usá-la em seu programa para realizar operações matemáticas.

O Scratch oferece uma variedade de blocos para realizar operações como adição, subtração, multiplicação e divisão. Esses blocos podem ser encontrados na categoria Operadores Matemáticos. Por exemplo, se você quiser criar um programa que calcule a média de três números digitados pelo usuário, você pode usar uma variável para armazenar cada número digitado e depois usar os blocos de operações matemáticas para calcular a média. Além disso, o Scratch também permite que você crie expressões matemáticas mais complexas usando parênteses para definir a ordem das operações. Isso é útil quando você precisa realizar cálculos mais avançados ou lidar com fórmulas matemáticas específicas. Um exemplo prático do uso de variáveis e operações matemáticas no Scratch é a criação de um programa que simula uma calculadora simples. Nesse programa, o usuário pode digitar dois números e escolher uma operação (adição, subtração, multiplicação ou divisão) para obter o resultado desejado. As variáveis são usadas para armazenar os números digitados e os blocos de operações matemáticas são usados para realizar a operação selecionada.

### **3.3 Eventos e Controle de Fluxo no Scratch**

No Scratch, os eventos são acionadores que iniciam a execução de um programa. Eles podem ser acionados por ações do usuário, como clicar em um objeto ou pressionar uma tecla, ou por eventos internos do programa, como o início ou término de uma animação.

Figura 7- Blocos de eventos



Fonte: Scratch.MIT

Os eventos no Scratch são representados por blocos especiais que podem ser encontrados na categoria Eventos. Esses blocos permitem que você defina quais ações devem ser executadas quando determinado evento ocorrer. Por exemplo, se você quiser criar um programa em que um personagem mude de cor quando o usuário clicar nele, você pode usar o bloco "quando este objeto for clicado" da categoria Eventos para definir essa ação. Além dos eventos, o Scratch também oferece blocos de controle de fluxo que permitem controlar a ordem em que as ações são executadas. Por exemplo, você pode usar blocos como "se", "senão" e "repita" para criar programas com lógica condicional e repetição.



Figura 8- Blocos de controle de fluxo



Fonte: Scratch.MIT

Estes blocos permitem criar programas mais interativos e dinâmicos. Por exemplo, você pode usar um bloco "se" para verificar se determinada condição é verdadeira antes de executar uma determinada ação. Se a condição for falsa, você pode usar o bloco "senão" para executar outra ação. Um exemplo prático do uso de eventos e controle de fluxo no Scratch é a criação de um jogo simples em que o jogador deve clicar em objetos que aparecem aleatoriamente na tela. Nesse jogo, você pode usar o bloco "quando este objeto for clicado" para detectar quando o jogador clica em um objeto e usar blocos de controle de fluxo para controlar a pontuação e a velocidade dos objetos.

### Aplicação do Pensamento Computacional

Problema: simular um jogo de adivinhação de números. Neste jogo, o jogador tentará adivinhar um número secreto e receberá dicas para ajustar suas suposições.

Implementação:

**Passo 1:** Abra o Scratch e crie um novo projeto.

**Passo 2:** Crie um sprite de "Máquina de Jogo" (você pode desenhar uma máquina ou usar uma imagem de uma máquina de sua escolha).

**Passo 3:** Vá até a categoria "Variáveis" e crie duas variáveis: "Número Secreto" e "Tentativa".

**Passo 4:** Na categoria "Eventos", adicione os seguintes blocos:

- Quando a bandeira verde for clicada:
  - Definir "Número Secreto" como um número aleatório entre 1 e 100 (ou qualquer intervalo de números que você preferir).
  - Definir "Tentativa" como 0 (pois o jogador ainda não fez nenhuma tentativa).

**Passo 5:** Na categoria "Sensores", adicione os seguintes blocos:

- Quando a tecla [espaço] for pressionada:
  - Aumentar "Tentativa" em 1.
  - Peça ao jogador que adivinhe o número e espere até que uma resposta seja inserida usando um bloco "Pedir [Digite sua suposição] e aguarde".
  - Verifique se a suposição do jogador é igual ao "Número Secreto". Se for, exiba uma mensagem de vitória. Caso contrário, exiba uma dica, como "Tente um número maior" ou "Tente um número menor".

**Passo 6:** Repita o bloco "Quando a tecla [espaço] for pressionada" até que o jogador adivinhe corretamente ou atinja um número máximo de tentativas (você pode definir um limite, como 10 tentativas).

**Passo 7:** No final do jogo, adicione uma mensagem para informar ao jogador se ele ganhou ou perdeu.

**Passo 8:** Clique na bandeira verde para iniciar o jogo.

Agora, você criou um jogo de adivinhação de números simples no Scratch que utiliza pensamento computacional e controle de fluxo para permitir que o jogador faça

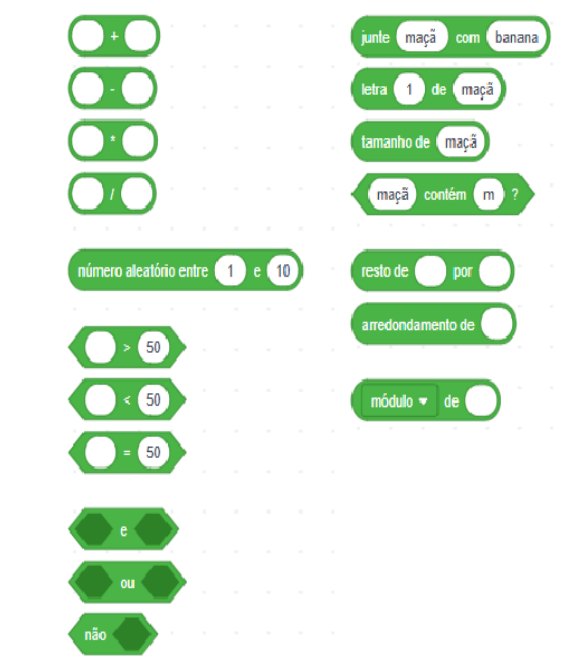
suposições e receba dicas até adivinhar o número secreto ou atingir o limite de tentativas.

Em resumo, os comandos e blocos no Scratch permitem criar programas interativos e visualmente atrativos. As variáveis e operações matemáticas permitem armazenar e manipular valores, enquanto os eventos e o controle de fluxo permitem controlar a ordem das ações executadas. Esses conceitos básicos são fundamentais para aprender lógica de programação utilizando o pensamento computacional e a plataforma Scratch.

### 3.4 Operadores

Tem função matemática e lógica utilizada de forma associativa com os demais grupos de blocos. É importante para gerar as dependências entre as partes do código, pois existem componentes que dependem uns dos outros. A figura a seguir apresenta os blocos de operadores que são usados para realizar operações matemáticas e lógicas, podendo ser combinados com variáveis, constantes e blocos de controle (como condicionais e loops) para criar scripts mais complexos e dinâmicos em projetos do Scratch.

Figura 9- Blocos de operadores



Fonte: Scratch.MIT

## Capítulo 4: Loops e Repetições no Scratch

Em Scratch, os loops são blocos de comando que permitem que um conjunto de instruções seja repetido várias vezes. São úteis para evitar a repetição de código e tornar os scripts mais eficientes e fáceis de manter. Eles são uma parte fundamental da programação e podem ser encontrados na seção "Controle" do conjunto de blocos de comandos do Scratch


### 4.1 Introdução aos Loops no Scratch

Os loops são uma parte fundamental da programação, pois permitem que um conjunto de instruções seja repetido várias vezes. No contexto do Scratch, os loops são usados para criar programas que executam a mesma sequência de ações várias vezes, economizando tempo e esforço. Uma das formas mais comuns de loop no Scratch é o loop "Repetir". Esse bloco permite que você especifique quantas vezes deseja repetir um conjunto de instruções. Por exemplo, se você quiser fazer um personagem se mover para frente e para trás cinco vezes, pode usar o bloco "Repetir" com o valor 5. No entanto, os loops não se limitam apenas a repetições fixas. É possível criar loops personalizados que se adaptam às necessidades específicas do programa. Esses loops personalizados podem ser criados usando variáveis e condicionais.

### 4.2 Utilizando o Bloco "Repetir" no Scratch

O bloco "Repetir" é uma ferramenta poderosa no Scratch, pois permite que você repita um conjunto de instruções várias vezes sem precisar escrever cada instrução separadamente. Isso torna o código mais eficiente e fácil de entender. A figura a seguir apresenta os principais blocos de repetição.

Figura 10- Blocos de Loop

Scratch	Definição
	Repete até a quantidade de vezes estipulada.
	Repete até que uma condição seja atendida.
	Repete infinitamente.

Fonte: Scratch.MIT

Quando o usuário especifica um número muito pequeno, as ações podem não ser executadas tantas vezes quanto necessário. Ao utilizar o bloco "Repetir", é importante definir corretamente o número de repetições desejado. Se você especificar um número muito grande, seu programa pode ficar preso em um *loop* infinito, travando ou consumindo muitos recursos do computador. Desse modo, uma maneira de evitar loops infinitos é usar uma variável como contador. A cada repetição do loop, o valor da variável é incrementado e verificado em uma condição. Quando a condição não for mais atendida, o *loop* será interrompido. Por exemplo, imagine que você queira criar um programa no Scratch que faça um personagem se mover para frente até chegar a uma determinada posição. Você pode usar um *loop* "Repetir" com uma variável contador para controlar o número de vezes que o personagem se move para frente. A cada repetição do loop, o valor da variável contador é incrementado e comparado à posição desejada. Quando o valor da variável for igual ou maior à posição desejada, o loop será interrompido.

### 4.3 Criando Loops Personalizados no Scratch

Além do bloco "Repetir", é possível criar *loops* personalizados no Scratch usando variáveis e condicionais. Esses *loops* personalizados permitem que você

controle as repetições com base em condições específicas. Por exemplo, imagine que o usuário esteja criando um jogo no Scratch em que um personagem precisa coletar moedas em diferentes posições na tela. Em vez de usar um *loop* fixo para percorrer todas as posições possíveis das moedas, você pode criar um *loop* personalizado que verifica se ainda existem moedas a serem coletadas. Para fazer isso, você pode usar uma lista para armazenar as posições das moedas e um *loop* "Repetir até" para verificar se a lista está vazia. Dentro do loop, você pode verificar se o personagem está na mesma posição de alguma das moedas e removê-la da lista caso positivo. Dessa forma, o loop continuará até que todas as moedas tenham sido coletadas. Isso torna o código mais eficiente, pois não é necessário percorrer todas as posições possíveis das moedas a cada repetição do loop. Outra maneira de criar loops personalizados é usando condicionais. Por exemplo, imagine que você esteja criando um programa no Scratch em que um personagem precisa se mover para frente até encontrar uma parede. Você pode usar um loop "Repetir até" com uma condição que verifica se o personagem está colidindo com uma parede. Quando a condição for atendida, o loop será interrompido. Esses são apenas alguns exemplos de como os loops podem ser personalizados no Scratch. A capacidade de criar loops personalizados permite que você crie programas mais flexíveis e adaptáveis às necessidades específicas do seu projeto.

#### **4.4 Utilizando o Pensamento Computacional para criar Loops de Repetição para desenhar uma espiral**

Neste exemplo, usaremos um loop para desenhar repetidamente segmentos de uma espiral enquanto controlamos os ângulos de rotação.

Passo 1: Abra o Scratch e crie um novo projeto.

Passo 2: Crie um sprite que será responsável por desenhar a espiral (por exemplo, você pode usar o sprite padrão do gato).

Passo 3: Vá até a categoria "Blocos" e selecione a categoria "Movimento".

Passo 4: Adicione o seguinte bloco de código para posicionar o sprite inicialmente no centro do palco

vá para x: (0) y: (0)

Passo 5: Ainda na categoria "Movimento", adicione um bloco "repetir" e insira um valor (por exemplo, 360) para controlar quantos segmentos a espiral terá. Isso fará com que o código dentro do loop seja repetido 360 vezes, criando os segmentos da espiral.

Passo 6: Agora, adicione o seguinte bloco dentro do loop:

mover 10 passos, girar 15 graus no sentido horário

Neste bloco, o sprite se move 10 passos para frente e gira 15 graus no sentido horário em cada iteração do loop. Isso cria a aparência de uma espiral. Você pode ajustar o valor de "10" para controlar o comprimento dos segmentos da espiral e o valor de "15" para controlar a quantidade de rotação em cada passo.

Passo 7: Clique na bandeira verde para iniciar o projeto.

Você verá o sprite desenhar uma espiral no palco à medida que repete os passos especificados no loop. A espiral será criada gradualmente, com cada segmento adicionado a cada iteração do loop.

Este exemplo demonstra como usar loops de repetição no Scratch para criar um padrão complexo (nesse caso, uma espiral) e como aplicar o pensamento computacional para controlar o movimento e a rotação do sprite.

Os loops são uma parte essencial da programação e permitem que um conjunto de instruções seja repetido várias vezes. No Scratch, os loops podem ser criados usando o bloco "Repetir" ou através da criação de loops personalizados usando variáveis e condicionais.

Ao utilizar o bloco "Repetir", é importante definir corretamente o número de repetições desejado para evitar loops infinitos ou insuficientes. O uso de variáveis como contadores pode ajudar a controlar as repetições e garantir que o loop seja interrompido quando necessário. Além disso, os loops personalizados permitem maior flexibilidade ao criar programas no Scratch. Eles podem ser usados para controlar a execução de instruções com base em condições específicas, como verificar se todas as moedas foram coletadas ou se o personagem encontrou uma parede.

Os loops de repetições são bastante utilizados estruturas condicionais e de tomada de decisão, o capítulo a seguir tratará de apresentar como funcionam essas estruturas.

## Capítulo 5: Condicionais e Tomada de Decisão no Scratch

As estruturas condicionais são fundamentais na programação, e no Scratch elas permitem que o usuário tome decisões com base em certas condições, desse modo o usuário pode criar interações dinâmicas em seus projetos, permitindo que sprites e cenários reajam de forma inteligente com base em diferentes situações.

### 5.1 Introdução às Condicionais no Scratch

As condicionais são uma parte fundamental da programação, pois permitem que o programa tome decisões com base em certas condições. No Scratch, as condicionais são representadas pelos blocos "Se" e "Senão", que permitem que o programa execute diferentes ações dependendo de uma condição específica. Ao utilizar as condicionais no Scratch, os programadores podem criar programas mais interativos e dinâmicos. Por exemplo, imagine um jogo em que o personagem principal precisa pular sobre obstáculos. Com as condicionais, podemos definir uma ação para o personagem pular apenas quando ele estiver próximo de um obstáculo. Uma das formas mais simples de utilizar as condicionais é através do bloco "Se". Esse bloco permite que o programa execute uma ação somente se uma determinada condição for verdadeira. Por exemplo, podemos usar o bloco "Se" para fazer com que um personagem mude de cor quando clicado pelo usuário. A figura a seguir apresenta um exemplo no qual se o preço for menor que mil, o produto será comprado, caso contrário, não será comprado.

Figura 10- Estrutura condicional





Além disso, também podemos utilizar o bloco "Senão" para definir uma ação alternativa caso a condição não seja verdadeira. Por exemplo, se quisermos fazer com que um personagem se mova para a direita quando pressionarmos a seta direcional do teclado, mas pare quando pressionarmos qualquer outra tecla, podemos usar os blocos "Se" e "Senão".

No entanto, as condicionais no Scratch não se limitam apenas aos blocos "Se" e "Senão". Podemos criar estruturas condicionais mais complexas utilizando operadores lógicos como "E" e "OU". Esses operadores nos permitem combinar várias condições em uma única estrutura condicional. Por exemplo, imagine que queremos criar um jogo em que o personagem principal só pode pular se estiver no chão e não estiver pulando. Podemos utilizar os operadores lógicos "E" para combinar essas duas condições em uma única estrutura condicional.

Além disso, também podemos utilizar os operadores lógicos "OU" para criar estruturas condicionais mais flexíveis. Por exemplo, se quisermos fazer com que um personagem mude de cor quando pressionarmos a tecla A ou a tecla B, podemos utilizar o operador lógico "OU" para combinar essas duas condições.

Figura 11- Condicional lógica

Operador	Exemplo
 Igual (=)	 É o <b>preço</b> igual a <b>1000</b> ?
 Maior (>)	 É o <b>preço</b> maior que <b>1000</b> ?
 Menor(<)	 É o <b>preço</b> menor que <b>1000</b> ?

Em resumo, as condicionais são uma parte fundamental da programação no Scratch. Elas permitem que o programa tome decisões com base em certas condições e criem programas mais interativos e dinâmicos. Além dos blocos "Se" e "Senão", também podemos utilizar operadores lógicos como "E" e "OU" para criar estruturas condicionais mais complexas. Com as condicionais, os programadores podem criar jogos, animações e projetos incríveis no Scratch.

### Utilizando os Blocos "Se" e "Senão" no Scratch

Os blocos "Se" e "Senão" são fundamentais para a criação de condicionais no Scratch. Eles permitem que o programa execute diferentes ações com base em uma condição específica. O bloco "Se" é utilizado para definir uma ação que será executada somente se uma determinada condição for verdadeira. Por exemplo, podemos utilizar o bloco "Se" para fazer com que um personagem mude de cor quando clicado pelo usuário. Nesse caso, a condição seria o clique do usuário no personagem. Já o bloco "Senão" é utilizado para definir uma ação alternativa caso a condição do bloco "Se" não seja verdadeira. Por exemplo, se quisermos fazer com que um personagem se mova para a direita quando pressionarmos a seta direcional do teclado, mas pare quando pressionarmos qualquer outra tecla, podemos usar os blocos "Se" e "Senão". Além disso, também podemos utilizar os blocos "Se" e "Senão" em conjunto com outros blocos e variáveis para criar programas mais complexos. Por exemplo, imagine que queremos criar um jogo em que o personagem principal só pode pular se estiver no chão e não estiver pulando. Podemos utilizar os blocos "Se" e "Senão", juntamente com variáveis para controlar o estado do personagem. No entanto, é importante lembrar que as condicionais no Scratch são avaliadas de cima para baixo. Isso significa que o programa irá verificar cada bloco condicional na ordem em que eles aparecem no código. Portanto, é importante organizar as condicionais de forma lógica e coerente.

## 5.2 Criando Estruturas Condicionais Complexas no Scratch

Além dos blocos "Se" e "Senão", o Scratch também permite a criação de estruturas condicionais mais complexas utilizando operadores lógicos como "E" e

"OU". Esses operadores nos permitem combinar várias condições em uma única estrutura condicional. O operador lógico "E" é utilizado para combinar duas ou mais condições, fazendo com que todas elas precisem ser verdadeiras para que a estrutura condicional seja executada. Por exemplo, imagine que queremos criar um jogo em que o personagem principal só pode pular se estiver no chão e não estiver pulando. Podemos utilizar o operador lógico "E" para combinar essas duas condições em uma única estrutura condicional. Já o operador lógico "OU" é utilizado para combinar duas ou mais condições, fazendo com que pelo menos uma delas precise ser verdadeira para que a estrutura condicional seja executada. Por exemplo, se quisermos fazer com que um personagem mude de cor quando pressionarmos a tecla A ou a tecla B, podemos utilizar o operador lógico "OU" para combinar essas duas condições.

Além disso, também podemos utilizar os operadores lógicos "E" e "OU" em conjunto com os blocos "Se" e "Senão" para criar estruturas condicionais ainda mais complexas. Por exemplo, imagine que queremos criar um jogo em que o personagem principal só pode pular se estiver no chão e não estiver pulando, mas também só pode pular se a pontuação do jogador for maior do que 100 pontos. Podemos utilizar os operadores lógicos "E" e "OU", juntamente com os blocos "Se" e "Senão", para combinar todas essas condições em uma única estrutura condicional.

Aplicação do Pensamento Computacional: Vamos criar um exemplo no Scratch que utilize a estrutura condicional de decisão "SE" (if) para simular um jogo de adivinhação de cores. Neste jogo, o jogador tentará adivinhar uma cor específica e receberá feedback sobre sua suposição. Usaremos pensamento computacional para criar esse jogo.

**Passo 1:** Abra o Scratch e crie um novo projeto.

**Passo 2:** Crie um sprite que representará a cor a ser adivinhada. Por exemplo, você pode desenhar um círculo colorido e nomeá-lo "Cor a Adivinhar".

**Passo 3:** Na categoria "Variáveis", crie uma variável chamada "Tentativa" para rastrear o número de tentativas do jogador.

**Passo 4:** Na categoria "Eventos", adicione os seguintes blocos:

- Quando a bandeira verde for clicada:
  - Definir "Tentativa" como 0 (pois o jogador ainda não fez nenhuma tentativa).
  - Escolher uma cor aleatória para a variável "Cor a Adivinhar" (você pode usar o bloco "Definir [Cor a Adivinhar] como [cor aleatória]" na categoria "Operadores").

**Passo 5:** Na categoria "Sensores", adicione o seguinte bloco:

- Quando a bandeira verde for clicada:
  - Peça ao jogador que adivinhe a cor usando um bloco "Pedir [Qual é a cor?] e aguarde" e armazene a resposta na variável "Resposta do Jogador".

**Passo 6:** Adicione uma estrutura condicional "SE" (if) na categoria "Controle" após o bloco de pedir a resposta do jogador:

- Se "Resposta do Jogador" = "Cor a Adivinhar", então:
  - Mostre uma mensagem de vitória, como "Parabéns! Você adivinhou a cor correta!".
  - Encerre o jogo.
- Caso contrário, dentro do bloco "SE", adicione uma mensagem informando que a suposição está incorreta, como "Tente novamente."

**Passo 7:** Após a estrutura condicional "SE", adicione um bloco para aumentar a variável "Tentativa" em 1.

**Passo 8:** Você pode adicionar uma condição para limitar o número de tentativas. Por exemplo, você pode criar uma nova variável chamada "Tentativas Restantes" e diminuí-la em 1 a cada tentativa, exibindo a quantidade restante de tentativas ao jogador.

**Passo 9:** Clique na bandeira verde para iniciar o jogo.

Agora, você criou um jogo de adivinhação de cores simples no Scratch que utiliza a estrutura condicional "SE" para verificar se a suposição do jogador está correta ou não. O jogo continuará até que o jogador visualize corretamente ou até que o número de tentativas seja excedido, dependendo de como você configurou as variáveis. Como apresentado, condicionais e tomada de decisão no Scratch permitem que o usuário controle o comportamento dos seus projetos com base em certas condições. Com os blocos "Se-Então" e "Se-Então-Senão", o usuário pode criar ramificações no seu código, onde certas ações são executadas apenas se uma condição específica for verdadeira. Isso permite criar interações dinâmicas e adaptáveis em seus projetos.

Desse modo, pode-se utilizar operadores de comparação para verificar igualdades, desigualdades e outras condições lógicas. Além disso, é possível aninhar condicionais para criar uma lógica mais complexa e lidar com diferentes cenários. O próximo capítulo abordará como implementar tipos de Lista de Dados utilizando Scratch.

## **Capítulo 6: Trabalhando com Listas e Dados no Scratch**

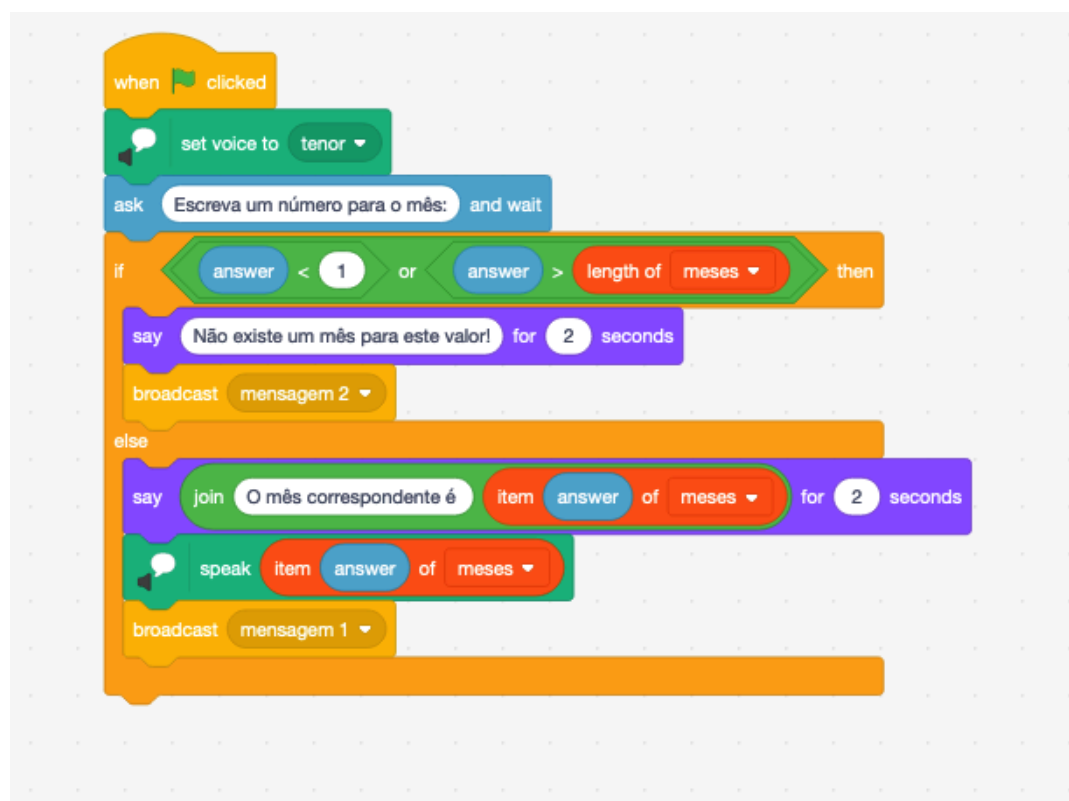
No Scratch, a manipulação de listas de dados permite armazenar e organizar conjuntos de informações em uma estrutura sequencial. É uma habilidade importante para criar projetos mais sofisticados e dinâmicos. Com a prática e o domínio desses conceitos, nas seções a seguir, serão apresentados conceitos detalhados sobre.

### **6.1 Manipulando Listas de Dados no Scratch**

As listas são uma estrutura de dados fundamental na programação, e o Scratch oferece recursos poderosos para manipular e trabalhar com elas. Ao utilizar listas no Scratch, os usuários podem armazenar e organizar informações de forma eficiente, permitindo a criação de programas mais complexos e dinâmicos. Tanenbaum (2019) afirma que Listas ligadas representam uma estrutura dinâmica de dados fundamental na computação, permitindo uma organização flexível e eficiente de elementos onde cada nó contém um valor e um ponteiro para o próximo nó na sequência, possibilitando operações de inserção, remoção e acesso aos dados de forma escalável."

Uma das principais vantagens de usar listas no Scratch é a capacidade de armazenar múltiplos valores em uma única variável. No exemplo a seguir, temos uma cadeia de blocos que tem por objetivo armazenar dados na lista e depois recuperá-los, onde o usuário entrará com um mês do ano e terá como resposta qual é o mês correspondente, ou seja, é criada uma variável para armazenar os meses do ano, inicializando-a com os respectivos valores, o usuário insere o mês e depois é verificado se esse mês está na lista e qual é a sua posição.

Figura 11- Lista de armazenamento



Fonte: Scratch.MIT

Além disso, o Scratch oferece uma variedade de blocos específicos para

manipular listas. O usuário pode adicionar itens a uma lista utilizando o bloco "adicionar [valor] à [lista]", remover itens usando o bloco "remover [valor] da [lista]" ou verificar se um item está presente em uma lista usando o bloco "contém [valor] em [lista]".

Além disso, o Scratch também permite percorrer uma lista usando loops. Por exemplo, você pode usar um loop "para cada [item] em [lista]" para executar uma determinada ação para cada item na lista. Isso é especialmente útil quando você precisa realizar a mesma operação em todos os itens da lista, como exibir todos os elementos em uma tela ou calcular a média de uma série de números. Para ilustrar a importância e utilidade das listas no Scratch, vamos considerar um exemplo prático, a possibilidade de gerenciar uma biblioteca virtual ao desenvolver uma aplicação. Imagine que você está criando um programa para gerenciar uma biblioteca virtual. Você pode criar uma lista chamada "livros" para armazenar informações sobre os livros disponíveis na biblioteca, como título, autor e ano de publicação. Com essa lista, você pode adicionar novos livros, remover livros existentes e até mesmo pesquisar por livros específicos com base em critérios como autor ou ano de publicação.

## Capítulo 7: Desenvolvimento de Programas Eficientes e Organizados

Desenvolver programas no Scratch pode ser uma atividade divertida e educativa, especialmente para iniciantes em programação.

### 7.1 Boas Práticas para o Desenvolvimento de Programas no Scratch

Uma das principais boas práticas é a modularização do código. Isso significa dividir o programa em pequenas partes ou módulos, cada um responsável por uma tarefa específica. Essa abordagem facilita a compreensão do programa como um todo e permite reutilizar partes do código em diferentes projetos. Por exemplo, se você estiver criando um jogo no Scratch, pode criar módulos separados para lidar com a movimentação dos personagens, detecção de colisões e pontuação. Outra boa prática é utilizar nomes descritivos para variáveis, sprites e blocos de código. Isso torna mais fácil entender o propósito de cada elemento do programa e ajuda na manutenção futura. Por exemplo, em vez de usar nomes genéricos como

"variável1" ou "sprite2", é melhor usar nomes mais descritivos como "pontuacao" ou "personagemPrincipal". Além disso, é importante comentar o código adequadamente. Comentários são trechos de texto que explicam o que determinada parte do código faz. Eles ajudam outros programadores (ou até mesmo você mesmo no futuro) a entenderem a lógica por trás do programa. É recomendado adicionar comentários sempre que o código não for autoexplicativo. Desse modo, se o usuário estiver utilizando uma fórmula matemática complexa para calcular algo, é útil adicionar um comentário explicando a lógica por trás dessa fórmula.

Outra boa prática é evitar a repetição de código. Se você perceber que está escrevendo o mesmo trecho de código várias vezes, é melhor criar uma função ou bloco personalizado para essa tarefa e reutilizá-lo sempre que necessário. Isso torna o programa mais eficiente e facilita a manutenção, pois qualquer alteração precisa ser feita apenas em um único lugar. Também é importante testar o programa regularmente durante o desenvolvimento. Isso ajuda a identificar erros e problemas antes que eles se tornem mais difíceis de corrigir. O Scratch oferece ferramentas de depuração, como a visualização do valor das variáveis em tempo real e a possibilidade de pausar a execução do programa para analisar seu estado. Utilize essas ferramentas para identificar e corrigir problemas no seu código.

Por fim, é recomendado documentar o programa adequadamente. Isso inclui fornecer informações sobre como usar o programa, quais são suas funcionalidades principais e quaisquer requisitos específicos. Essa documentação pode ser útil para outros usuários do programa ou até mesmo para você mesmo no futuro. Desse modo, boas práticas para o desenvolvimento de programas no Scratch incluem modularização do código, uso de nomes descritivos, comentários adequados, evitar repetição de código, testes regulares durante o desenvolvimento e documentação adequada do programa. Seguir essas práticas ajudará a criar programas mais eficientes, organizados e fáceis de entender e manter. Uma das principais formas de estruturar o código no Scratch é através do uso adequado dos blocos gráficos disponíveis na plataforma. Os blocos devem ser organizados em sequência lógica, facilitando assim a compreensão da ordem em que as instruções serão executadas. Por exemplo, se você estiver criando um jogo no Scratch, pode organizar os blocos relacionados à movimentação dos



personagens em uma sequência lógica separada dos blocos relacionados à detecção de colisões. Além disso, é importante utilizar espaços vazios entre os blocos para melhorar a legibilidade do código. Isso ajuda a evitar que os blocos fiquem muito próximos uns dos outros, dificultando a leitura. Os espaços vazios também podem ser usados para agrupar blocos relacionados, facilitando a identificação de partes específicas do código.

Outra forma de estruturar o código é através da criação de funções ou blocos personalizados. Esses blocos podem encapsular um conjunto de instruções que são executadas repetidamente em diferentes partes do programa. Por exemplo, se você estiver criando um jogo no Scratch e precisar verificar se o jogador perdeu uma vida, pode criar um bloco personalizado chamado "verificarVida" que contém as instruções necessárias para essa verificação. Dessa forma, você pode reutilizar esse bloco em diferentes partes do programa sempre que necessário. Além da estruturação do código, a documentação adequada também desempenha um papel importante na compreensão e manutenção do programa. É recomendado adicionar comentários explicativos ao longo do código para fornecer informações adicionais sobre o propósito de determinadas partes ou instruções. Esses comentários ajudam outros programadores (ou até mesmo você mesmo no futuro) a entenderem a lógica por trás do programa. Além dos comentários, é útil criar uma descrição detalhada do projeto no Scratch. Essa descrição deve incluir informações sobre como usar o programa, quais são suas funcionalidades principais e quaisquer requisitos específicos. Essa documentação pode ser acessada pelos usuários do programa e ajuda a evitar confusões ou dúvidas sobre seu funcionamento.

Desse modo, você pode adicionar comentários explicativos ao longo do código para fornecer informações adicionais sobre as operações matemáticas. Por exemplo, você pode adicionar um comentário explicando como funciona a operação de multiplicação e quais são os possíveis resultados. Por fim, é importante criar uma descrição detalhada do projeto no Scratch. Essa descrição deve incluir informações sobre como usar o programa (por exemplo, como inserir os números e selecionar a operação desejada), quais são as funcionalidades principais (por exemplo, realizar cálculos matemáticos) e quaisquer requisitos específicos (por exemplo, ter o Adobe Flash Player instalado). A estruturação e

documentação adequadas do código no Scratch são fundamentais para garantir que o código seja legível, compreensível e fácil de manter. Isso inclui organizar os blocos gráficos em sequência lógica, utilizar espaços vazios entre os blocos, criar funções ou blocos personalizados para reutilização de código, adicionar comentários explicativos ao longo do código e criar uma descrição detalhada do projeto no Scratch.

## **7.2 Testes e Depuração de Programas no Scratch**

No capítulo 7.3, o livro "Pensamento Computacional e Scratch: um guia de boas práticas para aprendizagem de Lógica de Programação" explora a importância dos testes e da depuração de programas no Scratch. Essas etapas são cruciais para identificar erros e problemas no código, garantindo que o programa funcione corretamente. Os testes são realizados para verificar se o programa está funcionando conforme o esperado. Existem diferentes tipos de testes que podem ser aplicados no Scratch, como testes unitários, testes funcionais e testes de integração. Os testes unitários são usados para verificar se cada parte individual do programa funciona corretamente. Por exemplo, se você estiver criando um jogo no Scratch, pode realizar um teste unitário para verificar se a movimentação do personagem principal está funcionando adequadamente. Os testes funcionais são usados para verificar se o programa como um todo funciona corretamente. Por exemplo, você pode realizar um teste funcional em um jogo para garantir que todas as funcionalidades principais estão operando corretamente (por exemplo, detecção de colisões, pontuação). Os testes de integração são usados para verificar se as diferentes partes do programa trabalham bem juntas. Por exemplo, você pode realizar um teste de integração em um jogo para garantir que a movimentação do personagem principal esteja sincronizada com a detecção de colisões. Além dos testes, a depuração é uma etapa importante para identificar e corrigir erros no código. A depuração envolve a análise do programa em execução para identificar onde ocorrem os erros e como corrigi-los. O Scratch oferece ferramentas de depuração que podem ser usadas durante o desenvolvimento do programa.

Por exemplo, você pode usar a visualização do valor das variáveis em tempo

real para verificar se elas estão sendo atualizadas corretamente. Você também pode pausar a execução do programa em pontos específicos para analisar seu estado e identificar possíveis problemas. Além disso, é útil adicionar blocos de código que exibam mensagens de erro ou informações relevantes durante a execução do programa. Essas mensagens podem ajudar na identificação de problemas e fornecer informações úteis para a depuração. Para ilustrar a importância dos testes e da depuração no Scratch, vamos considerar um exemplo prático. Suponha que você esteja criando um programa no Scratch para simular o movimento dos planetas ao redor do sol. Nesse programa, você pode realizar testes unitários para verificar se cada planeta está se movendo corretamente em sua órbita específica.

Além disso, é útil adicionar blocos de código que exibam mensagens de erro ou informações relevantes durante a execução do programa. Essas mensagens podem ajudar na identificação de problemas e fornecer informações úteis para a depuração. Para ilustrar a importância dos testes e da depuração no Scratch, vamos considerar um exemplo prático. Suponha que você esteja criando um programa no Scratch para simular o movimento dos planetas ao redor do sol. Nesse programa, você pode realizar testes unitários para verificar se cada planeta está se movendo corretamente em sua órbita específica. Além disso, o pensamento computacional também pode ser aplicado no desenvolvimento de experimentos científicos. Os cientistas podem usar técnicas de programação para controlar equipamentos laboratoriais automatizados, coletar dados em tempo real e ajustar parâmetros experimentais de forma eficiente. O pensamento computacional desempenha um papel fundamental na ciência, permitindo que os pesquisadores resolvam problemas complexos, analisem grandes volumes de dados e desenvolvam modelos computacionais para simular fenômenos naturais. Essa abordagem tem o potencial de acelerar a descoberta científica e impulsionar avanços em diversas áreas do conhecimento. Para aqueles interessados em se aprofundar no assunto, recomenda-se a leitura de livros como "Computational Thinking for the Modern Problem Solver" de David D. Riley e "Computational Thinking: A Beginner's Guide to Problem Solving and Programming" de Karl Beecher.

### 7.3 Uso do Pensamento Computacional na Arte e Design

O pensamento computacional também desempenha um papel importante na arte e design, permitindo aos artistas explorarem novas formas de expressão criativa e criarem obras interativas e dinâmicas. Uma das aplicações mais evidentes do pensamento computacional na arte é a criação de arte digital. Os artistas podem usar algoritmos e técnicas de programação para gerar imagens, animações ou até mesmo esculturas digitais. Essa abordagem permite que eles explorem padrões complexos, simetrias ou até mesmo criem obras que mudam ao longo do tempo.

Um exemplo famoso é o trabalho do artista digital Aaron Koblin, que usa dados coletados da internet para criar visualizações interativas. Em uma de suas obras, ele mapeou os voos comerciais em todo o mundo ao longo de 24 horas, criando uma representação visual fascinante dos movimentos globais. Além disso, o pensamento computacional também pode ser aplicado no design de produtos e interfaces interativas. Os designers podem usar técnicas de programação para criar protótipos digitais, testar diferentes interações e ajustar o design com base no feedback dos usuários. Um exemplo prático disso é o desenvolvimento de jogos eletrônicos. Os game designers usam o pensamento computacional para criar mecânicas de jogo, definir regras e implementar a lógica do jogo. Eles também podem usar algoritmos para gerar conteúdo proceduralmente, como níveis ou personagens. O capítulo a seguir trará exemplos e conceitos de como o Pensamento Computacional pode ser aplicado em nosso dia a dia.

## Capítulo 8: Aplicações do Pensamento Computacional na Vida Pessoal e Profissional

O pensamento computacional é uma habilidade fundamental que vai além da programação e tem diversas aplicações na vida pessoal e profissional. É uma habilidade versátil e essencial que pode ser aplicada em uma ampla variedade de contextos na vida pessoal e profissional, contribuindo para o sucesso e o desenvolvimento pessoal e profissional das pessoas. As seções a seguir, tratarão do tema em detalhes.

## **8.1 Utilizando o Pensamento Computacional para Resolver Problemas Cotidianos**

O pensamento computacional não se limita apenas ao campo da programação e da tecnologia. Na verdade, ele pode ser aplicado em várias áreas da vida cotidiana para resolver problemas de forma estruturada e eficiente. Uma das principais habilidades do pensamento computacional é a decomposição, que envolve dividir um problema complexo em partes menores e mais gerenciáveis. Isso pode ser útil em situações do dia a dia, como planejar uma viagem ou organizar um evento. Ao decompor o problema em etapas menores, é mais fácil identificar as tarefas necessárias e estabelecer uma sequência lógica de ações. Outro aspecto importante do pensamento computacional é o reconhecimento de padrões. Essa habilidade permite identificar tendências ou regularidades em dados ou eventos, o que pode ser útil para tomar decisões informadas. Por exemplo, ao analisar os gastos mensais, podemos identificar padrões de consumo e ajustar nosso orçamento de acordo.

A abstração também desempenha um papel fundamental no pensamento computacional. Ela envolve simplificar um problema complexo, removendo detalhes desnecessários e focando nos aspectos essenciais. Isso pode ser aplicado na resolução de problemas cotidianos, como encontrar soluções criativas para economizar tempo ou recursos. Além disso, o uso de algoritmos no pensamento computacional pode ajudar a otimizar processos diários. Um algoritmo é uma sequência ordenada de instruções que leva à solução de um problema específico. Ao seguir uma receita de culinária, estamos seguindo um algoritmo para preparar uma refeição. Ao aplicar o pensamento computacional, podemos criar algoritmos para resolver problemas cotidianos de maneira mais eficiente. Um exemplo prático do uso do pensamento computacional na vida cotidiana é a organização de tarefas diárias. Ao utilizar técnicas como a decomposição e a abstração, podemos dividir nossas tarefas em etapas menores e priorizá-las com base em sua importância e urgência. Isso nos ajuda a gerenciar nosso tempo de forma mais eficaz e aumentar nossa produtividade. Outro exemplo é o planejamento de viagens. Ao utilizar o pensamento computacional, podemos decompor o processo em etapas como pesquisa de destinos, reserva de

passagens e hospedagem, criação de um itinerário e organização das atividades durante a viagem. Isso nos permite planejar cada aspecto da viagem de forma estruturada e garantir que nada saia do planejado.

## **8.2 O Papel do Pensamento Computacional nas Carreiras Tecnológicas**

O pensamento computacional desempenha um papel fundamental nas carreiras tecnológicas. À medida que a tecnologia continua avançando rapidamente, as habilidades relacionadas ao pensamento computacional se tornam cada vez mais importantes para os profissionais que desejam ter sucesso nesse campo. Uma das principais habilidades do pensamento computacional é a resolução de problemas. Nas carreiras tecnológicas, os profissionais são frequentemente confrontados com desafios complexos que exigem uma abordagem estruturada e lógica para encontrar soluções eficientes. O pensamento computacional fornece as ferramentas necessárias para analisar problemas, identificar padrões e desenvolver algoritmos para resolvê-los.

Além disso, o pensamento computacional também promove a criatividade. Na área da tecnologia, muitas vezes é necessário pensar fora da caixa e encontrar soluções inovadoras para problemas complexos. Através do uso de técnicas como a abstração e o reconhecimento de padrões, os profissionais podem encontrar novas maneiras de resolver problemas e desenvolver produtos ou serviços únicos. O trabalho em equipe também é essencial nas carreiras tecnológicas, e o pensamento computacional pode ajudar nesse aspecto. Ao decompor um problema complexo em partes menores, os profissionais podem atribuir tarefas específicas a membros da equipe com base em suas habilidades individuais. Isso permite uma divisão clara do trabalho e facilita a colaboração entre os membros da equipe. Outra vantagem do pensamento computacional nas carreiras tecnológicas é sua aplicabilidade em várias áreas. O pensamento computacional não se limita apenas à programação; ele pode ser aplicado em campos como análise de dados, segurança cibernética, inteligência artificial e design de interfaces de usuário. Portanto, desenvolver habilidades de pensamento computacional pode abrir portas para uma ampla gama de oportunidades

profissionais. A aplicação do papel do pensamento computacional nas carreiras tecnológicas é o desenvolvimento de software. Ao utilizar técnicas como a decomposição e a abstração, os programadores podem dividir um projeto em módulos menores e desenvolvê-los separadamente. Isso permite que eles trabalhem de forma mais eficiente e colaborem com outros membros da equipe para criar um produto final de alta qualidade. Outro exemplo é a análise de dados. Ao aplicar o pensamento computacional, os analistas podem decompor conjuntos complexos de dados em partes menores e identificar padrões ou tendências significativas. Isso ajuda as empresas a tomar decisões informadas com base em evidências concretas e maximizar seu desempenho.

### **8.3 Benefícios do Desenvolvimento do Pensamento Computacional na Vida Profissional**

O desenvolvimento do pensamento computacional traz uma série de benefícios para a vida profissional. Essa habilidade não se limita apenas às carreiras tecnológicas; ela pode ser aplicada em diversas áreas profissionais para melhorar o desempenho e impulsionar o sucesso. Uma das principais vantagens do pensamento computacional é a capacidade de resolver problemas complexos de forma estruturada e eficiente. No ambiente profissional, os desafios podem surgir em diferentes formas, desde problemas operacionais até questões estratégicas. Ao aplicar o pensamento computacional, os profissionais podem analisar esses problemas, identificar padrões e desenvolver soluções lógicas e eficazes. Além disso, o pensamento computacional promove a criatividade no ambiente de trabalho. Ao utilizar técnicas como a abstração e o reconhecimento de padrões, os profissionais podem encontrar novas maneiras de abordar problemas e desenvolver soluções inovadoras. Isso pode levar a melhorias significativas nos processos internos da empresa ou na criação de produtos ou serviços únicos.

Outro benefício do desenvolvimento do pensamento computacional é a melhoria das habilidades de comunicação e colaboração. O pensamento computacional envolve decompor um problema complexo em partes menores, o que facilita a atribuição de tarefas específicas aos membros da equipe com base

em suas habilidades individuais. Isso promove uma divisão clara do trabalho e facilita a colaboração entre os membros da equipe para alcançar objetivos comuns. Além disso, o pensamento computacional também ajuda no desenvolvimento do pensamento crítico. Ao analisar problemas complexos e desenvolver soluções lógicas, os profissionais são incentivados a questionar suposições e avaliar diferentes perspectivas. Isso permite uma tomada de decisão mais informada e ajuda a evitar erros ou falhas potenciais. Um exemplo prático dos benefícios do desenvolvimento do pensamento computacional na vida profissional é a gestão de projetos. Ao utilizar técnicas como a decomposição e o uso de algoritmos, os gerentes de projeto podem dividir um projeto em tarefas menores e estabelecer uma sequência lógica de atividades. Isso ajuda a garantir que o projeto seja concluído dentro do prazo e do orçamento estabelecidos. Outro exemplo é a resolução de problemas operacionais. Ao aplicar o pensamento computacional, os profissionais podem analisar dados complexos, identificar padrões ou tendências significativas e tomar decisões informadas com base nessas informações. Isso pode levar a melhorias nos processos internos da empresa, aumentando sua eficiência e produtividade. Em resumo, o desenvolvimento do pensamento computacional traz uma série de benefícios para a vida profissional, incluindo habilidades de resolução de problemas, criatividade, comunicação e colaboração, pensamento crítico melhoria dos processos internos da empresa. Essas habilidades são valiosas em qualquer área profissional e podem ajudar os profissionais a alcançarem sucesso. O capítulo a seguir traz uma conclusão a partir da análise do que foi apresentado até então, desenvolvendo reflexões sobre o tema Pensamento Computacional e como pode ser aplicado em diversos contextos.

## Capítulo 9: Conclusão

No decorrer do livro: “um guia de boas práticas: Pensamento Computacional utilizando *Scratch*”, foram abordados diversos conceitos fundamentais para o desenvolvimento dessa habilidade essencial na era digital. Vamos recapitular alguns dos principais conceitos discutidos ao longo do livro.

### 9. Recapitulação dos Principais Conceitos Abordados



O pensamento computacional foi apresentado como uma forma estruturada de resolver problemas, utilizando conceitos como decomposição, reconhecimento de padrões, abstração e algoritmos. Esses conceitos são a base para a resolução eficiente de problemas complexos, tanto na programação quanto em outras áreas da vida. A plataforma Scratch foi introduzida como uma ferramenta educacional poderosa para ensinar programação para crianças e iniciantes. Através do uso de blocos gráficos, os leitores aprenderam passo a passo como criar programas simples e desenvolver suas habilidades em lógica de programação. Ao longo do livro, foram explorados desde os conceitos básicos da linguagem Scratch até tópicos mais avançados, como loops, condicionais e variáveis. Os leitores puderam compreender como utilizar esses recursos para criar programas mais complexos e funcionais. Além disso, o livro também apresentou boas práticas para o desenvolvimento de programas eficientes e organizados. Os leitores aprenderam a importância da clareza no código, da utilização adequada das estruturas de controle e da criação de algoritmos otimizados. Os exercícios práticos propostos ao longo do livro permitiram que os leitores aplicassem os conhecimentos adquiridos e desenvolvessem suas habilidades em raciocínio lógico e criatividade. Esses exercícios estimularam a resolução de problemas de forma autônoma, incentivando os leitores a encontrar suas próprias soluções. Além dos conceitos técnicos, o livro também apresentou estudos de caso e exemplos reais de aplicação do pensamento computacional em diferentes áreas, como ciência, arte e jogos. Isso ajudou os leitores a entenderem como o pensamento computacional pode ser utilizado para resolver problemas do mundo real.

## REFERÊNCIAS BIBLIOGRÁFICAS

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, 72, 145-157.

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In Annual American Educational Research Association Conference.

Brennan, K., & Resnick, M. (2012). Preparing and supporting teachers to implement Scratch. In Proceedings of the 43rd ACM technical symposium on Computer Science Education (pp. 141-146).

Grover, S., & Pea, R. (2013). Computational thinking in K-12: A review of the state of the field. *Educational Researcher*, 42(1), 38-43.

Grover, S., & Pea, R. (2013). Using exploratory data analysis to support teaching and learning about variation in computer models. *International Journal of Computers for Mathematical Learning*, 18(2), 173-195.

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education (TOCE)*, 10(4), 16.

Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., ... & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60-67.

Sentance, S., & Csizmadia, A. (2017). *Teaching Computing in Secondary Schools: A Reader*. Routledge.

Weintrop, D., & Wilensky, U. (2015). To block or not to block, that is the question: students' perceptions of blocks-based programming. In Proceedings of the 14th international conference on Interaction design and children (pp. 131-140).

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, 49(3), 33-35.

## ANEXO A – MATRIZ CURRICULAR SEDUC

COMPONENTES CURRICULARES/ANO	1º ANO		2º ANO		3º ANO		TOTAL
	1º SEM	2º SEM	1º SEM	2º SEM	1º SEM	2º SEM	
	S	T	S	T	S	T	
DISCIPLINAS							



GOVERNO DO  
ESTADO DO CEARÁ  
Secretaria da Educação

ESCOLA ESTADUAL DE EDUCAÇÃO PROFISSIONAL - EEEP  
EIXO TECNOLÓGICO: **INFORMAÇÃO E COMUNICAÇÃO**  
CURSO TÉCNICO DE NÍVEL MÉDIO EM INFORMÁTICA  
TURMAS INICIADAS EM 2022

FORMAÇÃO PROFISSIONAL	Informática Básica	5	100																				100	
	Lógica de Programação			5	100																			100
	Arquitetura e Manutenção de Computadores			4	80																			80
	HTML / CSS			3	60																			60
	Planejamento de Carreira			2	40																			40
	Sistemas Operacionais					3	60																	60
	Programação Orientada a Objetos					4	80																	80
	Programação Web					4	80																	80
	Noções de Robótica					2	40																	40
	Gestão de Startups I					2	40																	40
	Redes de Computadores							3	60															60
	Design Gráfico							2	40															40
	Gerenciador de Conteúdo							3	60															60
	Banco de Dados							4	80															80
	Gestão de Startups II							2	40															40
	Laboratório Software									3	60													60
	Laboratório Hardware									3	60													60
	Projeto Integrador									6	120													120
	Gestão de Startups III									2	40													40
	Estágio Curricular																		15	300				300
	<b>SUBTOTAL</b>			<b>5</b>	<b>100</b>	<b>14</b>	<b>280</b>	<b>15</b>	<b>300</b>	<b>14</b>	<b>280</b>	<b>14</b>	<b>280</b>	<b>15</b>	<b>300</b>									<b>1.540</b>

Fonte: SEDUC, 2024