

INSTITUTO FEDERAL DO ESPÍRITO SANTO

PÓS-GRADUAÇÃO LATO SENSU EM PRÁTICAS PEDAGÓGICAS

**EWERTON LYRIO NASCIMENTO**

**UMA PROPOSTA PEDAGÓGICA PARA O DESENVOLVIMENTO DO  
PENSAMENTO COMPUTACIONAL NAS SÉRIES INICIAIS: UTILIZAÇÃO  
CONJUNTA DE TÉCNICAS DESPLUGADAS E PLUGADAS**

SANTA TERESA

2023

EWERTON LYRIO NASCIMENTO

**UMA PROPOSTA PEDAGÓGICA PARA O DESENVOLVIMENTO DO  
PENSAMENTO COMPUTACIONAL NAS SÉRIES INICIAIS: UTILIZAÇÃO  
CONJUNTA DE TÉCNICAS DESPLUGADAS E PLUGADAS**

Monografia apresentada à Coordenadoria do Curso de Pós-Graduação Lato Sensu em Práticas Pedagógicas, do Instituto Federal do Espírito Santo, Campus Santa Teresa, como requisito parcial para a obtenção do título de Especialista em Práticas Pedagógicas.

Orientador: Me. Bruno Gutierrez Ratto Clemente.

Santa Teresa

2023

(Biblioteca Major Bley do Instituto Federal do Espírito Santo)

N244u Nascimento, Ewerton Lyrio.

Uma proposta pedagógica para o desenvolvimento do pensamento computacional nas séries iniciais: utilização conjunta de técnicas desplugadas e plugadas / Ewerton Lyrio Nascimento. – 2023.

52 f. il; 30 cm.

Orientador: Prof. Bruno Gutierrez Ratto Clemente

Monografia (Especialização) – Instituto Federal do Espírito Santo. Programa de Pós-Graduação *Latu Sensu* em Práticas Pedagógicas. Santa Teresa, 2023.

Inclui bibliografias.

1. Pensamento computacional. 2. Educação. 3. Práticas pedagógicas. I. Clemente, Bruno Gutierrez Ratto. II. Instituto Federal do Espírito Santo. III. Título.

CDD 23 – 371.33

Bibliotecário: Domingos Sávio Côgo - CRB6-ES nº 430

**EWERTON LYRIO NASCIMENTO**

**UMA PROPOSTA PEDAGÓGICA PARA O DESENVOLVIMENTO DO  
PENSAMENTO COMPUTACIONAL NAS SÉRIES INICIAIS: UTILIZAÇÃO  
CONJUNTA DE TÉCNICAS DESPLUGADAS E PLUGADAS**

Monografia apresentada à Coordenadoria do Curso de Pós-Graduação Lato Sensu em Práticas Pedagógicas, do Instituto Federal do Espírito Santo, Campus Santa Teresa, como requisito parcial para a obtenção do título de Especialista em Práticas Pedagógicas.

Aprovado em 18 de Setembro de 2023

**COMISSÃO EXAMINADORA**

Prof. Dr. Bruno Gutierrez Ratto Clemente  
Instituto Federal do Espírito Santo  
Orientador

Prof. Dr. Charles Moreto  
Instituto Federal do Espírito Santo  
Membro interno

Prof. Dr. Milton Cesar Paes Santos  
Instituto Federal do Espírito Santo  
Membro externo

Documento assinado digitalmente



**BRUNO GUTIERREZ RATTO CLEMENTE**

Data: 14/03/2024 10:45:20-0300

Verifique em <https://validar.iti.gov.br>

Documento assinado digitalmente



**MILTON CESAR PAES SANTOS**

Data: 18/03/2024 09:30:04-0300

Verifique em <https://validar.iti.gov.br>

Documento assinado digitalmente



**CHARLES MORETO**

Data: 18/03/2024 11:01:01-0300

Verifique em <https://validar.iti.gov.br>

## RESUMO

A tecnologia se faz presente cada dia mais cedo na vida das novas gerações. Por conta disso, pode-se observar um crescimento no número de trabalhos focados em desenvolver o pensamento computacional mais cedo dentro do ensino regular formal, estudando seus benefícios e potencialidades. Entretanto, para que este conhecimento seja devidamente construído na mente dos jovens, é preciso que a forma de ensinar seja adaptada ao perfil dos estudantes, se fazendo valer de ferramentas intuitivas, de fácil compreensão, lúdicas e que promovam interação e protagonismo. Esse Trabalho Final de Curso tem a intenção de identificar a importância de se ensinar Pensamento Computacional nas séries iniciais, analisar os estudos sobre esta temática para então culminar na elaboração de uma proposta pedagógica que possa socializar o ensino deste conteúdo aos estudantes, independente da região, condições socioeconômicas e do acesso a ferramentas tecnológicas, podendo ser usado em qualquer escola mesmo sem o uso de computadores, mas que também pode ser expandido para escolas com acesso à laboratórios e máquinas para um aprofundamento ainda maior dos conceitos.

**Palavras-chave:** Pensamento Computacional. Método Desplugado. Método Plugado. Educação. Processo Pedagógico.

## ABSTRACT

Technology is part of the lives of new generations earlier every day. Because of this, we can observe a growth in the number of works focused on developing computational thinking earlier within regular formal education, studying its benefits and potential. However, for this knowledge to be properly developed in the minds of young people, it is necessary that the way of teaching is adapted to the students' profile, making use of intuitive, easy-to-understand, playful tools that promote interaction and protagonism. This Final Course Work intends to identify the importance of teaching Computational Thinking in the early grades, analyse the studies on this subject and then culminates in the elaboration of a pedagogical proposal that can socialize the teaching of this content to students, regardless of the region, socio-economic conditions and access to technological tools, which can be used in any school even without the use of computers, but which can also be expanded to schools with access to laboratories and machines for an even greater depth of concepts.

**Keywords:** Computational Thinking. Unplugged Method. Plugged Method. Education. Pedagogical Proposal.

## LISTA DE FIGURAS

Figura 1 – Número de publicações entre 2012 e 2015 nos eventos WIE, SBIE e WEI. .....	13
Figura 2 – Abordagem de estímulos ao Pensamento Computacional.....	16
Figura 3 – Abordagens recursos e ferramentas para promoção do Pensamento Computacional. ....	17
Figura 4 – Programação em bloco desplugado.....	23
Figura 5 – Tela do aplicativo Scratch. ....	23
Figura 6 – Site de acesso ao Code.org.....	25
Figura 7 – Exemplo de aplicação publicada no Code.org. ....	25
Figura 8 – Exemplo de desenvolvimento do Pensamento Computacional em Bloco Desplugado. ....	31
Figura 9 – Exemplo de desenvolvimento do Pensamento Computacional em Bloco Plugado. ....	33
Figura 10 – Exemplos de comandos usando a notação do Scratch.....	38
Figura 11 – Exemplos de comandos usando a notação do Scratch - Bloco SE e SE SENÃO.....	40
Figura 12 – Exemplos de comandos usando a notação do Scratch - Bloco SE e SE SENÃO.....	42
Figura 13 – Exemplos de comandos usando a notação do Scratch. Bloco de repetições.....	44
Figura 14 – Exemplos de comandos usando a notação do Scratch. Bloco acumuladores e contadores. ....	46
Figura 15 – Exemplos de comandos usando a notação do Scratch. Bloco acumuladores e contadores. ....	46

## **LISTA DE TABELAS**

Tabela 1 – Tabela programática e carga horária do projeto.....	35
--	----



## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO</b> .....	9
1.1.	PESQUISADOR E SEU CONTEXTO.....	9
1.2.	OBJETIVO GERAL.....	12
1.3.	OBJETIVOS ESPECÍFICOS .....	12
1.4.	PENSAMENTO COMPUTACIONAL .....	12
<b>1.4.1.</b>	<b>A importância do Pensamento Computacional na Pedagogia/Ensino</b> .....	14
<b>2.</b>	<b>ABORDAGENS DE ENSINO DO PENSAMENTO COMPUTACIONAL..</b>	16
2.1.	ABORDAGENS DE ENSINO.....	16
<b>2.1.1.</b>	<b>Abordagem de resolução de problemas</b> .....	17
<b>2.1.2.</b>	<b>Abordagem baseada em jogos e atividades lúdicas</b> .....	18
<b>2.1.3.</b>	<b>Abordagem baseada em projetos</b> .....	19
<b>2.1.4.</b>	<b>Abordagem de Programação</b> .....	20
<b>2.1.5.</b>	<b>Abordagem de Modelagem</b> .....	20
<b>2.1.6.</b>	<b>Ensino Integrado</b> .....	21
<b>2.1.7.</b>	<b>Ferramenta Scratch / Programação em Blocos</b> .....	22
<b>2.1.8.</b>	<b>Code.org</b> .....	24
<b>2.1.9.</b>	<b>Legó Mindstorms</b> .....	26
<b>2.1.10.</b>	<b>Pensamento Computacional Desplugado</b> .....	27
<b>2.1.11.</b>	<b>Jogos Educacionais</b> .....	28
<b>3.</b>	<b>PROPOSTA PEDAGÓGICA</b> .....	29
3.1.	OBJETIVO PEDAGÓGICO DA PROPOSTA.....	29
3.2.	DEFINIÇÃO DE METODOLOGIAS, FERRAMENTAS E TECNOLOGIAS .....	30
<b>3.2.1.</b>	<b>Computação desplugada</b> .....	30
<b>3.2.2.</b>	<b>Programação em blocos</b> .....	32

<b>3.2.3.</b>	<b>Combinação das metodologias.....</b>	<b>33</b>
<b>3.3.</b>	<b>FORMALIZAÇÃO DA PROPOSTA .....</b>	<b>34</b>
<b>3.3.1.</b>	<b>Tarefa 1.....</b>	<b>35</b>
<b>3.3.2.</b>	<b>Tarefa 2.....</b>	<b>37</b>
<b>3.3.3.</b>	<b>Tarefa 3.....</b>	<b>39</b>
<b>3.3.4.</b>	<b>Tarefa 4.....</b>	<b>41</b>
<b>3.3.5.</b>	<b>Tarefa 5.....</b>	<b>43</b>
<b>3.3.6.</b>	<b>Tarefa 6.....</b>	<b>45</b>
<b>3.3.7.</b>	<b>Tarefa 7.....</b>	<b>47</b>
<b>4.</b>	<b>CONCLUSÃO .....</b>	<b>49</b>
<b>4.1.</b>	<b>TRABALHOS FUTUROS.....</b>	<b>50</b>
	<b>REFERÊNCIAS.....</b>	<b>51</b>

## 1. INTRODUÇÃO

### 1.1. PESQUISADOR E SEU CONTEXTO

Nos caminhos sinuosos da vida, encontramos desafios que nos moldam, desvendando oportunidades que nos levam a uma jornada de autodescoberta e crescimento. Minha trajetória, marcada pela paixão pela tecnologia e pelo desejo constante de aprender, é um reflexo dessa jornada.

Meu nome é Ewerton Lyrio Nascimento e minha jornada educacional abrange uma série de realizações acadêmicas que me levaram a uma compreensão mais profunda do mundo da tecnologia da informação. Começando com minha formação em Ciências da Computação pela Ucam-Campos e em conjunto com o Tecnólogo em Desenvolvimento de Software pelo IFF-Campos, adentrei o vasto campo da computação, explorando suas nuances e desafios à medida que prossegue na minha busca pelo conhecimento.

Observei um alto grau na complexidade do ensinamento do Pensamento Computacional. Minha formação técnica em Automação Industrial pelo IFF-Campos ampliou ainda mais essa observação, visto que os colegas de sala sempre mostravam dificuldades em aprender o Pensamento Computacional. No entanto, foi na Licenciatura em Informática que encontrei a possibilidade de vencer esse desafio de transmitir o conhecimento intrigante do Pensamento Computacional.

O presente trabalho de conclusão de curso mergulha em minha jornada educacional e profissional, explorando os desafios que enfrentei ao aprender e ensinar o Pensamento Computacional na minha breve experiência em sala de aula. É um estudo para criação de uma metodologia que destaca a importância desse conceito na sociedade atual e as dificuldades que muitos enfrentam ao tentar compreendê-lo e aplicá-lo em suas carreiras e vidas cotidianas.

Ao pensar em educação, a ênfase recai no desenvolvimento do estudante e no perfil previsto para eles com base no Plano Pedagógico do curso em que estão matriculados. Sendo assim, sempre almeja uma formação completa e, em 2023, não temos como desconsiderar a importância da tecnologia na sociedade. Portanto, é

fundamental não apenas oportunizar o contato com as tecnologias desde cedo, mas também observar como a interação com a tecnologia pode potencializar o processo de ensino-aprendizagem.

No início do uso dos computadores nas escolas, esses equipamentos eram vistos como uma ferramenta facilitadora e que auxiliava no aumento de desempenho do aprendizado, criando novas formas de apresentar o conteúdo educacional, dinamizando e ampliando a desenvoltura dos estudantes e a disseminação do conhecimento. Com o surgimento de novas tecnologias, pode-se observar um crescente investimento acadêmico e científico em ampliar o conhecimento da tecnologia aplicada à educação e ao papel do computador na mesma, que passou de uma ferramenta de construção de apresentações para um coletivo de soluções programáveis para potencializar o processo de ensino-aprendizagem de inúmeras formas.

Tendo em vista o mencionado, houve o entendimento, formalizado na Base Nacional Comum Curricular, de que os estudantes devem conseguir utilizar as tecnologias digitais de forma completa e, assim, o Pensamento Computacional deve ser abordado na área da matemática junto com o ensinamento básico matemático, conforme detalhado no capítulo sobre “As tecnologias digitais e a computação” da BNCC, dentre outras afirmações (BRASIL, 2018). Surge então o conceito de Pensamento Computacional aplicado na educação.

Mas o que vem a ser Pensamento Computacional? Ainda existe muita discussão e inúmeras formas de definição, mas conforme definido na BNCC, o Pensamento Computacional é uma junção de processos e métodos que podem ajudar a desenvolver um algoritmo que resolva problemas de um amplo espectro de assuntos de forma padronizada, podendo essa solução ser realizada de forma manual ou com o auxílio de computador, conforme citação que segue:

Pensamento Computacional envolve as capacidades de compreender, analisar, definir, modelar, resolver, comparar e automatizar problemas e suas soluções, de forma metódica e sistemática, por meio de desenvolvimento de algoritmos (BRASIL, 2018).

Pode soar mais natural e intuitivo que o Pensamento Computacional seja desenvolvido única e exclusivamente por intermédio dos computadores ou rede de computadores, mas o Pensamento Computacional envolve uma etapa anterior ao ato de se interagir com o computador: envolve a compreensão do problema, a construção de um algoritmo que possa resolvê-lo, para só então culminar na implementação de uma possível solução.

Todavia, o dinamismo do mundo vem exigindo uma grande capacidade do indivíduo e suas habilidades devem ser desenvolvidas em um curto prazo, além da necessidade de ampliação no contexto social, econômico e cultural. Dessa forma, a tecnologia deve ser aplicada não somente de forma consumista, mas de forma construtiva, em que o estudante possa ser capaz de criar e propagar conhecimento tecnológico.

A programação é uma parte muito importante das ciências computacionais, que exige uma grande dedicação, e é através da programação que estimulamos o aprendizado do Pensamento Computacional.

Para aprender a programar é preciso que haja uma real compreensão do problema, para que seja possível elaborar uma forma lógica de resolvê-lo. Só então esta solução deve ser implementada por meio de uma linguagem de programação. Essa é a visão para o ensino da programação nos cursos dedicados ao desenvolvimento e análise de programa de nível técnico e ou superior.

A partir do estudo comparativo entre Zanetti *et al.* (2016), Araújo, Andrade e Guerrero (2016) e Grebogy (2021), observa-se um aumento significativo de estudos e publicações acadêmicas abordando a utilização de ferramentas de programação aplicadas já na educação básica e iniciando o desenvolvimento do Pensamento Computacional desde as séries iniciais. Isso se dá com o intuito de desenvolver e encorajar esta competência já nas primeiras séries, melhorando assim o raciocínio lógico dos estudantes e possibilitando que outras habilidades possam ser desenvolvidas e aprimoradas por este estudo, se fazendo valer de aplicações mais lúdicas e gráficas, como a robótica, que acabam despertando muito o interesse por parte dos estudantes.

## 1.2. OBJETIVO GERAL

O objetivo do presente trabalho é desenvolver uma proposta de prática pedagógica para o ensino e aplicação do Pensamento Computacional no início da educação das séries iniciais, incluindo ferramentas, procedimentos, métodos e metodologias que possam auxiliar sua aplicação e conseqüentemente no processo do desenvolvimento intelectual dos estudantes.

## 1.3. OBJETIVOS ESPECÍFICOS

- a) Identificar as principais abordagens pedagógicas utilizadas no ensino do Pensamento Computacional para as séries iniciais;
- b) Investigar as ferramentas e tecnologias mais utilizadas para o ensino do Pensamento Computacional em escolas ou ambientes de aprendizagem formal e informal;
- c) Verificar os benefícios do ensino do Pensamento Computacional para o desenvolvimento cognitivo e socioemocional de crianças e jovens, tais como raciocínio lógico, solução de problemas, criatividade, trabalho em equipe e autoconfiança;
- d) Elaborar uma proposta de prática pedagógica que possa ser aplicada para melhorar o processo de ensino e aprendizagem do Pensamento Computacional para crianças e jovens, como a formação de educadores, a criação de materiais didáticos adequados, a utilização de jogos e atividades lúdicas.

## 1.4. PENSAMENTO COMPUTACIONAL

O Pensamento Computacional é uma forma de pensamento que se desenvolve a partir dos conceitos e práticas da ciência da computação, porém é algo presente em todo tipo de atividades que são desempenhadas de forma cotidiana no mundo moderno. Segundo Alvarenga (2008), é uma habilidade que envolve a resolução de problemas, o raciocínio lógico, a criatividade e a colaboração, entre outras competências. O objetivo do Pensamento Computacional é desenvolver a capacidade

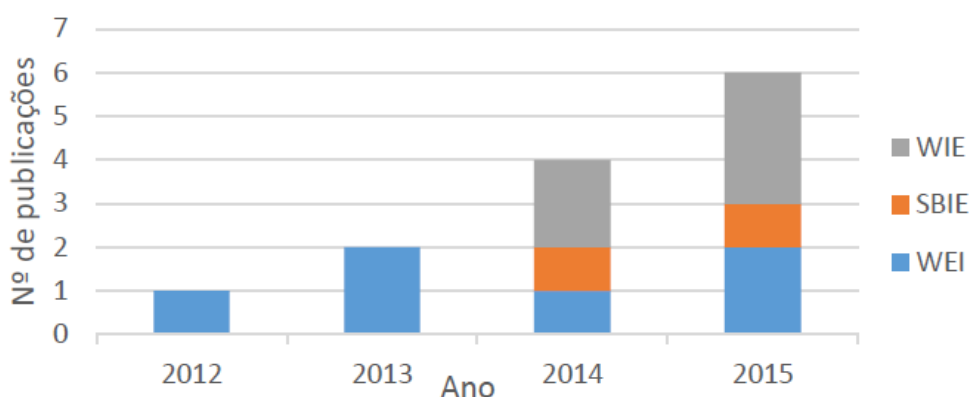
de resolver problemas de forma eficaz e sistemática, utilizando recursos computacionais e metodologias científicas.

Glizt (2017) afirma que os processos de aprendizagem do Pensamento Computacional são complexos, tanto para estudantes iniciais como para estudantes e profissionais experientes, principalmente com a abordagem tradicional. Zanetti e Oliveira (2015) afirmam que jovens geralmente não estão interessados em abordagens tradicionais, tornando-se motivados quando a robótica pedagógica ou outra atividade dinâmica é introduzida como uma maneira de conectar outras disciplinas ou áreas de interesse.

O Pensamento Computacional pode ser aprendido para cunho profissional, mas é possível usá-lo em qualquer área de atuação que for necessário. Segundo Rodrigues e Souza (2021) o termo foi conhecido a partir da publicação realizada por Jeannette M. Wing, em 2006, sendo que tanto humanos quanto máquinas possuem capacidade de solucionar problemas baseados no poder e nos limites da computação e existem inúmeras iniciativas para ensiná-lo.

Artigos publicados ou apresentados nos principais eventos e revistas de publicação nacional na área de informática e educação relatam um alto índice de trabalhos relacionados ao Pensamento Computacional, e não é diferente no âmbito internacional. Estudos de Zanetti *et al.* (2016), catalogaram e avaliaram entre 2012 e 2015 várias publicações; a Figura 1 mostra, ainda, um crescimento com o passar dos anos.

Figura 1 – Número de publicações entre 2012 e 2015 nos eventos WIE, SBIE e WEI.



Fonte: Zanetti et al. (2016).

O Pensamento Computacional é uma habilidade cada vez mais valorizada na sociedade atual, uma vez que a tecnologia está presente em praticamente todos os aspectos da nossa vida, desde a comunicação até a educação, passando pelo trabalho, lazer e saúde, afirma Vilaça e Araujo (2016). O Pensamento Computacional é, portanto, uma habilidade fundamental para o sucesso pessoal e profissional no mundo contemporâneo.

#### **1.4.1. A importância do Pensamento Computacional na Pedagogia/Ensino**

Uma das aplicações mais importantes do Pensamento Computacional é a programação, que é a habilidade de escrever códigos de computador para resolver problemas e realizar tarefas específicas. A programação permite que as pessoas criem aplicativos, jogos, sites e outras ferramentas que tornam a vida mais fácil e eficiente. A programação também é uma habilidade valiosa para o mercado de trabalho, já que muitas empresas procuram profissionais que possam desenvolver soluções personalizadas para suas necessidades específicas.

O Pensamento Computacional teve um grande destaque nos últimos anos, ao ponto de ganhar muitos investimentos e apoio de grandes nomes. Oliveira (2016) relata que o ex-presidente dos Estados Unidos, Barack Obama, elaborou diversos programas, dentre eles a iniciativa “Computer Science for All”, que pretendia efetuar investimentos na faixa de bilhões de dólares para a formação de professores, aquisição de material, dentre outras áreas. Oliveira (2016) ainda aponta que o ex-presidente afirma que: “Na nova economia, a programação não é opcional, é uma habilidade básica da mesma maneira que a escrita, leitura e aritmética”.

Na atualidade, vemos que as crianças já são consumidoras de tecnologia, porém um grande desafio é direcionar esse consumo para algo produtivo e a longo prazo, para que elas se tornem criativas e ativas nas decisões do mundo atual e com um futuro.

Conforme BRASIL (2018), a Base Nacional Comum Curricular (BNCC) já cita o pensamento computacional como foco especial no ensino da Matemática e acentua a cultura digital com uma das suas competências, além de afirmar que a cultura digital tem promovido mudanças sociais significativas. Os jovens têm se engajado cada vez



mais como protagonistas da cultura digital, muito pelo crescente e alto índice do uso da tecnologia por eles e favorecendo o crescimento do eu crítico de cada.

Nessa perspectiva o ensino do Pensamento Computacional e a importância do eu crítico, Pereira (2022) faz uma relação do ensino computacional na Educação Básica referenciando a Pedagogia Crítica de Freire, na qual a premissa de uma formação crítica deve conduzir ao desenvolvimento dos cidadãos, sendo eles capazes de analisar suas realidades sociais, históricas e culturais possibilitando transformá-las.

Com a sociedade atual sendo altamente tecnológica e cada vez mais aplicada ao nosso cotidiano, a tecnologia, junto com o Pensamento Computacional, torna-se primordial e melhora as habilidades analíticas dos estudantes.

## 2. ABORDAGENS DE ENSINO DO PENSAMENTO COMPUTACIONAL

Com base nos objetivos deste trabalho foi realizada uma pesquisa bibliográfica-documental em acervos digitais, buscando por palavras-chave relacionadas ao tema deste trabalho e enfatizando trabalhos que unissem abordagens do ponto de vista pedagógico e também tecnológico.

Os principais trabalhos aqui descritos também foram selecionados partindo da referência de ano, considerando trabalhos pré e pós pandemia de COVID-19 no Brasil, para que fosse possível analisar a produção científica sobre esses temas antes e depois deste evento único.

### 2.1. ABORDAGENS DE ENSINO

Araújo, Andrade e Guerrero (2016) realizaram um levantamento e classificação de publicações avaliadas em seu estudo com base nas abordagens de estímulo ao Pensamento Computacional, conforme ilustra o quadro a seguir:

Figura 2 – Abordagem de estímulos ao Pensamento Computacional.

Abordagem	Detalhamento	#	Referências
Programação	Scratch	7	[França e Amaral, 2013; Viel <i>et al.</i> , 2014; Rodriguez <i>et al.</i> , 2015; Ramos <i>et al.</i> , 2015; Schoeffel <i>et al.</i> , 2015; Araujo <i>et al.</i> , 2015; Rabello <i>et al.</i> , 2015]
	Code.org; Robomind	1	[Schoeffel <i>et al.</i> , 2015]
	Desempenho em programação	1	[Rodrigues <i>et al.</i> , 2015]
	Stencyl	1	[França e Tedesco, 2015b]
Robótica/Programação	S4A (Scratch for Arduino)	1	[Zanetti e Oliveira, 2015]
Atividades Desplugadas	Propostas de novas atividades e aplicações	8	[Scaico <i>et al.</i> , 2012; Santos <i>et al.</i> , 2015; Andrade <i>et al.</i> , 2013; Campos <i>et al.</i> , 2014; Hinterholz <i>et al.</i> , 2015; Araujo <i>et al.</i> , 2015; Ferreira <i>et al.</i> , 2015; Paiva <i>et al.</i> , 2015]
Ferramenta / Jogos	Tinta digital/ Avaliação de aplicativos	3	[Pereira Junior <i>et al.</i> , 2014; Gomes e Alencar, 2015; Gomes <i>et al.</i> , 2015]
Modelo	Associado à programação	1	[França e Tedesco, 2014]
Prova	Avaliação de questões PISA	1	[Mestre <i>et al.</i> , 2015]

Fonte: Araújo; Andrade; Guerrero (2016).

Grebody (2021) apresenta dados mais atualizados, tendo um retorno de 2.355 trabalhos levantados, com uma busca simples pelo termo “Pensamento Computacional” ou “*computational thinking*”, nas bases de dados científicas e repositórios digitais on-line, como ACM Digital Library, EI Compendex, Scielo, Portal de periódicos da Capes, BDTD - Biblioteca Digital Brasileira de Teses e Dissertações, anais de eventos do Congresso da Sociedade Brasileira de Computação e outras mais.

Figura 3 – Abordagens recursos e ferramentas para promoção do Pensamento Computacional.

Estudos por Abordagem	Estudos por Categoria de Recursos		Estudos por Ferramenta				
Recursos Digitais	68	Jogos Digitais (JD), 17	Code.Org	5	Blockly Games	1	
			Lightbot	5	Bloxor	1	
			Criação de JD	4	Classcraft	1	
			Codecombat	2	Kahoot	1	
			Matching Math	1	Mesa multitoque	1	
			NUI	1	The Foos	1	
		Linguagem de programação (LP)	4	Phyton	1	Visualg	1
				Portugol Studio	1	Logo	1
		Linguagem de programação Visual (LPV)	36	Scratch	26	Furbot	2
				App Inventor	3	Dr. Scratch	1
				Não especificado	3	Stencyl	1
				Processing2	2	Etoys	1
				Unity	2	Poredu	1
		Robótica pedagógica (RP)	12	Scratch for Arduino (S4A)	8	Zerobot	2
				Kit Lego Mindstorms	3	Blockly	1
Computação Desplugada		38	Utilização de jogos	10	Atividades objetos concretos	6	
			Atividades adaptadas ou criadas	9	Algoritmos situações cotidianas	5	
			Atividades do CS Unplugged	7	Histórias em quadrinhos	2	
Abordagem Mista (Recursos Digitais e Computação Desplugada)		26	CD + Jogos digitais	9	CD + Lego	1	
			CD + Scratch	7	CD + S4A	1	
			CD + Code.Org	6	CD + Construct	1	
			CD + Logo			1	

Fonte: Grebody; Santos; Castilho (2021).

Esses estudos nos possibilitaram identificar as principais abordagens avaliadas nos estudos nacionais. A escolha da abordagem ou ferramenta de ensino é de grande importância. Nas subseções que se seguem há um levantamento sobre cada uma delas, ferramentas usadas, vantagens e desvantagens, para que seja possível compreendê-las com maior propriedade.

### 2.1.1. Abordagem de resolução de problemas

Essa abordagem enfatiza a resolução de problemas como a principal forma de ensinar o Pensamento Computacional. Os estudantes são apresentados a problemas reais ou

fictícios que exigem Pensamento Computacional para serem resolvidos e, no processo de construção do aprendizado, aprendem a decompor o problema, identificar padrões e relações, desenvolver algoritmos e testar e avaliar a solução encontrada.

- Vantagens:
  - Ajuda a desenvolver habilidades de pensamento crítico, raciocínio lógico e resolução de problemas;
  - Ajuda a entender os processos envolvidos na resolução de problemas complexos;
  - Ajuda a desenvolver habilidade de abstração e generalização;
  - Ajuda a desenvolver habilidades de comunicação e colaboração;
  - Pode ser aplicada a diferentes áreas do conhecimento e problemas do mundo real.
  
- Desvantagens:
  - Pode ser difícil para os iniciantes aplicarem as técnicas de resolução de problemas as situações desconhecidas;
  - Pode ser necessário um conhecimento prévio em matemática e lógica para entender e aplicar as técnicas de resolução de problemas;
  - Pode ser difícil avaliar a compreensão dos estudantes em relação às técnicas de resolução de problemas;
  - Pode requerer tempo significativo para a aprendizagem das técnicas de resolução de problemas.

### **2.1.2. Abordagem baseada em jogos e atividades lúdicas**

Essa abordagem utiliza jogos e atividades lúdicas para ensinar o Pensamento Computacional. Os jogos são projetados para incentivar a resolução de problemas e o pensamento criativo e podem ser usados para ensinar conceitos como algoritmos, lógicas, *loops* e condições.

- Vantagens:
  - Ajuda a tornar o processo de aprendizagem mais divertido e envolvente para as crianças;

- Ajuda a desenvolver habilidades motoras finas e coordenação olho-mão;
- Ajuda a desenvolver habilidades sociais e emocionais, como colaboração, comunicação e resolução de conflitos.
- Desvantagens:
  - Pode não ser suficiente para o ensino de conceitos mais avançados de Pensamento Computacional;
  - Pode não ser tão eficaz para estudantes mais velhos que podem achar as atividades infantis;
  - Pode requerer investimento significativo em materiais e recursos.

### **2.1.3. Abordagem baseada em projetos**

Essa abordagem enfatiza a criação de projetos como uma forma de ensinar Pensamento Computacional. Os estudantes trabalham em projetos que envolvem a resolução de problemas usando Pensamento Computacional. Eles aprendem a planejar, implementar e testar soluções, bem como a trabalhar em equipe e se comunicar efetivamente.

- Vantagens:
  - Ajuda a desenvolver habilidades práticas e aplicáveis do Pensamento Computacional;
  - Ajuda a desenvolver habilidades de trabalho em equipe e colaboração;
  - Ajuda a tornar o aprendizado mais interessante e envolvente.
- Desvantagens:
  - Requer tempo e recursos significativos para o desenvolvimento do projeto;
  - Pode ser difícil de avaliar e avaliar o desempenho dos estudantes em projetos diferentes;
  - Pode não fornecer uma cobertura abrangente de todos os conceitos do Pensamento Computacional.

#### 2.1.4. Abordagem de Programação

Essa abordagem ensina Pensamento Computacional através da programação de computadores. Os estudantes aprendem a criar algoritmos e a escrever códigos para resolver problemas. Eles também aprendem a depurar e testar seus programas.

- Vantagens:
  - Ensina habilidades práticas e aplicáveis de programação;
  - Ajuda a desenvolver habilidades de resolução de problemas e pensamento algorítmico;
  - Pode ajudar os estudantes a compreender melhor a tecnologia e a sua utilização.
  
- Desvantagens:
  - Pode ser difícil de aprender, especialmente para aqueles sem experiência prévia em programação;
  - Requer recursos significativos, incluindo acesso a hardware e software;
  - Pode ser menos envolvente e interessante para os estudantes que não estão tão interessados na programação.

#### 2.1.5. Abordagem de Modelagem

Essa abordagem ensina Pensamento Computacional através da criação de modelos e simulações. Os estudantes aprendem a representar problemas e sistemas usando modelos, e a usar esses modelos para testar soluções e prever resultados.

- Vantagens:
  - Ensina habilidades de pensamento crítico e resolução de problemas;
  - Ajuda os estudantes a desenvolverem uma compreensão mais profunda dos problemas e das soluções;
  - Ajuda os estudantes a entenderem melhor as relações entre os componentes de um problema;
  - Ajuda a desenvolver habilidades de abstração e generalização;
  - Ajuda a desenvolver habilidades de comunicação e colaboração.

- Desvantagens:
  - Pode ser difícil para os estudantes compreenderem e aplicarem conceitos abstratos de modelagem;
  - Pode ser necessário um conhecimento prévio em matemática e lógica para compreender a modelagem;
  - Pode ser difícil avaliar a compreensão dos estudantes da modelagem;
  - Pode requerer tempo significativo para a aprendizagem da modelagem.

### 2.1.6. Ensino Integrado

Essa abordagem envolve a integração do Pensamento Computacional em outras disciplinas, como matemática, ciências e artes. Os estudantes aprendem a aplicar o Pensamento Computacional em contextos reais e a compreender como ele se relaciona com outras áreas do conhecimento.

- Vantagens:
  - Ajuda a contextualizar o Pensamento Computacional e sua relação com outras áreas do conhecimento;
  - Ajuda a tornar o aprendizado mais interessante e envolvente;
  - Ajuda a desenvolver habilidades de resolução de problemas em áreas específicas.
- Desvantagens:
  - Pode ser difícil para os professores integrarem efetivamente o Pensamento Computacional em outras áreas do conhecimento;
  - Pode não fornecer uma cobertura abrangente de todos os conceitos do Pensamento Computacional;
  - Pode ser menos eficaz para estudantes que não estão tão interessados nas outras áreas do conhecimento que estão sendo integradas.

Cada abordagem tem suas vantagens e desvantagens e é importante escolher a que melhor se adapta às necessidades e interesses dos estudantes. É possível também combinar diferentes abordagens para criar uma abordagem híbrida e mais abrangente. O importante é que a abordagem escolhida proporcione uma experiência

de aprendizado envolvente e desafiadora e que crie condições para que os estudantes desenvolvam seus conhecimentos sobre Pensamento Computacional.

Junto com a escolha da abordagem e suas vantagens, é necessário que se escolha a ferramenta adequada, podendo dessa forma ampliar as vantagens e melhorar o ensino do Pensamento Computacional.

### **2.1.7. Ferramenta Scratch / Programação em Blocos**

Plataforma de programação visual criada pelo Massachusetts Institute of Technology (MIT), que permite aos iniciantes de programação criar jogos, histórias interativas, animações e outros projetos usando blocos de código que podem ser arrastados e soltos.

É uma abordagem pedagógica para ensinar programação que usa blocos gráficos que representam o código/parte de uma programação e tem como objetivo ensinar de forma divertida e acessível aos iniciantes do Pensamento Computacional.

- Vantagens:
  - Permitir que os iniciantes em programação criem “programas” facilmente, sem a necessidade de escrever códigos;
  - Possui uma grande comunidade de usuários e recursos disponíveis online, o que permite aos estudantes compartilhar seus projetos e aprender com outros estudantes;
  - Permite a criação de projetos criativos e interativos, incluindo jogos, animações e histórias interativas;
  - Oferece suporte para vários idiomas e permitindo a criação em sua língua nativa;
  - Amplamente estudado pela comunidade acadêmica, o que propicia um cenário favorável para sua aplicação com boa fundamentação teórica;
  - Ferramenta gratuita;
  
- Desvantagens:



- Pode ser limitado em sua capacidade de ensinar conceitos mais avançados de programação, já que o desenvolvimento em blocos é mais limitado do que o desenvolvimento por código escrito;
- Alguns estudantes podem considerar a programação em blocos menos interessante ou desafiadora do que a programação por código escrito.

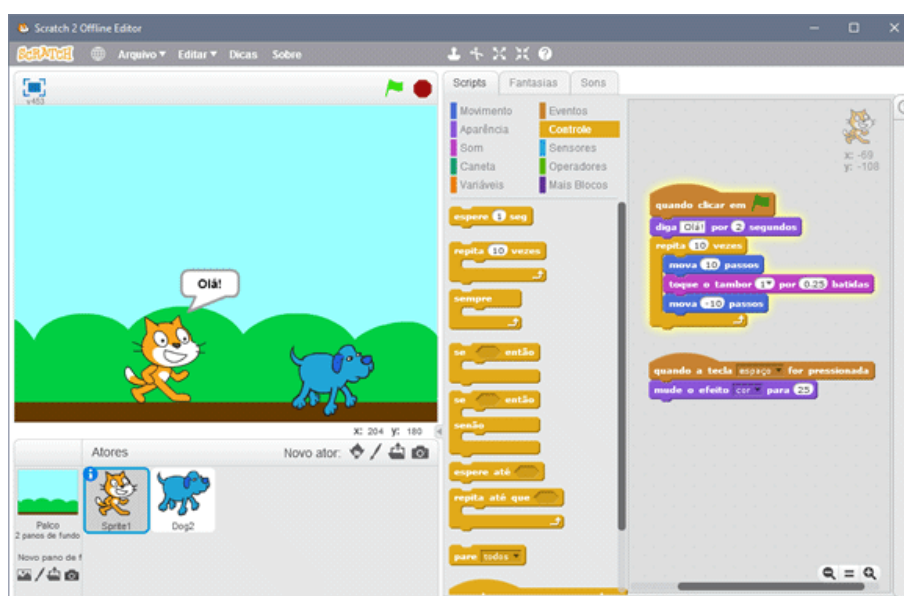
São ferramentas pedagógicas valiosas para o ensino do Pensamento Computacional e ou programação para os iniciantes, pois possuem interfaces amigáveis e acessíveis que é possível verificar nas Figuras a seguir:

Figura 4 – Programação em bloco desplugado.



Fonte: Rede Brasileira de Aprendizagem Criativa (2021).

Figura 5 – Tela do aplicativo Scratch.



Fonte: Motta (2023).

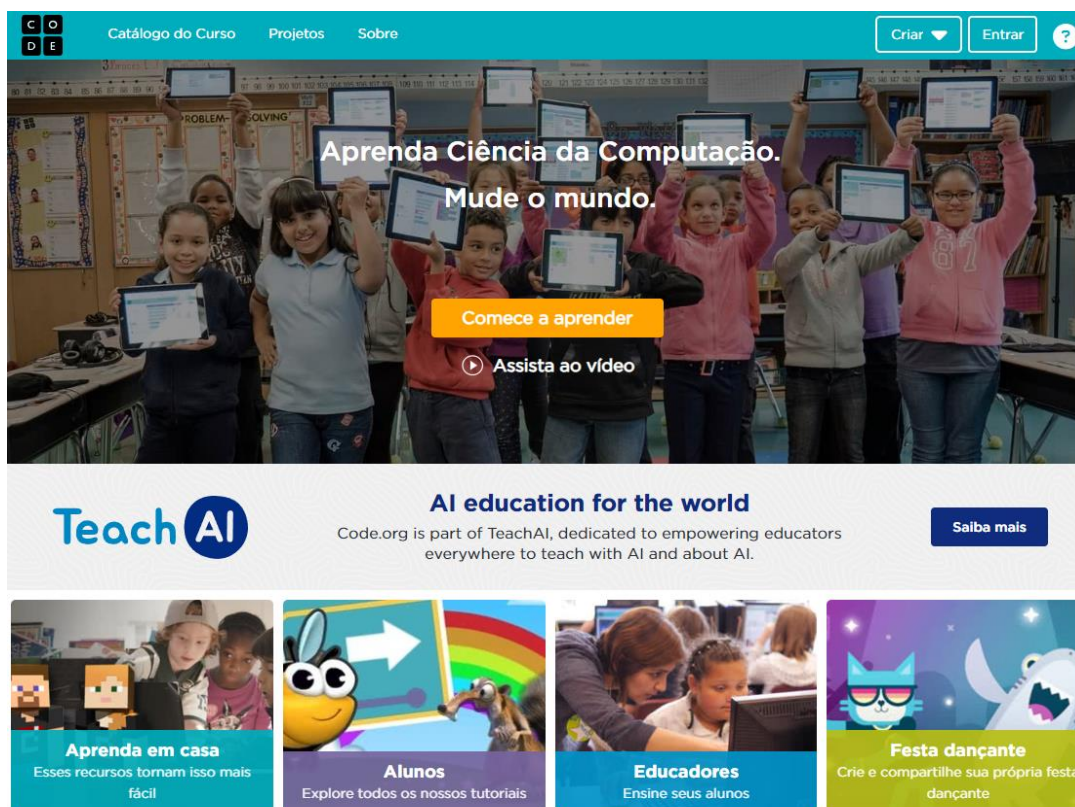
Esse tipo de interface proporciona aos iniciantes uma prática da programação e exploração do Pensamento Computacional sem grandes limitações de linguagem ou sintaxe.

### **2.1.8. Code.org**

Uma plataforma educacional que oferece cursos gratuitos de programação para estudantes de todas as idades.

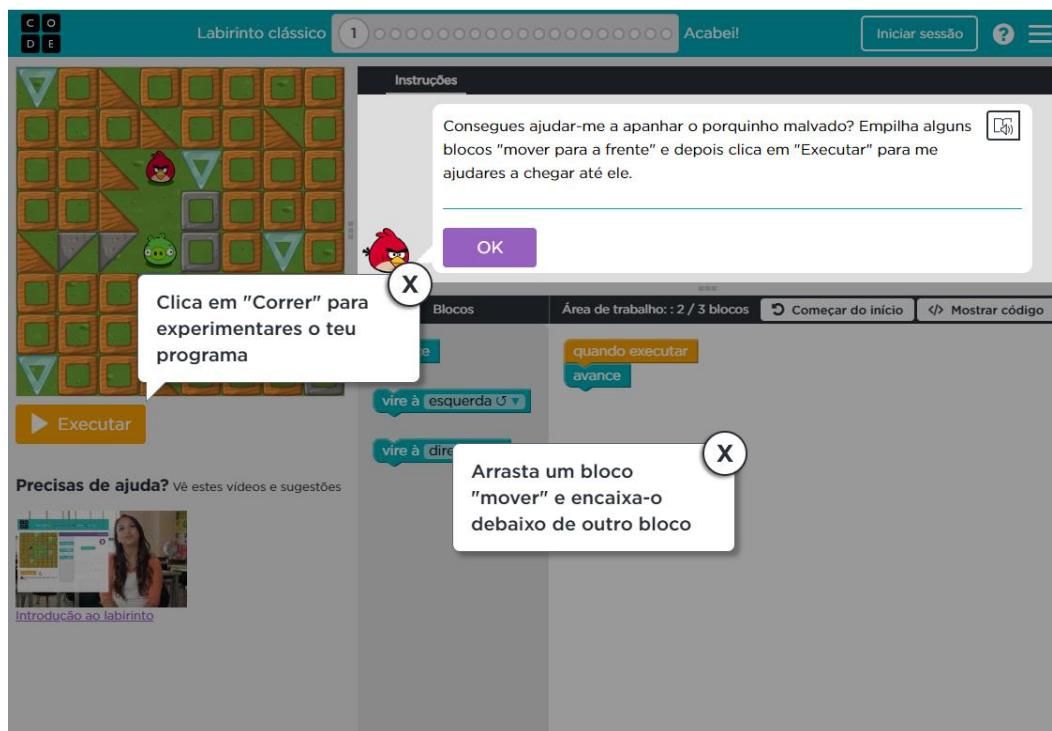
- **Vantagens:**
  - Oferece cursos de programação gratuitos e acessíveis para estudantes de todos os níveis, desde o jardim de infância até o ensino médio;
  - Os cursos são projetados para serem divertidos e interativos, o que ajuda a manter o interesse dos estudantes;
  - Os cursos são projetados para serem autoexplicativos, o que significa que os estudantes podem aprender programação sem a necessidade de um tutor;
  - A plataforma é suportada por muitas empresas de tecnologia de renome, o que garante um conteúdo atualizado e relevante para os estudantes;
  - Os cursos incluem projetos que incentivam os estudantes a aplicar o que aprenderam em projetos do mundo real.
  
- **Desvantagens:**
  - A plataforma pode ser limitada em comparação com outras ferramentas de programação disponíveis no mercado, sendo a falta de personalização seu maior ponto fraco, ou seja, as atividades são padronizadas e limitadas ao exemplo;
  - A plataforma pode não oferecer recursos avançados ou suporte para estudantes que já têm algum conhecimento de programação;
  - A plataforma é baseada em uma conexão com a internet, o que pode ser um problema para escolas e estudantes com conexões limitadas.

Figura 6 – Site de acesso ao Code.org



Fonte: Code (2024a).

Figura 7 – Exemplo de aplicação publicada no Code.org.



Fonte: Code (2024b).

Em resumo, o Code.org é uma plataforma educacional que oferece cursos de programação gratuitos e acessíveis para estudantes de todas as idades. É uma ótima opção para escolas e estudantes que buscam uma introdução à programação, pois oferece uma abordagem divertida e interativa para o ensino de programação.

### **2.1.9. Lego Mindstorms**

Plataforma de robótica educacional que permite aos estudantes construir e programar robôs usando blocos LEGO e um software de programação específico. Algumas vantagens e desvantagens do uso do LEGO Mindstorms para o ensino são:

- **Vantagens:**
  - Ajuda a desenvolver habilidades de Pensamento Computacional, resolução de problemas, pensamento crítico e trabalho em equipe;
  - Permite que os estudantes experimentem e visualizem o resultado de suas programações;
  - É uma forma lúdica e interativa de aprender programação e robótica;
  - Estimula a criatividade e a imaginação dos estudantes;
  - Pode ser integrado com outras disciplinas, como matemática, física e ciências.
  
- **Desvantagens:**
  - Pode ser caro para algumas escolas e instituições;
  - Pode ser necessário um treinamento prévio dos professores e educadores para utilizar o LEGO Mindstorms de forma eficaz;
  - Pode ser limitado em sua capacidade de ensinar conceitos mais avançados de programação e robótica;
  - Pode ser demorado para montar e programar um robô completo.

O LEGO Mindstorms, em geral, pode ser uma excelente ferramenta para o ensino de programação e robótica para estudantes de todas as idades. Ele é uma forma lúdica e interativa de aprender e pode ajudar os estudantes a desenvolver habilidades importantes para o seu futuro acadêmico e profissional.

### 2.1.10. Pensamento Computacional Desplugado

É o ensino de Pensamento Computacional que não envolve o uso de dispositivos eletrônicos. Em vez disso, é uma abordagem que utiliza atividades lúdicas e práticas que ajudam os estudantes a desenvolver habilidades de pensamento crítico, resolução de problemas, criatividade e colaboração.

- Vantagens:
  - Não requer o uso de dispositivos eletrônicos, tornando-o acessível a todas as escolas e estudantes;
  - Encoraja o trabalho colaborativo, o que ajuda a desenvolver habilidades sociais e de comunicação além do próprio Pensamento Computacional;
  - Frequentemente acompanhado de práticas pedagógicas mais envolventes e lúdicas, com manipulação de materiais físicos e dinâmicas de interação, o que ajuda a manter o interesse e o engajamento dos envolvidos;
  - As atividades podem ser adaptadas para diferentes níveis de habilidade e idade, tornando-as adequadas para estudantes de todas as idades;
  - As atividades ajudam a desenvolver habilidades de pensamento crítico, resolução de problemas e criatividade, que são importantes em muitas áreas além da tecnologia.
  
- Desvantagens:
  - As atividades podem exigir mais tempo e recursos para preparação e execução em comparação com a abordagem baseada em tecnologia;
  - Alguns estudantes podem preferir trabalhar com tecnologia e podem achar as atividades desplugadas menos interessantes;
  - A abordagem desplugada não é tão realista quanto o trabalho com tecnologia e pode não oferecer uma visão completa da programação;
  - Baixa escalabilidade, ficando difícil simular problemas e exercícios mais complexos e em maior escala.

Em resumo, o Pensamento Computacional desplugado é uma abordagem valiosa para o ensino de habilidades de pensamento crítico, resolução de problemas e

criatividade, especialmente para os estudantes que não têm acesso à tecnologia. É uma ótima opção para escolas que desejam fornecer uma introdução ao Pensamento Computacional de uma forma divertida e envolvente.

### **2.1.11. Jogos Educacionais**

Essa abordagem utiliza jogos que são projetados para incentivar a resolução de problemas e o pensamento crítico e podem ser usados para ensinar conceitos como algoritmo, lógica e etc.

- Vantagens:
  - Ajuda a tornar o processo de aprendizagem mais divertido e envolvente para os estudantes;
  - Ajuda a desenvolver habilidades motoras finas e coordenação motora de ação e reação.
  
- Desvantagens:
  - Pode não surtir o mesmo impacto em estudantes mais avançados, os quais podem considerar as atividades como sendo direcionadas para um público mais jovem;

### 3. PROPOSTA PEDAGÓGICA

À luz dos dados levantados no presente trabalho, das abordagens e das ferramentas estudadas e catalogadas, foi elaborada uma proposta pedagógica para o ensino do Pensamento Computacional para estudantes das séries iniciais.

A aplicação de metodologias ativas e de gamificação no aprendizado do pensamento computacional é de grande valia, pois colocar o estudante como protagonista é uma eficiente alternativa às metodologias mais antigas e tradicionais de ensino.

Outro fator importante é a necessidade de um conhecimento prévio por parte do professor que conduzirá a proposta pedagógica, havendo necessidade de conhecimento no uso das ferramentas e tecnologias que serão adotadas. Dito isso, cabe mencionar que as ferramentas propostas neste trabalho são simples e introdutórias aos conceitos de programação, não sendo necessário uma capacitação extensa para que seja aplicada de forma eficaz.

Nas subseções a seguir serão destacados o objetivo pedagógico da proposta, as metodologias e ferramentas utilizadas e, por fim, a própria proposta, em um formato que pode ser apreciado e aplicado por educadores da rede pública e privada de ensino para que possam se beneficiar dos estudos que levaram à referida proposta.

#### 3.1. OBJETIVO PEDAGÓGICO DA PROPOSTA

O principal objetivo da proposta pedagógica é desenvolver nos estudantes noções de Pensamento Computacional através de um conjunto de metodologias que o possibilitem transitar desde a interpretação de problemas até o desenvolvimento de um programa de computador.

Essa jornada inclui etapas valiosas, como o desenvolvimento de um programa inicialmente de forma desplugada e, em seguida, através de uma ferramenta de desenvolvimento. Durante todo o processo os estudos sobre as ferramentas e metodologias aqui descritas servem como referência.

Essa abordagem auxilia na aplicação da proposta, tornando mais acessível a união da prática didática com o estudo acadêmico. O resultado é a facilitação da disseminação desse conteúdo nas escolas.

### 3.2. DEFINIÇÃO DE METODOLOGIAS, FERRAMENTAS E TECNOLOGIAS

Para desenvolver o Pensamento Computacional será necessário aplicar diferentes estratégias no processo de ensino e aprendizagem.

Com base nas ferramentas levantadas, catalogadas e estudadas nos capítulos anteriores, optou-se inicialmente pelo uso de abordagem desplugada para a programação para o ensino do Pensamento Computacional em estudantes de todas as idades – em especial para o público-alvo a que se destina esta proposta –, aliada com programação em blocos, que é uma forma visual e interativa de programar, na qual os estudantes arrastam e soltam blocos de código para criar programas. A abordagem desplugada envolve o uso de jogos, brinquedos e atividades divertidas para incentivar o aprendizado. Faremos uma breve introdução das motivações e das vantagens de cada abordagem em separado e então trataremos as motivações e vantagens de utilizá-las de forma combinada.

#### 3.2.1. Computação desplugada

A abordagem desplugada é uma maneira divertida e engajante de aprender Pensamento Computacional, pois envolve atividades que incentivam a exploração e o raciocínio lógico. Ao utilizar jogos e brinquedos, os estudantes são incentivados a experimentar e testar diferentes estratégias para alcançar seus objetivos, o que ajuda a desenvolver habilidades de solução de problemas.

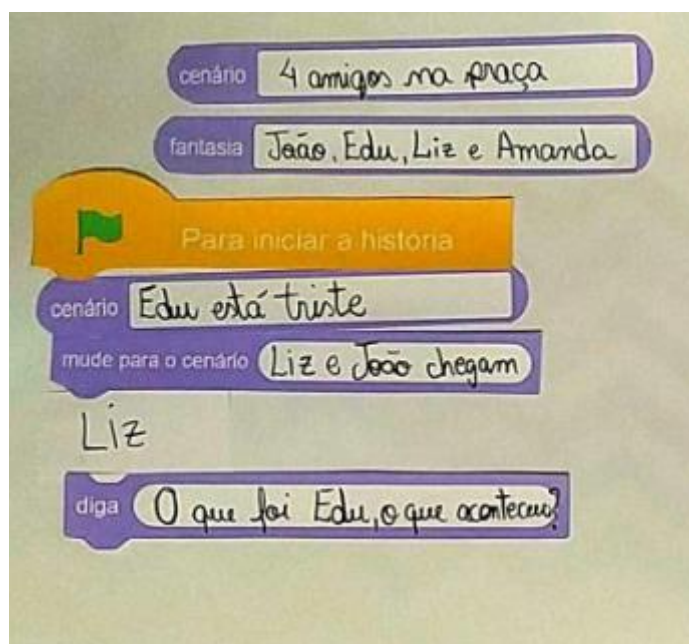
A abordagem desplugada envolve atividades offline, com materiais de fácil acesso e que permitem aos estudantes explorar conceitos fundamentais do Pensamento Computacional. Utilizando materiais como cartões, quebra-cabeças, jogos de tabuleiro ou com blocos de programação impressos, os estudantes podem manipular fisicamente os blocos e construir algoritmos e estudar os resultados. Essa abordagem proporciona uma base sólida para o desenvolvimento de habilidades do pensamento sequencial, da iteração e da tomada de decisões.



Aqui cabe mencionar também que a natureza física e material desta abordagem dá ao estudante algo concreto para se pautar ao estudar conceitos muitas vezes abstratos existentes na programação. A possibilidade de se tocar e manipular peças ou cartões, de se contar a quantidade de frutas acrescentadas em um comando de repetição ou de se analisar os resultados de uma condicional auxiliam na percepção material de conceitos abstratos, garantindo uma aprendizagem mais significativa para o estudante.

Outro ponto interessante de se trabalhar com a computação desplugada é que ela pode ser feita fazendo uso da estrutura de blocos que será trabalhada na próxima etapa, para que o estudante já tenha esse contato inicial com o Pensamento Computacional num formato que será aproveitado nas próximas etapas da metodologia e sua estrutura básica, como podemos visualizar na Figura 8, abaixo.

Figura 8 – Exemplo de desenvolvimento do Pensamento Computacional em Bloco Desplugado.



Fonte: Rede Brasileira de Aprendizagem Criativa (2021).

Observa-se então uma aplicação de uma programação desplugada que não apenas permite a manipulação física dos cartões, mas também já introduz o conceito de programação em blocos.

### 3.2.2. Programação em blocos

A programação em blocos é uma linguagem de programação visual e fácil de aprender, tornando-a uma excelente escolha para professores e escolas que desejam introduzir os estudantes à programação. Além disso, é uma ferramenta que pode ser usada tanto no modo desplugado quanto via computacional, assim funcionando como uma metodologia evolutiva.

A programação em blocos é uma excelente maneira de ensinar Pensamento Computacional para estudantes iniciantes, pois torna a programação mais acessível e fácil de entender, permitindo uma visualização de blocos de código de uma forma lógica e estruturada, contrapondo conceitos naturalmente abstratos do ensino tradicional de programação. Isso permite que os estudantes aprendam a pensar de forma sequencial e lógica, o que é uma habilidade fundamental do Pensamento Computacional.

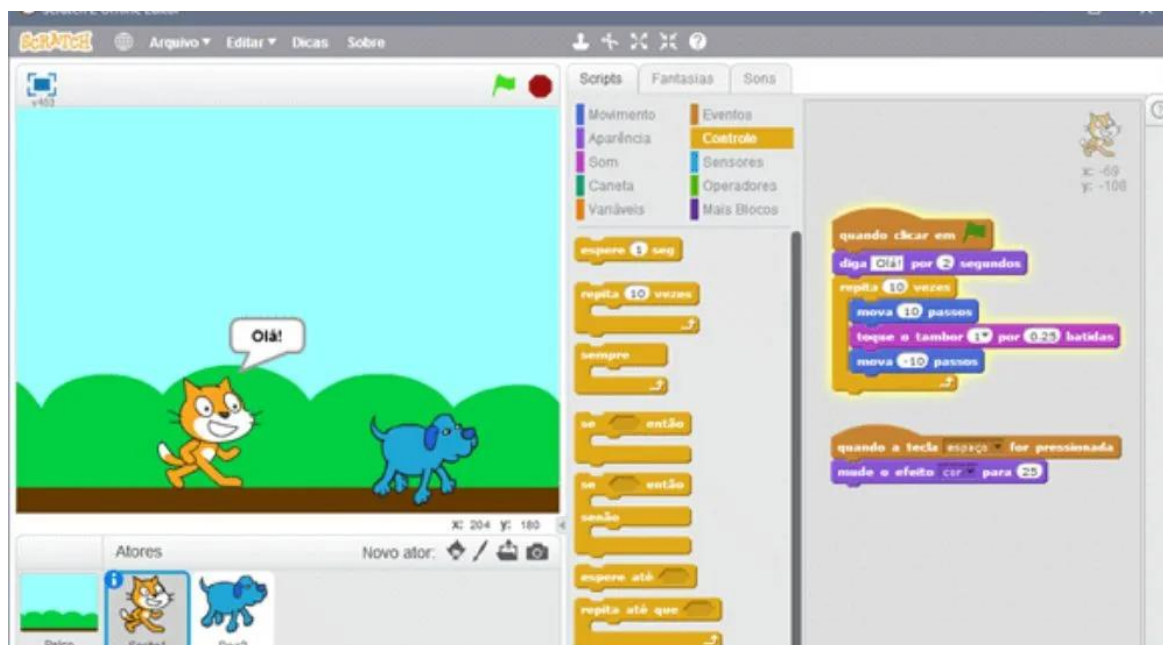
Outra vantagem desta abordagem é que a mesma permite que os estudantes vejam imediatamente os resultados de suas ações. Isso é importante porque torna a programação mais tangível e permite que os estudantes experimentem com diferentes soluções para os mesmos problemas. Ferramentas como o Scratch possibilitam a criação de personagens, cenários e cenas, simplificando o processo apenas à escolha do avatar (a aparência do personagem) e já permitindo que o mesmo veja o resultado em tela, o que é bem mais motivador do que resultados de uma função na forma de texto, pelo console da aplicação.

Além disso, é importante destacar que, embora a programação em blocos seja uma forma simplificada de programação, ela ainda fornece uma base sólida para a aprendizagem de outras linguagens de programação no futuro, além de poder ser aplicada de forma desplugada também, onde uma completaria o aprendizado da outra.

A abordagem plugada utiliza ferramentas de programação em blocos em um ambiente computadorizado, como Scratch ou Code.org. Essas plataformas oferecem uma interface amigável que permite aos estudantes criarem programas visualmente, arrastando e soltando blocos de código. Os estudantes podem explorar as funcionalidades dessas ferramentas, criando histórias interativas, jogos simples ou

animações. Essa abordagem estimula a criatividade e o pensamento lógico, proporcionando uma transição suave da metodologia desplugada para a plugada.

Figura 9 – Exemplo de desenvolvimento do Pensamento Computacional em Bloco Plugado.



Fonte: Main Leaf (2022).

### 3.2.3. Combinação das metodologias

Conforme resultados apresentados em vários dos trabalhos estudados e apresentados na seção 2, essa combinação de abordagens já se provou eficaz, engajante e pode ajudar a desenvolver habilidades importantes em áreas como solução de problemas, lógica de programação, trabalho em equipe e criatividade.

A abordagem desplugada e a programação em blocos também incentivam o trabalho em equipe e a colaboração, pois muitas atividades envolvem o trabalho em grupos para alcançar um objetivo comum. Isso ajuda a desenvolver habilidades sociais e emocionais, bem como habilidades técnicas de programação.

A escolha das formas de ensino do pensamento plugadas e desplugadas vêm inicialmente de forma conflitante, porém uma metodologia pode complementar a outra e servir para aplicação em vários ambientes e de forma distintas.

Assim como citado, o uso da metodologia desplugada pode iniciar o aprendizado da programação em blocos, usando de brinquedos antigos com o estilo de quebra-cabeça, que pode gerar um conforto no processo de aprendizado, focando em uma aplicabilidade teórica, tendo somente um diferencial com a desplugada, que é a sua execução automática e com visual em um computador.

### 3.3. FORMALIZAÇÃO DA PROPOSTA

Neste capítulo será apresentado uma proposta de ensino do Pensamento Computacional aplicado à prática de programação para estudantes das séries iniciais.

A proposta foi desenvolvida pensando na simplicidade e na atuação dos professores em qualquer escola, não importando a estrutura tecnológica. São empregadas várias ferramentas e metodologias ativas para o desenvolvimento adaptativo: com o método desplugado o professor poderá interagir com os estudantes realizando práticas durante a “compilação” dos modelos gerados; com o método plugado, tendo como ferramenta o Scratch, o professor poderá agilizar e demonstrar via computador a execução, simulando a vivência real de um programador.

Os exemplos apresentados podem ser adaptados ou trocados, afim de aproximar-se da realidade da localidade onde está sendo ensinado. Além disso, o professor tem a liberdade de efetuar práticas complementares. Também é um fator importante a necessidade do professor já possuir uma experiência tanto com o Pensamento Computacional quanto com as ferramentas/métodos que serão utilizadas. Esse requisito é importante para que os estudantes possam aproveitar na íntegra o aprendizado. A proposta foi modelada para ser implementada sem a necessidade de computadores, mas, havendo a possibilidade, o professor pode usá-los até como uma prova de conceito para o processo de aprendizado dos estudantes.

A seguir uma tabela com o resumo programático para o projeto. O número de encontros é uma expectativa baseada na complexidade dos conteúdos a serem abordados nas tarefas, cabendo adequação do professor para o tempo disponível com os estudantes no momento presencial.

Tabela 1 – Tabela programática e carga horária do projeto.

Conteúdo	Descrição		Número de Encontros
1 Pensamento Computacional	Definir o Pensamento Computacional e como aplicá-lo na resolução de problemas do dia a dia.	Momento presencial	2 Encontros
2 Converter o Pensamento Computacional em código de blocos	Conhecer os blocos iniciais e iniciar a conversão do Pensamento Computacional em código em blocos.	Momento presencial	2 Encontros
3 Introdução do uso do bloco de Decisão	Conhecer os blocos e formas de uso do bloco que auxilia na tomada de decisão.	Momento presencial	3 Encontros
4 Introdução ao uso dos blocos de Sequência	Conhecer a forma de aplicar o bloco de Decisão como escolha Sequenciada.	Momento presencial	3 Encontros
5 Introdução ao uso dos blocos de Repetição	Conhecer as formas de Repetição mais comuns utilizadas.	Momento presencial	4 Encontros
6 Introdução a acumuladores e contadores	Conhecer as formas de acumular valores e contadores e como podemos utilizar para o desenvolvimento do Pensamento Computacional.	Momento presencial	2 Encontros
7 Implementação modelo Plugado	Aplicar conhecimento adquirido nas tarefas usando método Desplugado em método Plugado (Scratch)	Momento presencial	3 Encontros
Carga horária total			19 Encontros

Fonte: Elaborado pelo autor.

### 3.3.1. Tarefa 1

#### *T1.1 Conteúdos*

Nesta etapa os assuntos abordados são os tópicos:

- Definição do Pensamento Computacional;
- Passos básicos na solução de problemas algorítmicos para projetar soluções;
- Avaliar maneiras pelas quais diferentes algoritmos podem ser usados para resolver o mesmo problema;

## *T1.2 Metodologia*

Preparar os estudantes e certificar que todos tenham o material necessário para a aula.

## *T1.3 Guia de Ensino*

Nesta atividade os estudantes irão conhecer sobre o Pensamento Computacional e como aplicá-lo para a solução de problemas simples e do dia a dia. De maneira simplificada, os estudantes irão escrever comandos em texto, que deverão funcionar como comandos para resolver o problema.

### *T1.3.1 Atividade principal*

Os estudantes deverão criar uma listagem de passos para montagem de sanduíche. Usarão inicialmente lápis e papel para construção das instruções de montagem do sanduíche.

Após a construção das instruções, o professor irá interpretar e implementar as instruções criadas pelos estudantes. Uma boa sugestão é que os estudantes trabalhem em duplas, de modo que os participantes trabalhem em conjunto para criar as instruções. Cabe ao professor observar como eles estão trabalhando e interagindo para garantir que ambos participem ativamente da construção das instruções.

Se os estudantes apresentarem dificuldades e não conseguirem seguir adiante, o professor deve discutir com eles para descobrir o que aconteceu e o que esperavam que acontecesse. As duplas também podem pensar juntas sobre outras coisas que podem tentar fazer.

## *T1.4 Avaliação*

Depois que todos os estudantes finalizarem os projetos o professor deve aplicar a avaliação com debate e discutir como os estudantes conseguiram visualizar na prática o Pensamento Computacional e como eles utilizaram esse conceito para criar as instruções.

### 3.3.2. Tarefa 2

#### *T2.1 Conteúdos*

Nesta atividade os assuntos abordados são os tópicos:

- Passos básicos na solução de problemas algorítmicos para projetar soluções;
- Avaliar maneiras pelas quais diferentes algoritmos podem ser usados para resolver o mesmo problema;
- Definição, entendimento e aplicação de estruturas;

#### *T2.2 Metodologia*

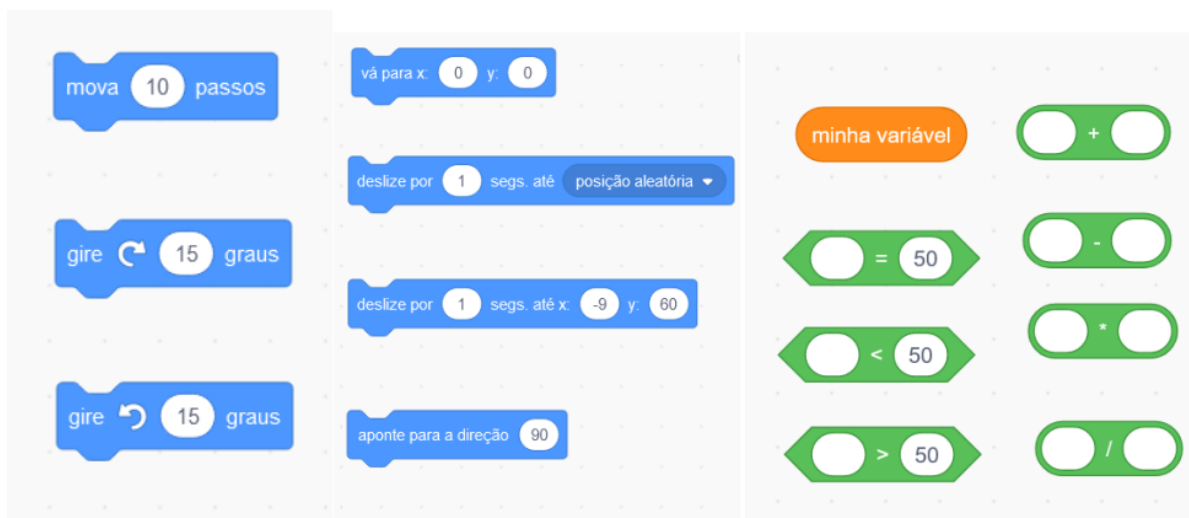
Preparar o laboratório para os estudantes e certificar que todos tenham o material para a realização da tarefa.

- Caneta/lápis;
- Papel;
- Lápis de cor;

#### *T2.3 Guia de Ensino*

Nesta tarefa os estudantes deverão entender o que é um comando em um “programa”. De maneira simples os estudantes irão passar comandos de um texto/instruções, que deverão ser convertidos seguindo esses comandos. A seguir alguns exemplos de possíveis comandos:

Figura 10 – Exemplos de comandos usando a notação do Scratch.



Fonte: Elaborado pelo autor.

Os estudantes irão usar modelos em blocos de comandos para “programar”.

### *T2.3.1. Atividade principal*

Os estudantes criarão um jogo em que os personagens disputam corrida. Para isso, os estudantes irão seguir o passo a passo abaixo, conforme o professor demonstra no telão:

- Desenhar um personagem no papel e recortar;
- Usar o bloco de comando onde a pista de corrida possui um total de 20 passos, e a cada 5 passos é obrigatório girar 90 graus para direita;
- A quantidade de passos é fornecida ao girar um dado;

A partir disso os estudantes possuirão um certo conhecimento com o processo e já entenderão como é possível fazer a conversão de instruções para programação em bloco.

Atenção: Deixar claro para os estudantes que os blocos são abordados nas próximas tarefas.

Peça para os estudantes executarem o código e testarem se este está funcionando. O professor apresenta em cartolina a pista de corrida. É esperado que alguns estudantes tenham dificuldades ou tenham algum problema nessa atividade, sendo



necessário acompanhar os estudantes e certificar que todos conseguiram desenvolver o código corretamente.

#### *T2.4 Avaliação*

Depois que todos os estudantes finalizarem o jogo, o professor deve aplicar um debate para avaliação e discutir com os estudantes sobre o seguinte:

- Quais tipos de problemas podemos resolver com algoritmos?
- Qual problema real, sem saber, já usava algoritmos para resolver?
- Quais tipos de dados o computador pode representar?

### **3.3.3. Tarefa 3**

#### *T3.1 Conteúdos*

Nesta atividade os assuntos abordados são:

- Definição, entendimento e aplicação de padrões;
- Estrutura de seleção: se simples;
- Estrutura de seleção: se... senão com alternativa.

#### *T3.2 Metodologia*

Preparar o laboratório para os estudantes e certificar que todos tenham o material para a realização da tarefa.

- Caneta/lápis;
- Papel;
- Lápis de cor;

#### *T3.3 Guia de Ensino*

Esta atividade desenvolve a capacidade de saber se uma condição é verdadeira ou falsa. Os estudantes aprenderão a usar a estrutura do SE, que mostra quando um determinado comando deve ser executado. Existe uma ampliação, com o SE / SENÃO, estrutura essa que será usada quando um comando deve ser executado e o que será executado quando a condição inicial não for satisfeita. Apresentando os blocos de comando SE e SENÃO:

Figura 11 – Exemplos de comandos usando a notação do Scratch - Bloco SE e SE SENÃO.



Fonte: Elaborado pelo autor.

### *T3.3.1 Atividade principal*

Criar algumas regras com a classe relacionadas aos comandos da programação em bloco. Um exemplo de regra: Perguntar a resposta de uma operação matemática, como  $6 \cdot 8$ . Caso a resposta for correta, ou seja, 48, deve ser apresentada uma mensagem: "Parabéns, você respondeu corretamente!". Podemos usar o Senão quando a resposta estiver errada, com a mensagem: "Poxa, infelizmente você não acertou".

O professor pode dividir a turma em grupos e fazer com que os estudantes discutam e criem operações usando as estruturas apresentadas.

### *T3.4 Avaliação*

Debater com os estudantes de forma avaliativa, as seguintes questões:

- O que é condicional?
- Como as condicionais podem ser usadas na programação em blocos?
- Podemos criar uma outra forma de condicional?

### 3.3.4. Tarefa 4

#### *T4.1 Conteúdos*

Nesta atividade os assuntos abordados são:

- Estrutura de seleção: escolha sequencial (se... senão, se);
- Estrutura de seleção: série de possibilidades (selecionar ou caso...);

#### *T4.2 Metodologia*

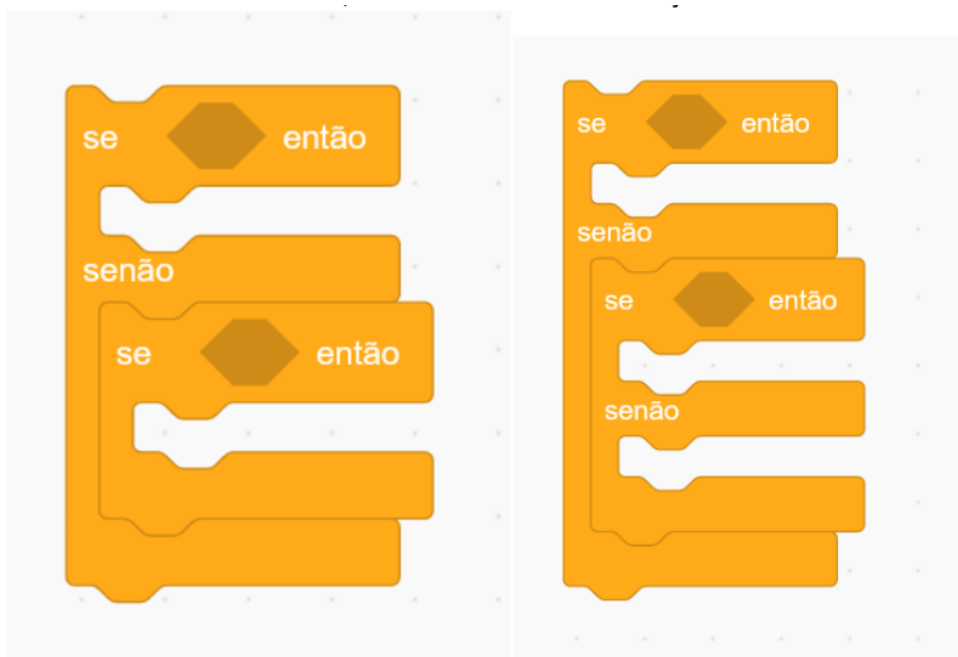
Preparar o laboratório para os estudantes e certificar que todos tenham o material para a realização da tarefa.

- Caneta/lápis;
- Papel;
- Lápis de cor;

#### *T4.3 Ensino guiado*

Na tarefa anterior os estudantes apresentaram as primeiras estruturas de decisão: o “SE” e o “SE...SENÃO”. Nesta, o objetivo é apresentar uma nova estrutura de decisão, o “SE... SENÃO SE”. O raciocínio dessa nova estrutura é semelhante ao da estrutura vista anteriormente. Primeiro é verificada uma condição. Caso a condição seja verdadeira, o código é executado; e caso seja falsa, outro comando é executado – contudo, quando a condição é falsa, ao invés de apenas executar um outro comando, inserimos uma outra condição.

Figura 12 – Exemplos de comandos usando a notação do Scratch - Bloco SE e SE SENÃO.



Fonte: Elaborado pelo autor.

Na programação em blocos, basta usar uma a junção de dois blocos de comandos já estudados, desta forma podemos unir as condições de duas formas distintas:

- Utilizando a estrutura SE...SENÃO e na condição SENÃO fazendo uso de uma estrutura SE, causando a verificação de duas condições e, caso ambas falhem, nada acontecerá;
- Utilizando a estrutura SE...SENÃO e na condição SENÃO fazendo uso de uma estrutura SE...SENÃO, causando a verificação de duas condições e mantendo um comportamento previsto caso ambas falhem;

É preciso firmar bem essa diferença para os estudantes, além de garantir que a segunda estrutura funcione como a estrutura de múltiplas escolhas.

#### *T4.3.1 Atividade principal*

O exercício prático dessa etapa propõe pedir aos estudantes que desenhem um boneco e caprichem na sua roupa. Depois, o professor, poderá apresentar um balão, perguntando se a cor da CAMISA do personagem é branca. Os estudantes devem criar codificar o programa seguindo os passos a seguir:

- Se a cor da camisa é branca:

A resposta deve ser apresentada: Sim, minha camisa é branca e é a minha cor favorita para roupas

- Senão, se a cor da camisa for azul:

A resposta deve ser apresentada: Não, eu prefiro a cor Azul para minhas camisas.

- Senão, se a cor da camisa for rosa:

A resposta deve ser apresentada: Não, minha camisa tem a cor rosa.

O professor pode dividir a turma em equipes para que os estudantes possam debater e criar novas ideias.

#### *T4.4 Avaliação*

A avaliação e o debate entre o professor e os estudantes, segue:

- Qual a diferença do SE...SENÃO para o SE...SENÃO SE?
- Se seria possível colocar mais condições?
- Que outras coisas poderíamos fazer para utilizar a múltipla escolha?

### **3.3.5. Tarefa 5**

#### *T5.1 Conteúdos*

Nesta atividade os assuntos abordados são os tópicos:

- Estrutura de repetição: meio laço: enquanto... faça ou faça... enquanto;
- Estrutura de repetição: laço completo: para.

#### *T5.2 Metodologia*

Preparar o laboratório para os estudantes e certificar que todos tenham o material para a realização da tarefa.

- Caneta/lápis;
- Papel;
- Lápis de cor;

### T5.3 Guia de Ensino

Nesta tarefa, o foco é nas estruturas de repetição. A primeira estrutura apresentada será o ENQUANTO... FAÇA. Esse padrão tem por objetivo trabalhar com ações que devem ser executadas várias vezes e, geralmente, não se sabe exatamente quantas vezes essa ação deve ser executada. Sendo assim, usamos uma verificação, uma condição para definir quantas vezes o comando será repetido.

Utilizar exemplos do mundo real para mostrar para os estudantes como funciona essa estrutura é muito importante.

a) No mundo real:

Quando você pede para os seus pais para poder assistir TV e eles dão a seguinte resposta: Certo, pode! Mas antes você deve lavar as louças. Enquanto existir louças sujas, você não pode assistir TV.

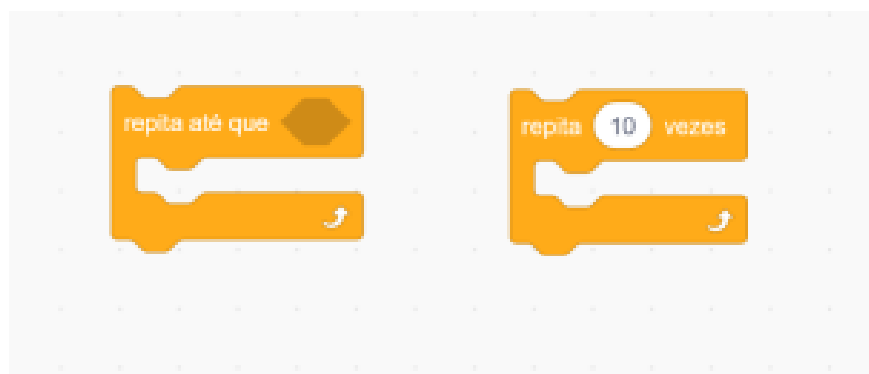
Outra possibilidade que pode ser abordada nesta tarefa é a estrutura PARA. Essa estrutura é utilizada em casos em que uma ação precisa ser repetida um número vezes já definido.

b) No mundo real:

Escovar os dentes é uma ação que deve ser executada no mínimo três vezes ao dia.

Apresentando os blocos de comando de repetição:

Figura 13 – Exemplos de comandos usando a notação do Scratch. Bloco de repetições.



Fonte: Elaborado pelo autor.

### *T5.3.1 Atividade principal*

Criar algumas regras com a classe, relacionadas aos comandos da programação em bloco apresentados. A turma pode ser dividida em equipes para que os estudantes possam debater e criar suas instruções em blocos. Podemos usar os exemplos da vida real apresentados e aguardar como os estudantes irão apresentá-los.

### *T5.4 Avaliação*

Aplicar o questionário de avaliação e discutir com os estudantes sobre:

- Qual a diferença entre o ENQUANTO e o PARA?
- Qual o nome do bloco de comando da estrutura PARA dentro da programação em bloco?
- Você teve dificuldades com alguma das duas estruturas?

### **3.3.6. Tarefa 6**

#### *T.6.1 Conteúdos*

Nesta atividade os assuntos abordados são:

- CONTADOR ou ACUMULADOR.

#### *T6.2 Metodologia*

Preparar o laboratório para os estudantes e certificar que todos tenham o material para a realização da tarefa.

- Caneta/lápis;
- Papel;
- Lápis de cor;

#### *T6.3 Guia de Ensino*

Nesta tarefa é apresentada a estrutura de Contador de Valores. Nos processos computacionais a contagem ou acumulação ocorrem em diversas situações. Um

exemplo é quando se deseja saber o número de vezes que determinada operação foi executada, para o que podemos usar um contador de um em um ou acumular o valor desejado. Podemos apresentar e discutir o seguinte bloco de comandos como exemplo:

Figura 14 – Exemplos de comandos usando a notação do Scratch. Bloco acumuladores e contadores.

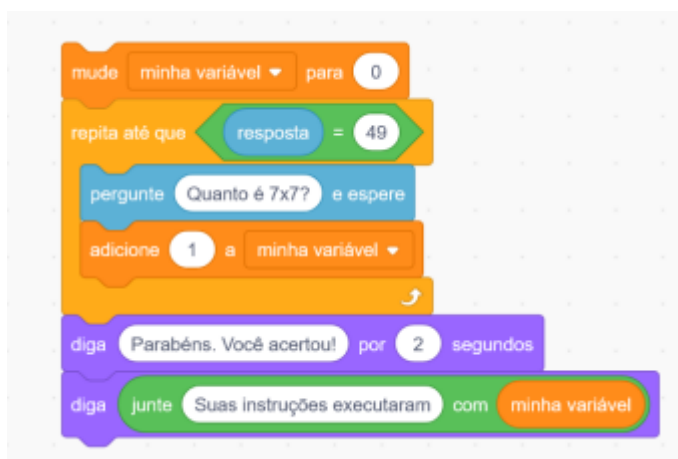


Fonte: Elaborado pelo autor

### T6.3.1 Atividade principal

O professor pode dividir a turma em equipes para que os estudantes possam debater e criar as instruções. Aqui está um exemplo do comando:

Figura 15 – Exemplos de comandos usando a notação do Scratch. Bloco acumuladores e contadores.



Fonte: Elaborado pelo autor



### *T6.4 Avaliação*

Discutir com os estudantes sobre:

- Existe diferença entre o CONTADOR e o ACUMULADOR?
- Quais exemplos do mundo real onde encontramos estas instruções?

### **3.3.7. Tarefa 7**

#### *T.7.1 Conteúdos*

Nesta atividade os assuntos abordados são:

- Conhecer a ferramenta Scratch;
- Implementação de tarefas anteriores em método plugado;

#### *T7.2 Metodologia*

Preparar o laboratório para os estudantes e certificar que todos tenham o material para a realização da tarefa.

- Computador;
- Scratch instalado;

#### *T7.3 Guia de Ensino*

Nessa tarefa será efetuada a apresentação da tela do Scratch e realizada a migração das tarefas anteriores, criadas em papel usando o método desplugado, para a ferramenta do Scratch, que utiliza o método plugado.

##### *T7.3.1 Atividade principal*

Conversão das tarefas do papel para o meio plugado do Scratch. Recomenda-se aproveitar da ludicidade do momento para promover a interação entre os estudantes, para que possam se sentir confortáveis em explorar a ferramenta, compartilhar os resultados com os colegas, executar o programa e ver os resultados em tempo real do que foi desenvolvido inicialmente no caderno.

Além disso, é importante que o professor encoraje a exploração através de ajustes no código, pedindo, por exemplo, que os estudantes alterem alguma condição ou a quantidade de repetições, para que possam ver na prática os resultados de suas alterações e melhor compreender como suas intervenções nos blocos do Scratch impactam no resultado final.

É encorajado aos estudantes a livre exploração da ferramenta, inclusive fazendo ações inusitadas e divertidas que possam promover o engajamento entre os estudantes. É importante que o professor esteja apto a responder às perguntas dos estudantes e ensinar novas possibilidades no Scratch de acordo com o interesse dos estudantes, despertando o interesse e as potencialidades da ferramenta.

#### *T7.4 Avaliação*

Discutir com os estudantes sobre:

- O processo de migração do meio desplugado (papel) para o meio plugado (programa Scratch);
- Debater com os estudantes qual melhor forma e se realmente existe a necessidade de tornar tudo plugado;
- Discutir sobre os resultados finais obtidos pelos estudantes, após o período de livre exploração da ferramenta.

#### 4. CONCLUSÃO

Neste trabalho foi possível abordar nosso objeto de estudo em duas óticas diferentes. Por um lado, uma pesquisa sobre o ensino do Pensamento Computacional para os estudantes das séries iniciais e um estudo sobre as ferramentas computacionais existentes para este objetivo. Por outro, a elaboração de uma proposta pedagógica que fizesse uso dos conceitos estudados, se beneficiando das técnicas, tecnologias e metodologias já comprovadas em outros trabalhos publicados para potencializar o processo de ensino-aprendizagem na execução da proposta, acrescentando critérios que consideramos relevantes, como a preocupação com o plugado e desplugado e os recursos disponíveis para aplicação da proposta.

Desta forma, nosso foco foi a criação de uma proposta que pudesse servir não apenas como ponto de partida para interessados em introduzir o Pensamento Computacional nas escolas, mas também como um modelo para adaptações pelos professores envolvidos. Esta pesquisa também propôs elaborar uma proposta praticável e com custo baixo do ensino do Pensamento Computacional em vários níveis de ensino e de infraestrutura.

Foi iniciada uma proposta de metodologia para ensinar o Pensamento Computacional para estudantes dessas séries, através da programação em bloco tanto desplugada quanto com o uso das ferramentas Scratch.

A aplicação do processo de ensino pode ser realizada de várias formas e com vários níveis. Ao professor é recomendado que tenha conhecimento prévio e que tenha treinamento no uso das ferramentas e do Pensamento Computacional, para assim possibilitar uma boa disseminação do conhecimento e ajudar os estudantes no processo de construção do aprendizado.

O esforço empenhado na natureza dupla deste trabalho - envolvendo a pesquisa e o desenvolvimento da proposta - não possibilitou uma efetiva aplicação da proposta em uma sala de aula para avaliação dos resultados, o que exige confiança nos resultados encontrados nos trabalhos citados, assim como nas propostas feitas no presente trabalho.

#### 4.1. TRABALHOS FUTUROS

Entendemos que este trabalho pode se beneficiar consideravelmente dos trabalhos futuros abaixo descritos, tornando a proposta mais confiável, robusta e englobando novas metodologias e tecnologias.

- Aplicar a proposta pedagógica desenvolvida nas escolas de ensino das séries iniciais para coletarmos dados sobre sua eficiência em um ambiente real;
- Aprimorar constantemente a metodologia com base nas aplicações e coletas de dados futuros, nunca se esquecendo dos valores teóricos aqui estudados;
- Incorporar novas metodologias ativas que possam engajar e motivar ainda mais os estudantes, colocando-os como autores no processo de construção do conhecimento;
- Agregar conceitos de gamificação para tornar a experiência ainda mais atrativa e motivadora para os estudantes, promovendo uma competição saudável, desenvolvimento de relações interpessoais, entre outros benefícios;
- Acompanhar a longo prazo o desempenho dos estudantes após aprendizado do Pensamento Computacional e verificar as consequências que o desenvolvimento desta competência acarreta não apenas no interesse e habilidades nas áreas exatas, mas em todas as outras áreas do conhecimento.

## REFERÊNCIAS

ALVARENGA, Rosana Cristina Macelloni. **O raciocínio lógico e a criatividade na resolução de problemas matemáticos no Ensino Médio**. 2008. 99 f. Dissertação (Mestrado em Educação) - Universidade Estadual Paulista, Faculdade de Filosofia e Ciências, Marília, 2008. Disponível em: <https://repositorio.unesp.br/items/dee673ba-a829-4f51-a24e-e582e0c8ba37>. Acesso em: 01 maio 2023.

ARAUJO, Ana Liz; ANDRADE, Wilkerson; GUERRERO, Dalton. Um Mapeamento Sistemático sobre a Avaliação do Pensamento Computacional no Brasil. *In: CBIE 2016: V Congresso Brasileiro de Informática na Educação*, 2016, Uberlândia, MG. **Anais dos Workshops do V Congresso Brasileiro de Informática na Educação (CBIE 2016)**, v. 1, n. 1. Uberlândia: Sociedade Brasileira de Computação – SBC, 2016, p. 1147-1158. DOI: <http://dx.doi.org/10.5753/cbie.wcbie.2016.1147>.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018.

CODE. **Code**. 2024a. Disponível em: <https://code.org/>. Acesso em: 01 maio 2023.  
\_\_\_\_\_. **Hour of Code**. 2024b. Disponível em: <https://hourofcode.com/br/learn>. Acesso em: 01 maio 2023.

GLIZT, Fabiana Rodrigues de Oliveira. **O Pensamento Computacional nos anos iniciais do Ensino Fundamental**. 2017. 92 f. Dissertação (Mestrado em Ensino de Ciência e Tecnologia) - Universidade Tecnológica Federal do Paraná, Ponta Grossa, 2017.

GREBOGY, Elaine Cristina; SANTOS, Icleia; CASTILHO, Marcos Alexandre. Mapeamento das Iniciativas de Promoção do Pensamento Computacional no Ensino Fundamental. *In: SIMPÓSIO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO*, 32. 2021, Online. **Anais...** Porto Alegre: Sociedade Brasileira de Computação, 2021, p. 965-975.

MAIN LEAF. Como criar um jogo no Scratch: Primeiros passos. 11 jan. 2022. **Main Leaf**. Disponível em: <https://mainleaf.com/pt/como-criar-um-jogo-no-scratch/>. Acesso em: 02 maio 2023.

MOTTA, Sérgio. Ensine seus filhos a programar com o Scratch. 9 jan. 2023. **Soft Download**. Disponível em: <https://www.softdownload.com.br/ensine-programacao-filhos-criancas-scratch.html>. Acesso em: 01 maio 2023.

OLIVEIRA, Vinícius de. EUA lançam plano para ensinar todo mundo a programar. 07 fev. 2016. **Cidade Nova**. Disponível em: [https://www.cidadenova.org.br/editorial/informa/2274-eua\\_lancam\\_plano\\_para\\_ensinar\\_todo\\_mundo](https://www.cidadenova.org.br/editorial/informa/2274-eua_lancam_plano_para_ensinar_todo_mundo). Acesso em: 01 maio 2023.

PEREIRA, Wellington G.; FRANÇA, Rozelma Soares de. **Ensino de Computação na Educação Básica: Onde está Paulo Freire?**. *In: SIMPÓSIO BRASILEIRO DE*

INFORMÁTICA NA EDUCAÇÃO, 33. 2022, Manaus. **Anais...** Porto Alegre: Sociedade Brasileira de Computação, 2022, p. 1404-1414.

REDE BRASILEIRA DE APRENDIZAGEM CRIATIVA – RBAC. **Aprendizagem Criativa**. 2021. Disponível em: <https://aprendizagemcriativa.org/atividade/minha-historia-em-blocos-scratch-desplugado?page=0>. Acesso em: 12 abr. 2023.

RODRIGUES, Elisângela Valevein; SOUZA, Leandro Delgado de. Instituto de Hackers: o Pensamento Computacional aplicado ao Ensino Médio Integrado Profissionalizante. **Informática na Educação: Teoria & Prática**, Porto Alegre, v. 24, n. 1, p. 115-128, 2021. Disponível em: <https://seer.ufgs.br/InfEducTeoriaPratica>. Acesso em: 20 mar. 2023.

VILAÇA, Márcio Luiz Corrêa; ARAUJO, Elaine Vasquez Ferreira de. **Tecnologia, Sociedade e Educação na Era Digital**. Duque de Caxias: Unigranrio, 2016. ISBN: 978-85-88943-69-8.

ZANETTI, Humberto Augusto Piovesana; BORGES, Marcos Augusto Francisco; RICARTE, Ivan Luiz Marques. Pensamento Computacional no Ensino de Programação: Uma Revisão Sistemática da Literatura Brasileira. *In*: CBIE 2016: V Congresso Brasileiro de Informática na Educação, 2016, Uberlândia, MG. **Anais do XXVII Simpósio Brasileiro de Informática na Educação (SBIE 2016)**. Uberlândia: Sociedade Brasileira de Computação – SBC, 2016, p. 21-30. Disponível em: <http://milanesa.ime.usp.br/rbie/index.php/sbie/article/view/6677>. Acesso em: 12 abr. 2023.

ZANETTI, Humberto Augusto Piovesana; OLIVEIRA, Cláudio Luís Vieira. Prática de ensino de Programação de Computadores com Robótica Pedagógica e aplicação de Pensamento Computacional. *In*: CBIE 2015: IV Congresso Brasileiro de Informática na Educação, Maceió, AL. **Anais dos Workshops do IV Congresso Brasileiro de Informática na Educação (CBIE 2015)**. Maceió: Sociedade Brasileira de Computação – SBC, 2015, p. 1236-1245.