

## **PROPOSTA DE OFICINA DE ENSINO UTILIZANDO A FERRAMENTA SCRATCH**

Halison Vieira Andrade

Igor Fujikawa Andrade

**Resumo:** Este artigo apresenta um estudo sobre o ensino de programação nas escolas com o objetivo de tornar o processo mais acessível e compreensível para as crianças. Muitas vezes a programação é percebida como algo complexo e difícil de ensinar, mas ao analisar as experiências de quem ensinou programação, incluindo suas metodologias, ferramentas e resultados, foi possível identificar práticas pedagógicas eficazes.

Como forma de identificar as experiências vividas nos artigos analisados, foram observados os relatos dos educadores, tabelas e gráficos que apontavam para os dados que foram gerados conforme a implementação de suas metodologias.

A pesquisa revelou que a linguagem de programação Scratch foi a ferramenta mais utilizada para o ensino de lógica de programação. O Scratch é uma linguagem visual e interativa que permite que as crianças criem projetos e jogos de forma intuitiva e divertida. A partir dessa constatação, foram desenvolvidas atividades específicas para alunos de 7 a 14 anos que incluem o uso do Scratch como ambiente de programação.

Essas atividades são projetadas para envolver os alunos na exploração da lógica de programação, estimular sua criatividade e tornar o processo de aprendizado mais divertido. Eles podem ser aplicados tanto em um ambiente escolar formal quanto em um ambiente de aprendizagem informal, como clubes de codificação ou atividades extracurriculares.

O objetivo final deste trabalho é contribuir para a difusão do ensino de programação para crianças e inspirar educadores e instituições de ensino a explorar esta área, que já tem se mostrado muito promissora.

**Palavras-chave:** Ensino de programação, Metodologias de ensino, Scratch.

Artigo apresentado como requisito parcial para a conclusão do curso de Ciência da Computação da Universidade do Sul de Santa Catarina – UNISUL, orientado pela professora Silvana Madeira Alves Dal-Bó.

Halison Vieira Andrade. E-mail: halison\_11@hotmail.com

Igor Fujikawa Andrade. E-mail: igorfujiaandrade@gmail.com

## TEACHING WORKSHOP PROPOSAL USING THE SCRATCH TOOL

**Abstract:** This article presents a study on teaching programming in schools with the aim of making the process more accessible and understandable for children. Programming is often perceived as something complex and difficult to teach, but by analyzing the experiences of those who taught programming, including their methodologies, tools and results, it was possible to identify effective pedagogical practices. As a way of identifying the experiences lived in the analyzed articles, the educators' reports, tables and graphs that pointed to the data that were generated according to the implementation of their methodologies were observed. The research revealed that the Scratch programming language was the most used tool for teaching programming logic. Scratch is a visual and interactive language that allows children to create projects and games in an intuitive and fun way. Based on this observation, specific activities were developed for students from 7 to 14 years old, which include the use of Scratch as a programming environment. These activities are designed to engage students in exploring programming logic, stimulate their creativity and make the learning process more fun. They can be applied both in a formal school setting and in an informal learning environment such as coding clubs or extracurricular activities.

The final objective of this work is to contribute to the dissemination of teaching programming to children and to inspire educators and educational institutions to explore this area, which has already shown to be very promising.

**Keywords:** Teaching programming, Teaching methodologies, Scratch.

## **1 INTRODUÇÃO**

Atualmente as tecnologias estão muito presentes na vida das pessoas, é notável que cada vez mais as crianças têm acesso a elas, fazendo com que aprendam a utilizá-las de forma muito rápida. Porém, o uso da programação no âmbito do ensino ainda é visto como algo muito complexo, que não é de fácil compreensão para as crianças, fazendo com que não seja considerada uma boa estratégia nas escolas.

No entanto, existem iniciativas educativas que incentivam crianças e jovens a aprender a programar, como no caso da organização Code.org, que ensina crianças a programar dentro da sua plataforma com cursos dirigidos ao ensino de crianças e jovens, além dos projetos autônomos cujo objetivo é mostrar os benefícios da programação para aumentar o interesse e contribuir para desmistificar a sua dificuldade de aprendizagem.

Considerando a problemática do ensino da programação, esse trabalho de pesquisa planeja responder a seguinte questão: Qual é a metodologia de ensino mais utilizada para ensinar programação levando-se em consideração a modalidade de ensino presencial?

Nesse contexto, essa pesquisa tem por objetivos identificar as técnicas, metodologias e ferramentas utilizadas com maior frequência no ensino da programação, e diante disso propor uma metodologia de oficina de ensino modelo para alunos com idade de 7 a 14 anos.

### **1.1 MÉTODO**

O presente trabalho realizou uma pesquisa exploratória tendo a finalidade de identificar as metodologias e conteúdo que estão sendo utilizados no ensino da programação no Brasil.

Gil (1999) considera que a pesquisa exploratória tem como objetivo principal desenvolver, esclarecer e modificar conceitos e ideias, tendo em vista a formulação de problemas mais precisos ou hipóteses pesquisáveis para estudos posteriores. Segundo o autor, estes tipos de pesquisas são os que apresentam menor rigidez no planejamento, pois são planejadas com o objetivo de proporcionar visão geral, de tipo aproximativo, acerca de determinado fato

Na pesquisa realizada foram coletados artigos científicos publicados no período de 2012 a 2023 baseados nos critérios de inclusão e exclusão que são detalhados na seção 3 em “Análise documental”. Foram incluídos na pesquisa, casos de uso que focaram no ensino de programação no contexto da educação.

Foram selecionados artigos que relataram experiências relacionadas ao ensino de programação, pensamento computacional e algoritmo. A escolha dos artigos a serem analisados ocorreu seguindo três etapas:

Primeira etapa: O primeiro passo foi selecionar os artigos pelo título. Nessa etapa foi realizada a leitura dos títulos dos trabalhos presentes nos meios de busca online. A seleção dos artigos foi feita manualmente com a leitura dos títulos e observando as palavras-chave que se adequam ao nosso estudo. Segunda etapa: Após a coleta dos artigos baseados pelo título foi feita a utilização de alguns critérios de inclusão e exclusão que serão apresentados posteriormente no Quadro 1. Após passar pelos critérios de inclusão foi realizada a leitura de todos os resumos dos artigos selecionados, além da metodologia de ensino aplicada em cada um deles. Terceira etapa: Foi realizada a leitura da conclusão dos artigos e a captação dos dados para tabular os resultados.

O trabalho está organizado da seguinte maneira: na seção 2 é apresentado o referencial teórico. Na Seção 3 é apresentada a análise documental. Na Seção 4 é apresentada a proposta de uma metodologia para uma oficina de ensino modelo. Por fim, na Seção 5 são apresentadas as considerações finais.

Acreditamos que o resultado da pesquisa contribua com a visão de que o uso do ensino de programação nas escolas é algo que ajudará no desenvolvimento intelectual e cognitivo dos alunos, fazendo com que desenvolvam seu raciocínio lógico, despertem o interesse pelo assunto e aprendam a analisar de forma mais clara como resolver problemas.

## **2. REFERENCIAL TEÓRICO**

### **2.1 PENSAMENTO COMPUTACIONAL**

O pensamento computacional representa o processo de reconhecimento de aspectos de computação no mundo que nos rodeia, aplicando ferramentas e técnicas da ciência da

computação para entender e raciocinar sobre sistemas e processos naturais e artificiais (FURBER, 2012 apud BEECHER, 2019).

Diversos métodos e modelos computacionais podem ser utilizados para resolver problemas em diferentes níveis de complexidade e projetar sistemas que os humanos não conseguem realizar, o que traz à tona a inteligência de cada um dos componentes envolvidos nesse processo. Portanto, o pensamento computacional pode ser definido como uma habilidade fundamental que todos nós possuímos para qualquer tarefa que precise ser realizada, a atividade mental para abstrair problemas e formular soluções que podem ser automatizadas (WING, 2014 apud BEECHER, 2019).

## 2.2 PROGRAMAÇÃO PARA CRIANÇAS

Estudar programação pode ser considerado como algo difícil ainda mais nas séries iniciais. Mas quando estudam programação, as pessoas não só aprendem a programar, como também programam para aprender, noções de programação devem ser ensinadas desde cedo às crianças (Serrano, 2014).

O estudo da programação ajuda “[...] a desenvolver habilidades para auxiliar na resolução de problemas, ser um elemento ativo na construção do conhecimento e entender o contexto tecnológico em que vivemos”, (ZANETTI; BORGES; RICARTE, 2016, p. 21).

Aprender a programar ajuda a criança na análise lógica, aprimora a capacidade de raciocínio e apesar de ser uma área exata, a computação também desafia constantemente a imaginação e criatividade. Segundo Kafai e Burke (2013), o ensino de programação para crianças poderia desenvolver o pensamento computacional e os passos lógicos para a resolução automatizada de problemas.

Uma habilidade que é bastante desenvolvida em programação para crianças é o pensamento computacional. De acordo com a visão de Wing(2006), o pensamento computacional é o processo de reconhecer aspectos da computação no mundo que nos rodeia e aplicar ferramentas e técnicas para entender e raciocinar sobre sistemas e processos naturais, sociais e artificiais.

O pensamento computacional, no âmbito escolar, possibilita que os alunos solucionem problemas por partes, abstraíam o que não for necessário para a conclusão de uma tarefa, saibam reconhecer padrões existentes em diferentes problemas e criem algoritmos para solucioná-los.

Segundo Selby e Woollard (2013), o pensamento computacional é um processo cognitivo ou de pensamento que envolve o raciocínio lógico pelo qual os problemas são resolvidos e os artefatos, procedimentos e sistemas são mais bem compreendidos.

## 2.3 SCRATCH

Para ensinar programação para crianças é importante fazer o uso de ferramentas que possam gerar o interesse e assim dinamizar o processo de aprendizagem. Um exemplo de ferramenta com interface voltada para o público infantil é o Scratch, projeto do grupo Lifelong Kindergarten do Media Lab do Massachusetts Institute of Technology (MIT). A ferramenta foi pensada para crianças, mas passou a ser utilizada por um público diferente por se tratar de uma ferramenta utilitária com diversas funções para programação baseada em blocos.

Segundo Kelleher e Pausch (2005), a programação baseada em blocos busca uma sintaxe abstrata para que iniciantes foquem apenas na semântica. Ele incentiva o uso de estratégias para usar a computação criativa em ambientes educacionais. A programação de blocos funciona arrastando e soltando componentes, que são representados como blocos. Ao construir um programa, considere que cada bloco tem um encaixe específico, vários blocos se encaixam como um quebra-cabeça, pois a linguagem de programação em blocos permite ao usuário visualizar onde cada bloco pode se encaixar (embora em alguns casos, dependendo da ferramenta, os blocos podem não estar instalados corretamente) (Weintrop et al., 2017; Kelleher e Pausch, 2005).

Com o uso da ferramenta Scratch, será possível ensinar às crianças desde cedo que a tecnologia não serve apenas para entretenimento, mas também para aprender e usar a criatividade. Ao apoiar a criação de projetos utilizando esta linguagem, as crianças poderão desenvolver a criatividade desde a fase de desenvolvimento de uma ideia até o desenvolvimento completo de um projeto, passando por diferentes etapas e áreas do conhecimento como gerenciamento de projetos, lógica de programação e até design, pois o ambiente e os personagens usados nos projetos Scratch podem ser criados pelas próprias crianças.

Com o Scratch, as crianças podem desenvolver a habilidade abstração e resolver problemas enquanto se divertem. Essa habilidade consiste em pensar em partes isoladas de um problema para assim facilitar o desenvolvimento da solução. É uma habilidade que faz parte de uma metodologia de resolução de problemas e exige muito foco e concentração. Com sua

metodologia de elaboração de projetos, o Scratch faz muita diferença no desenvolvimento dos alunos porque estimula a colaboração e o trabalho em equipe, principalmente em turmas com mais alunos.

Os projetos podem ser divididos em fases e grupos que trabalham juntos para desenvolver diferentes partes do software que está sendo construído até que esteja pronto. Além disso, a plataforma online própria do Scratch suporta a colaboração virtual através da possibilidade de interação do aluno com o programa de outra pessoa, por meio de comentários e até atualizações ao software desenvolvido por outra pessoa. Posteriormente na figura 1 é mostrado um exemplo de um programa na linguagem Scratch que resulta na idade de uma pessoa, com o cálculo baseado na subtração do ano atual com o ano de nascimento do indivíduo.

Figura 1 – Calculando a idade do usuário em Scratch



Fonte: Wagner Gaspar, 2023.

## 2.4 LINGUAGEM LOGO

O Logo é uma linguagem de programação criada por Seymour Papert no Massachusetts Instituto de Tecnologia (MIT) para auxiliar no processo de aprendizado e raciocínio. Nela o aluno é levado a pensar na solução de problemas de forma direta ou através da correção de seus próprios erros.

No ambiente de programação logo, se utiliza uma tartaruga robótica, que anda por uma sala sendo direcionada por meio de comandos. Na linguagem se utiliza o comando “para frente”

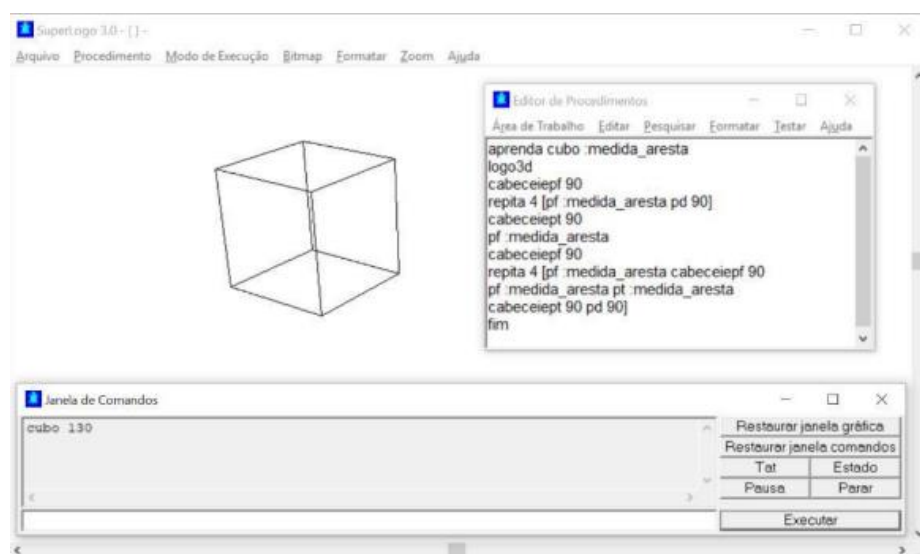
e “paratras” acompanhado o tamanho do passo, e também o “paradireita” e “paraesquerda” acompanhado do ângulo de giro.

Ao se movimentar pela tela a tartaruga desenha uma linha reta do tamanho exato dos passos expressos pelos comandos. Dessa forma, é possível desenhar várias figuras geométricas.

Existem vários outros comandos que podem ser usados para criar os mais diferentes tipos de desenhos no Logo. Também é possível criar um procedimento, o qual é um conjunto de instruções inseridas pelo usuário que recebem um nome e pode ser executado na linguagem para executar a tarefa. Os procedimentos são semelhantes às funções em outras linguagens de programação.

Ainda hoje, a linguagem logo é amplamente utilizada em escolas de todo o mundo, não apenas uma ferramenta de aprendizado para iniciantes em programação, principalmente auxilia o professor no processo de aprendizagem e raciocínio de outros conceitos de diversas áreas. A figura 2 que segue mostra um exemplo de linhas de código em Logo para desenhar um quadrado.

Figura 2 – Exercício usando comandos para desenhar um quadrado em logo



Fonte: Seer.ufrgs, 2023.

## 2.5 CODECOMBAT

CodeCombat Inc. é uma empresa de desenvolvimento de TI que criou o jogo RPG de navegador de mesmo nome.



Esse jogo ensina várias linguagens de programação como JavaScript e Python, bem como os fundamentos da Ciência da computação. A empresa foi fundada em fevereiro de 2013 por George Saines, Scott Erickson e Nick Winter, que anteriormente desenvolveram o aplicativo de aprendizado de idiomas Skritter.

Originalmente, o site era gratuito e a empresa planejava se financiar por meio de comissões com o recrutamento dos desenvolvedores que ela ajudou a descobrir. Depois de um tempo, a CodeCombat colocou uma assinatura paga que desbloqueia o acesso a conteúdo adicional no jogo.

CodeCombat é um jogo educacional que ensina programação. É recomendado para crianças de 9 a 16 anos, porém qualquer pessoa interessada em aprender a programar pode utilizar esta ferramenta. O jogo funciona por meio de desafios nos quais você deve escrever códigos de programação relacionados às ações do seu personagem, (FRWIKI.WIKI, 2016).

Na figura 3 que vem posteriormente, é demonstrada uma fase do jogo onde o jogador precisa digitar os códigos para coletar as moedas pelo caminho e correr até o final sem ser encontrado pelo inimigo.

Figura 3 – Exemplo de uma fase do jogo CodeCombat



Fonte: CodeCombat , 2023.

## 2.6 CODE.ORG

O Code.org é uma organização sem fins lucrativos lançada em janeiro de 2013 pelos irmãos iraniano-americanos Hadi Partovi e Ali Partovi com foco em tornar a programação de computadores mais acessível. Sua primeira ideia era criar um banco de dados com todas as salas de aula de ciência da computação nos Estados Unidos. Na época, Hadi Partovi afirmou que cerca de noventa por cento das escolas dos Estados Unidos não ensinavam programação apesar de agora ser um campo fundamental.

Em 2011, Hadi teve a ideia de criar um vídeo que estimulasse os jovens a aprender a programar, usando as palavras de personalidades do mundo da tecnologia. A primeira pessoa a enviar um depoimento seria Steve Jobs, mas ele morreu antes do início do projeto. Então Partovi chegou à conclusão de que precisava fazer algo mais eficaz do que apenas fazer vídeos.

No início o Code.org recebeu o apoio de 60 pessoas, entre elas Bill Gates, Mark Zuckerberg, Jack Dorsey e ainda o cantor Will.i.am e alguns políticos como Al Gore e Michel Bloomberg. Conseguiram então o apoio financeiro do Google, da Microsoft, da Amazon e do LinkedIn.

Segundo Partovi, a decisão de dedicar-se integralmente ao projeto veio após saber que o vídeo sobre o projeto havia recebido mais de 11 milhões de visualizações na internet. Em 9 de dezembro de 2013, o Code.org lançou a iniciativa “Hour of Code 2013”, em português, “A

hora do código 2013”, em que qualquer pessoa poderia dedicar 1 (uma) hora para aprender a programação de computadores.

A plataforma utiliza um método de ensino estruturado, no qual os usuários passam por módulos de aprendizagem aumentando gradualmente sua dificuldade e exploração dos conteúdos. Além disso, o material oferecido pela Code.org é de fácil acesso e utilização, o que permite que os professores possam usá-lo em sala de aula eficientemente.

A ferramenta possibilita a exploração interativa de conceitos relacionados à lógica de programação e variáveis em um ambiente intuitivo, tornando o processo de aprendizagem mais fácil e acessível.

A plataforma tem a possibilidade de registo tanto para estudantes como para professores. Ela é nativamente desenvolvida em inglês, mas com suporte em português. Conta com atividades online, realizadas no próprio site, e offline, com atividades desplugadas. O diferencial desta plataforma é que possui cursos com conteúdo metodológico pré-definido e um ambiente específico para acesso dos educadores, a plataforma permite a emissão de certificado de conclusão.

Para os professores, é possível criar turmas e acompanhar o andamento das atividades realizadas por cada um dos alunos. O aprendizado acontece em todas as etapas da plataforma, pois em todas as etapas dos jogos há a possibilidade de trabalhar a sequência de passos necessários para resolver a tarefa.

### **3. RESULTADOS**

#### **3.1. ANÁLISE DOCUMENTAL**

Nessa etapa da pesquisa utilizou-se o motor de busca “Google Acadêmico”, onde se encontrou aproximadamente 30.000 artigos filtrando pelos anos de 2012 a 2023 por meio das palavras-chave: ensino da programação estudos de caso. Como resultado, obteve-se muitos resultados fora dos critérios predefinidos, ainda assim conseguimos levantar um total de 30 artigos para análise. A lista com os artigos selecionados será apresentada posteriormente no Quadro 2.

No Quadro 1 que segue são demonstrados os critérios de inclusão e exclusão de artigos utilizados na pesquisa.

Quadro 1 – Critérios de inclusão e exclusão de artigos

Critérios de inclusão	Critérios de exclusão
<ul style="list-style-type: none"> <li>• Artigos que relatam experiências em estudos de caso realizados no Brasil.</li> <li>• Artigos de 2012 a 2023 para análise das diferenças nas metodologias de ensino no passar dos anos.</li> <li>• Artigos escritos na língua portuguesa.</li> <li>• Artigos que relatam o uso de ferramentas ou metodologias para ensino de programação.</li> </ul>	<ul style="list-style-type: none"> <li>• Artigos fora do nosso escopo de estudo como revisão sistemática ou revisões bibliográficas.</li> <li>• Artigos que não apresentam estudos de caso.</li> <li>• Artigos que relatam estudos de caso, mas são aplicados para pessoas fora da faixa etária de interesse para o estudo.</li> </ul>

Fonte: Autores (2023).

Quadro 2 – Listagem dos artigos analisados com hiperlink de acesso

ID	Título	Autores	Ano
<b>1</b>	Proposta Metodológica para a Inserção ao Ensino de Lógica de Programação com Logo e Lego Mindstorms	Friedrich et al.	<b>2012</b>

<b>2</b>	Ensino de Programação no Ensino Médio: Uma Abordagem Orientada ao Design com a linguagem Scratch	Scaico et al.	<b>2013</b>
<b>3</b>	Squeak Etoys na modalidade 1 para 1: programação e autoria multimídia no desenvolvimento da conceituação	Sperb et al.	<b>2013</b>
<b>4</b>	Curso Híbrido usando a Rede Social Facebook no Ensino de Programação de Computadores	Mata et al.	<b>2014</b>
<b>5</b>	Ensino de Computação com SCRATCH no Ensino Fundamental – Um Estudo de Caso	Wangenheim et al.	<b>2014</b>
<b>6</b>	Uma Metodologia Baseada em Semiótica para Elaboração e Análise de Práticas de Ensino de Programação com Robótica Pedagógica	Zanetti et al.	<b>2014</b>
<b>7</b>	GameMaking: Uma Metodologia para o Ensino de Informática para Alunos do Ensino Fundamental através da criação de Jogos Digitais	Mello et al.	<b>2015</b>
<b>8</b>	Plataforma Robocode como Ferramenta Lúdica de Ensino de Programação de Computadores - Pesquisa e Extensão Universitária em Escolas Públicas de Minas Gerais	Amaral et al.	<b>2015</b>
<b>9</b>	Pensamento Computacional e Robótica: Um Estudo Sobre Habilidades Desenvolvidas em Oficinas de Robótica Educacional	Oliveira et al.	<b>2016</b>

<b>10</b>	Ensino de lógica de programação no ensino fundamental utilizando o jogo Robotizen: um relato de experiência	Dantas et al.	<b>2019</b>
<b>11</b>	Ensino de Programação em Escolas Públicas: Relato de uma Ação do PET-TI	Silva et al.	<b>2019</b>
<b>12</b>	Uso do ambiente code.org para ensino de programação no ensino fundamental i - uma experiência no desafio hora do código	Kaminski et al.	<b>2019</b>
<b>13</b>	Manas Digitais: um relato sobre Ensino de Programação em Escolas Públicas no Estado do Pará	Silva et al.	<b>2019</b>
<b>14</b>	Uma contribuição ao ensino de programação na Educação Básica	Pires et al.	<b>2019</b>
<b>15</b>	Avaliação de Ferramentas para Ensino de Programação para Crianças e Adolescentes	Pontes et al.	<b>2019</b>
<b>16</b>	BDM: Desenvolvimento de jogos para incentivar meninas na área da tecnologia de informação: um estudo de caso no ensino médio de escola pública (ufpa.br)	MATOS, Geisiane de Souza	<b>2019</b>
<b>17</b>	Metodologia Para Ensino do Pensamento Computacional para Crianças Baseada na Alternância de Atividades Plugadas e Desplugadas	Berto et al.	<b>2019</b>
<b>18</b>	O Pensamento Computacional no Ensino Fundamental I	Castilho et al.	<b>2019</b>

<b>19</b>	Ensino de algoritmos, programação e matemática: panorama e estudo de caso para estudantes de escolas públicas brasileiras	Lima et al.	<b>2019</b>
<b>20</b>	Ensino de Programação nas Escolas: Um relato de experiência	Montoanelli et al.	<b>2019</b>
<b>21</b>	Aprendizagem em programação de jogos: ampliação da cognição, produção criativa e autoral de alunos da educação básica (ufal.br)	SILVA, Adelson da	<b>2020</b>
<b>22</b>	Propostas de ensino de programação e experiências na educação básica: diálogos	OLIVEIRA, Aline Bersa Santojo	<b>2020</b>
<b>23</b>	Programação em Blocos: impacto de um projeto de extensão executado em Escolas públicas de Diamantina/MG	Batista et al.	<b>2020</b>
<b>24</b>	Repositório Institucional da UnB: Programação e pensamento computacional no 8º e 9º ano do Ensino Fundamental : um estudo de caso	PEREIRA, João Pedro de Lima	<b>2020</b>
<b>25</b>	Uso de tics em conjunto com a metodologia de aprendizagem baseada em problemas no ensino de algoritmos: um estudo de caso aplicado a um curso técnico do ifmg	Almeida et al	<b>2021</b>

<b>26</b>	O uso de softwares educativos na aprendizagem de lógica de programação para crianças do ensino fundamental I	Moreno et al	<b>2022</b>
<b>27</b>	Uso de jogos educacionais no ensino transdisciplinar de temas ambientais: estudo de caso com a plataforma scratch	Aguiar, Edson Rodrigues de	<b>2022</b>
<b>28</b>	Ambiente de simulação STEM para o ensino de robótica e programação: Um estudo de caso	Trindade et al.	<b>2022</b>
<b>29</b>	Ensino remoto de programação em Java para estudantes do ensino técnico utilizando tablets: Um relato de experiência	Matias et al.	<b>2023</b>
<b>30</b>	Robótica educacional e habilidades do século XXI: Um estudo de caso com estudantes do ensino médio	Moraes et al.	<b>2023</b>

Fonte: Autores (2023).

**3.2 QUESTÃO DE PESQUISA: QUAL É A METODOLOGIA DE ENSINO MAIS UTILIZADA PARA ENSINAR PROGRAMAÇÃO LEVANDO-SE EM CONSIDERAÇÃO A MODALIDADE DE ENSINO PRESENCIAL?**



Essa questão de pesquisa pretende identificar quais foram as linguagens de programação, jogos, materiais, algoritmos, que ajudaram a desenvolver as habilidades do pensamento computacional dos alunos.

Como ajuda para responder essa pergunta, realizamos alguns questionamentos individuais de pesquisa nos artigos selecionados.

### 3.2.1 QUESTIONAMENTO 1: QUAIS FORAM OS MÉTODOS UTILIZADOS NAS PUBLICAÇÕES?

Para classificá-los os separamos em 5 categorias: linguagens de programação, robótica educacional, jogos, desenvolvimento de jogos e computação desplugada.

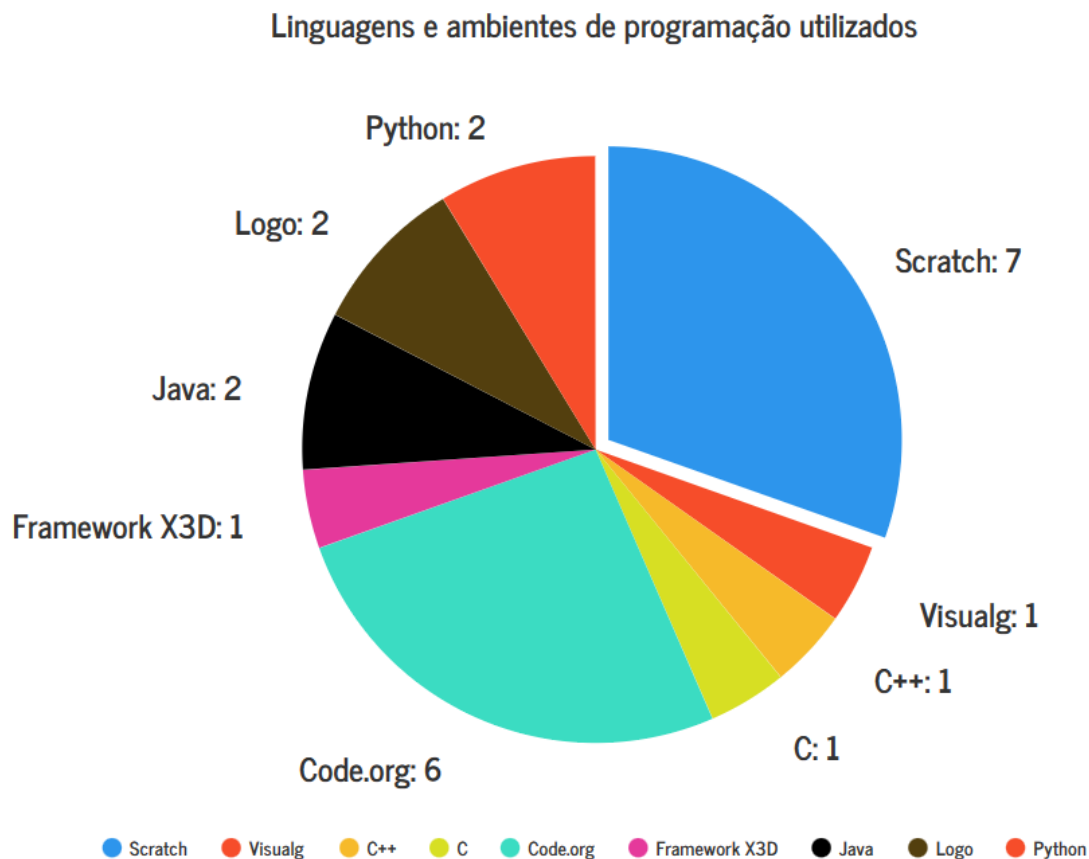
Os trabalhos foram classificados em cada uma das categorias seguindo o modelo a seguir:

([1] é indicado pela coluna ID do Quadro 2, no caso: [1] seria equivalente ao ID do primeiro artigo do quadro).

- Linguagem de programação: Dezenove artigos foram classificados nesta categoria. Os trabalhos selecionados [2], [5] e [27] utilizam a linguagem de programação visual Scratch, o trabalho [15] usou a linguagem Scratch e do ambiente Stencyl para o ensino. O trabalho [17] usou a linguagem Scratch além do Material da Computer Science without a computer e do Code.org. O trabalho [21] fez o uso da linguagem Scratch e do Code.org. O trabalho [26] usou a linguagem Scratch junto com o Google Doodle. O trabalho [1] usou a linguagem Logo junto com o brinquedo Lego Mindstorms, o trabalho [18] usou a linguagem Logo junto ao método Code.org. O trabalho [4] usou a linguagem C junto com a plataforma Facebook. O trabalho [11] usou a linguagem C++ junto com o Code.org. O trabalho [16] usou o Visualg e a ferramenta Construct 2. O trabalho [19] usou a linguagem Python junto com iVProg e pacote VPL (Virtual Programming Lab). O trabalho [24] usou a linguagem Python. Os trabalhos [25] e [29] utilizaram a linguagem Java. O trabalho [23] utilizou o MIT AppInventor junto com Code.org. O trabalho [22] utilizou o HTML, junto ao Framework X3D e o TAPVI3D. O trabalho [12] utilizou o Code.org para o ensino.

- Robótica educacional: Os trabalhos [6] e [9] envolveram o uso da robótica educacional. Um trabalho envolveu o uso de kit de robótica, o artigo [1] que trabalhou com o kit Lego Mindstorms. O trabalho [28] utilizou o simulador CoderZ para ensinar robótica educacional e programação. O trabalho [30] utilizou a ferramenta mBlock para ensinar robótica educacional.
- Jogos: O artigo [8] fez o uso do jogo Robocode, o [10] utilizou o jogo Robotizen e o [14] utilizou o jogo CodeCombat.
- Desenvolvimento de jogos: O trabalho [3] utilizou a ferramenta Squeak Etoys. O trabalho [7] utilizou a ferramenta GameMaker.
- Computação desplugada: A computação desplugada foi utilizada no artigo [13] e no artigo [20] junto ao code.org.

Gráfico 1– Linguagens e ambientes de programação utilizados



Fonte: Autores (2023).

### 3.2.2 QUESTIONAMENTO 2: QUAL FOI A LINGUAGEM DE PROGRAMAÇÃO MAIS UTILIZADA NAS PUBLICAÇÕES?

A linguagem de programação Scratch foi a mais utilizada com sete usos no total. Observando o gráfico 1, a linguagem de programação Scratch se destaca como sendo a mais utilizada nos artigos selecionados. Ela esteve presente de forma individual em três trabalhos, além de ser usada junto a outras estratégias, como no caso dos trabalhos [15],[17],[21] e [26].

No gráfico 2 que segue estão apresentadas as distribuições das categorias entre os artigos selecionados nesta análise. A partir da análise observamos o uso de linguagens de programação ocupando 61.3% por cento das estratégias utilizadas, seguido de 16.1% por cento da Robótica educacional. A categoria de Jogos foi a estratégia utilizada por 9.7% dos trabalhos.

Aproximadamente 6.5% utilizaram a estratégia de ensinar programação a partir do desenvolvimento de jogos. O uso da Computação desplugada foi apresentado em 6.5% dos trabalhos.

Gráfico 2– Divisão dos artigos por categorias



Fonte: Autores (2023).

### 3.2.3 QUESTIONAMENTO 3: QUAIS FORAM OS CONTEÚDOS ENSINADOS NO CONTEXTO DAS PUBLICAÇÕES?

Como resposta dessa questão criamos categorias com os tipos de conteúdo encontrados nos artigos selecionados que serão mostrados a seguir no Quadro 3. Utilizando uma forma de classificação, onde os conteúdos que abordam o mesmo assunto, indiferente de seus métodos, são expostos juntos.

Onde a categoria “Introdução a uma linguagem de programação e conteúdo de disciplinas escolares” utilizam da programação em conjunto de outras matérias. Como no

trabalho [5], onde usaram a ferramenta Scratch nas disciplinas de Literatura, levando os alunos a criarem e contarem histórias.

Na categoria “Ensino de conceitos de programação e prática com jogos” aborda os conceitos da programação, mostrando assim o funcionamento da lógica de programação e leva os alunos a exercitarem por meio de jogos educativos. Assim como o trabalho [14] onde após o ensino dos conceitos de lógica de programação, os alunos foram apresentados a ferramenta CodeCombat, onde via linhas de comando, são realizados os movimentos do personagem no jogo.

A “Introdução a lógica de programação e montagens de robôs” aborda o ensino da lógica de programação e utiliza a montagem de robôs para serem utilizados como ferramentas de teste. Os “Conceitos de algoritmos e estruturas de uma linguagem de programação” usam o ensino dos conceitos de lógica de programação para serem aplicados em uma linguagem específica, assim como no trabalho [18], onde é utilizado a linguagem Logo para o ensino da programação.

Os “Conceitos de desenvolvimento de jogos” ensinam através da criação de jogos os conceitos de computação, no trabalho [7] utilizam da ferramenta GameMaking que possibilita a criação de um jogo sem a necessidade de inserir uma linha de código, tornando a compreensão dos conceitos básicos mais fácil. A “Introdução a computação através de jogos” utiliza diretamente jogos que ensinam a programar para o ensino da lógica de programação. E por fim a “Avaliação das ferramentas utilizadas no ensino” abordam as ferramentas mais utilizadas para o ensino de programação e faz uma comparação entre elas.

Quadro 3– Conteúdos ensinados no contexto dos artigos

Conteúdo ensinado	ID
Ensino de conceitos de computação e prática com jogos.	[12], [13], [14], [17].
Introdução a lógica de programação e montagens de robôs.	[1], [6], [9], [11], [25],[28],[30].

Conceitos de algoritmos e estruturas de uma linguagem de programação.	[2], [8], [18], [19], [23], [24],[26],[27],[29].
Conceitos de desenvolvimento de jogos.	[3], [7], [16], [21].
Introdução a uma linguagem de programação e conteúdo de disciplinas escolares.	[4], [5], [20], [22].
Introdução a computação através de jogos.	[10].
Avaliação das ferramentas utilizadas no ensino.	[15].

Fonte: Autores (2023).

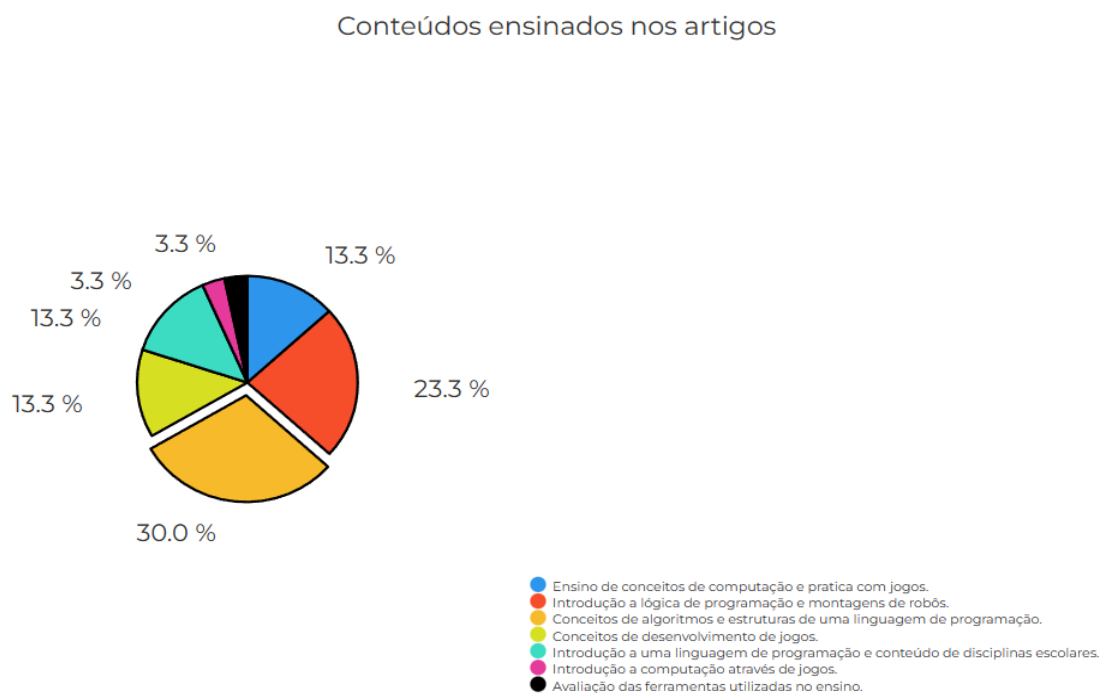
No gráfico 3 estão apresentadas as distribuições dos conceitos que foram ensinados entre os artigos selecionados.

A partir da análise observamos o uso de Conceitos de algoritmos e estruturas de uma linguagem de programação ocupando 30% por cento das estratégias utilizadas, seguido de Introdução a lógica de programação e montagens de robôs com 23.3% por cento de uso.

Os conceitos: O Ensino de conceitos de computação e prática com jogos, Conceitos de desenvolvimento de jogos e a Introdução a uma linguagem de programação e conteúdo de disciplinas escolares, ambas contaram com 13.3% por cento dos conceitos utilizados.

Introdução a computação através de jogos e Avaliação das ferramentas utilizadas no ensino utilizaram conceitos mais específicos e ficaram com 3.3% de uso.

Gráfico 3 – Conteúdos ensinados



Fonte: Autores (2023).

#### 4. PROPOSTA DE UMA METODOLOGIA PARA UMA OFICINA MODELO PARA O ENSINO DA PROGRAMAÇÃO PARA CRIANÇAS

A metodologia propõe um conjunto de práticas reunidas visando fornecer ao professor uma aula didática para a aplicação de conceitos de programação em um ambiente inovador e lúdico.

A estratégia é baseada na aprendizagem com foco no uso de dinâmicas para gerar a interação dos alunos acompanhada de atividades propostas ao decorrer das aulas. Propõe-se nessa metodologia que os professores apresentem um passo a passo de como usar os comandos do Scratch no início de cada tarefa e principalmente auxiliar quando os alunos desenvolverem suas próprias soluções para os exercícios.

Nesta proposta, as crianças serão introduzidas ao Scratch e aprenderão a criar projetos, jogos e animações usando blocos de programação. O objetivo é despertar o interesse delas pela programação, proporcionando um ambiente de aprendizado prático e estimulante.

## 4.1 PLANO DE AULA

O plano de aula proposto é dividido em 6 aulas. A divisão das aulas consiste em uma aplicação progressiva de conceitos teóricos de programação, dinâmicas para exemplificar os conceitos abordados, além de inserir desafios crescentes para manter os alunos motivados.

O objetivo da sequência é trabalhar as habilidades criando pré-requisitos para que o aluno progrida nas tarefas, além de estimular o interesse no uso da tecnologia.

As aulas estão divididas da seguinte forma:

Figura 4 – Planos de aula

Aula	Tema da Aula	Metodologia	Conceito	Prática
1	Algoritmo	Receita da imagem	Explicação do termo Algoritmo	Construir um desenho baseado nos comandos recebidos.
2	Uso dos algoritmos no dia a dia	Descreva o algoritmo	Explicar como os conceitos de algoritmos estão presentes no nosso cotidiano.	Realizar a descrição de atividades do dia a dia em algoritmos.
3	Conceito de Variáveis e Introdução a Ferramenta Scratch	Caixa Variável	Explicar os conceitos de variáveis e Introduzir a ferramenta Scratch aos alunos.	Exemplificar os conceitos de variáveis utilizando a dinâmica.
4	Conceito de condicionais	Brincadeira Morto vivo	Explicar o conceito e as funcionalidades das condicionais	Exemplificar os conceitos de condicionais utilizando a dinâmica e introduzir atividades dentro da ferramenta Scratch
5	Projeto do zero em Scratch e conceito de repetições	Criar primeiro programa e Jogo da memória	Ensinar a criação do primeiro jogo e explicar como funcionam as repetições.	Propor e auxiliar os alunos na criação do primeiro programa em Scratch. Utilizar o jogo da memória para exemplificar o conceito de repetições.
6	Revisão e criação de histórias	Criar história	Revisar o conteúdo ensinado	Revisão do conteúdo e criação de histórias pelos alunos.

Fonte: Autores (2023).



**Aula 1: Explicar o conceito de algoritmo:** A primeira aula pretende ensinar o conceito de algoritmo, para isso o professor fará uma apresentação para demonstrar o conceito de algoritmo e o seu funcionamento. Para exemplificar o conceito após a apresentação será realizada uma dinâmica com a turma com o nome de “Receita da imagem”.

Os alunos serão separados em duplas onde um fará o papel do computador e o outro enviará os comandos, o aluno responsável pelos comandos deverá escolher um desenho e dar o algoritmo para o colega desenhar o mesmo. O aluno “computador” não terá acesso prévio à imagem e deve seguir somente o passo a passo (receita) para desenhar o objeto.

Por meio dessas instruções, o aluno, que faz o papel do computador, poderá reproduzir os passos, ou seja, um “programa”, após terminar o desenho podem ser trocados os papéis para que os dois alunos tenham as mesmas experiências.

**Aula 2: Uso dos algoritmos no dia a dia:** Essa aula pretende mostrar como utilizamos algoritmos no nosso dia a dia quase o tempo todo, citando exemplos do nosso cotidiano. Para isso, o professor poderá fazer uma apresentação sobre o assunto e demonstrar um exemplo de algoritmo do cotidiano para maior entendimento dos alunos.

A partir disso será realizada uma dinâmica com o nome de “Descreva o algoritmo” utilizando exemplos de coisas que acontecem no nosso dia, com isso o professor solicitará aos alunos que descrevam tal função pensando qual seria o algoritmo para resolver aquela tarefa. Com isso seria gerado o interesse e a contribuição da turma com a dinâmica.

Um exemplo de algoritmo para exemplificar a atividade seria: realizar a troca de uma lâmpada. A seguir será demonstrado como seriam inseridos os algoritmos para realizar tal tarefa.

Figura 5 – Exemplo de algoritmo genérico para troca de uma lâmpada

## Exemplo: Algoritmo genérico para troca de uma lâmpada

- 1- Remova a lâmpada queimada
- 2- Coloque uma lâmpada nova

Fonte: Autores (2023).

Figura 6 – Exemplo de algoritmo detalhado para a troca de uma lâmpada

### Detalhando os passos

- 1.1 - Coloque uma escada embaixo da lâmpada queimada
- 1.2 - Suba na escada até alcançar a lâmpada queimada
- 1.3 - Gire a lâmpada no sentido anti-horário até que ela se solte
- 2.1 - Escolha uma lâmpada nova da mesma potência da queimada
- 2.2 - Posicione a lâmpada nova no soquete
- 2.3 - Gire a lâmpada no sentido horário até que ela se firme
- 2.4 - Desça da escada
- 2.5 - Guarde a escada

Fonte: Autores (2023).

Figura 7 – Exemplo de algoritmo detalhado e com a ordem correta para trocar uma lâmpada

### Ordenando os passos

- 1 - Coloque uma escada embaixo da lâmpada queimada
- 2 - Escolha uma lâmpada nova da mesma potência da queimada
- 3 - Suba na escada até alcançar a lâmpada queimada
- 4 - Gire a lâmpada no sentido anti-horário até que ela se solte
- 5 - Posicione a lâmpada nova no soquete
- 6 - Gire a lâmpada no sentido horário até que ela se firme
- 7 - Desça da escada
- 8 - Guarde a escada

Fonte: Autores (2023).

Na Figura 5 é demonstrado o algoritmo genérico de como seria feita a troca de uma lâmpada, considerando que essa seria a resposta inicial dos alunos, o professor poderia questioná-los utilizando os passos que estão sendo descritos na Figura 6, fazendo com que eles repensem o passo a passo da troca da lâmpada. Após toda a criação do método da Figura 6 ainda assim juntamente a eles realizar a ordem dos passos corretos para a troca como mostra a Figura 7. Esse seria um exemplo de atividade, com isso o professor fica livre para questionar outras tarefas para a turma descrever.

**Aula 3: Conceito de variáveis:** Essa aula visa explicar o conceito de variáveis para os alunos. Para isso, o professor fará uma apresentação exemplificando o tema a fim de explicar a sua funcionalidade e objetivo. Após isso, será feita uma dinâmica utilizando o conceito de variáveis com o nome de “Caixa variável”, a dinâmica consiste no professor ou os alunos escreverem apenas o seu primeiro nome no quadro, após isso todos os alunos da turma escrevem seus sobrenomes em um pedaço de papel, dobram e depositam dentro da “Caixa variável” que ficará com o professor. Após todos os alunos depositarem os nomes, um a um tirará um papel da “Caixa variável” e tentará adivinhar a quem pertence aquele sobrenome. Com isso, o

conceito de que uma variável pode receber valores e que também esses valores podem ser alterados ficará exemplificado para os alunos.

Após a aplicação dessa dinâmica, o professor dará início aos conceitos básicos da ferramenta Scratch. Um exemplo de abordagem para a introdução da ferramenta Scratch poderá ser realizada explicando passo a passo como funciona a sua interface e suas principais funções. Um exemplo de metodologia para ensinar as funcionalidades da ferramenta:

A interface da ferramenta Scratch é dividida em três partes principais, sendo elas:

**Área de blocos:** É onde você encontra os blocos de programação que serão usados para criar os scripts. Esses blocos contam com diferentes cores e formas, representando diferentes ações e eventos.

**Palco:** É onde você encaixa os blocos para criar comandos do seu programa. Podendo adicionar personagens, objetos e cenários.

**Área de Scripts:** Onde se encaixam os blocos para criar os comandos do seu programa. Os blocos podem ser arrastados e encaixados uns aos outros, formando uma sequência de ações.

Com a criação de scripts, utilizando os blocos na área de scripts, podemos começar a criar comandos para os personagens interagirem com eventos. Por exemplo “Quando a bandeira verde for clicada”, para iniciar um script quando a bandeira for clicada. Você deve arrastar e encaixar os blocos de ação, como “Mover 10 passos” ou “Repetir 5 vezes” para criar a lógica do programa.

O próximo passo após ensinar o básico da interface seria mostrar como funciona a criação dos projetos, adicionando cenários e personagens para dar início ao primeiro contato dos alunos com a ferramenta.

Após isso o professor poderá introduzir a ferramenta Scratch no ensino para os alunos começarem a tentar utilizar de forma independente os comandos pré-definidos pela ferramenta em exercícios básicos propostos, esses exercícios têm o objetivo de ajudar os alunos a se familiarizar com a ferramenta.

**Aula 4: Conceito de condicionais:** Essa aula visa explicar o conceito e as funcionalidades das condicionais, para isso o professor fará uma apresentação sobre o tema para explicar a sua funcionalidade. Após a apresentação teórica do assunto, será realizada a dinâmica da

brincadeira infantil “Morto vivo” para exemplificar que as estruturas condicionais nada mais são do que regras que devem ser cumpridas pelo programa, assim como na brincadeira que após um comando dado eles terão que realizar uma função no jogo, caso errem o comando pedido resulta na sua saída do jogo, para deixar mais claro o professor deverá criar mais regras no jogo fazendo com que deixe as condicionais evidentes na dinâmica para os alunos.

Após a finalização da dinâmica, poderão ser utilizados exercícios envolvendo condicionais na ferramenta Scratch para que os alunos pratiquem os conceitos que aprenderam na aula.

**Aula 5: Projeto do zero em Scratch e conceito de repetições:** Um dos objetivos dessa aula é que cada aluno construa a sua própria versão do jogo “Adivinhe o número” baseado nos passos que serão dados pelo professor.

O objetivo é utilizar os conceitos básicos de programação ensinados até o momento de uma forma divertida e interativa. O jogo é um excelente ponto de partida, por permitir que os alunos pratiquem a lógica de programação enquanto se divertem tentando adivinhar um número aleatório.

Durante o processo, eles terão a oportunidade de explorar a interface do Scratch, arrastar e soltar blocos de programação e desenvolver sua criatividade ao personalizar o jogo. Além disso, incentivará a colaboração e a troca de ideias, estimulando-os a compartilharem suas criações e a testarem os jogos uns dos outros. Isso promoverá a interação e o aprendizado em equipe, aumentando assim o interesse deles pela programação.

Nesta atividade, os alunos serão incentivados a pensar logicamente, resolver problemas e testar suas habilidades de adivinhação. O professor poderá incentivar as crianças a explorarem e personalizarem o jogo com suas próprias ideias e recursos adicionais, como efeitos sonoros, animações e elementos visuais. Em seguida será mostrado o roteiro para a criação do jogo.

### **Adivinhe o número**

Passos:

1. Criar o cenário:

- Crie um novo projeto no Scratch e escolha um cenário de fundo adequado para o jogo de adivinhação.
- Adicione um personagem principal.

2. Gerar um número aleatório:

- Crie uma variável chamada “número\_secreto”.
- Use um bloco de programação para atribuir um número aleatório à variável “número\_secreto”. Por exemplo, você pode usar o bloco “definir [número\_secreto] como (escolher [1] até [10])”.

3. Criar a lógica do jogo:

- Adicione um bloco de evento “quando a bandeira verde for clicada” para iniciar o jogo.
- Peça ao jogador para digitar um número em um balão de fala usando o bloco “pergunte [Digite um número] e espere”.
- Crie uma condicional para verificar se o número digitado é igual ao “número secreto”.
- Se o número estiver correto, mostre uma mensagem de parabéns.
- Se o número estiver errado, mostre uma mensagem de erro e peça ao jogador para tentar novamente.

4. Adicionar opções de reiniciar o jogo:

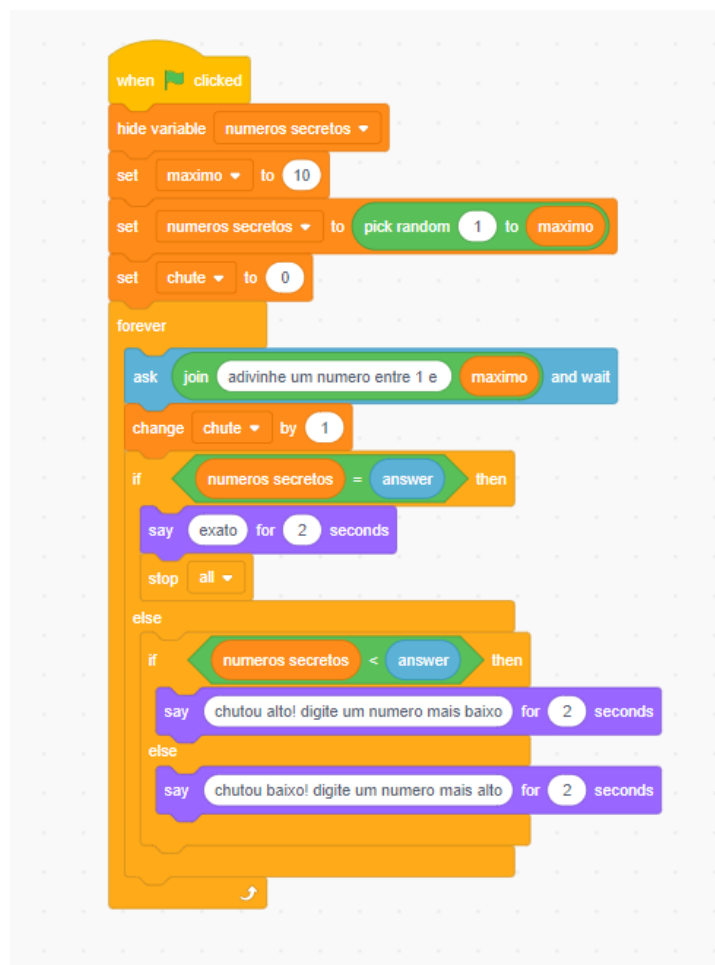
- Crie um bloco de evento “quando a tecla [espaço] for pressionada” para reiniciar o jogo.
- Defina a variável “número\_secreto” como um novo número aleatório.
- Limpe o balão de fala e permita que o jogador faça uma nova tentativa.

5. Testar e refinar o jogo:

6. Teste o jogo e identifique possíveis melhorias.

- Adicione recursos adicionais, como sons, animações ou pontuações, para tornar o jogo mais interessante.

Figura 8 – Exemplo da implementação do jogo.



Fonte: Site oficial Scratch (2018)

Após a finalização da dinâmica do jogo “Adivinhe o número” o professor abordará o conceito de repetições, para isso será feita uma apresentação do assunto, sua função e finalidades. Para exemplificar o conceito, será utilizada a dinâmica com jogos de memória, para mostrar que enquanto eles não conseguirem encontrar as figuras iguais, terão que repetir o processo de desvirar as cartas no jogo. Depois da dinâmica, serão utilizados em exercícios propostos no ambiente Scratch os comandos “Repita X vezes” e “Repita Sempre” para o uso do conceito abordado na aula.

**Aula 6: Revisão:** Nessa aula serão revisados os conteúdos aplicados na oficina, a fim de relembrar todo o ensinamento tratado nas aulas. Após isso o professor solicitará que cada aluno

crie a sua própria história com o Scratch para ser apresentada a turma, o assunto para a história pode ser livre fazendo com que os alunos trabalhem a sua criatividade na criação da sua história, o professor deverá auxiliá-los quando tiverem problemas. Após todos terminarem a sua história, farão uma apresentação delas para a turma.

## **4.2 AVALIAÇÃO DO OBJETIVO DA OFICINA**

O principal objetivo da oficina é aumentar o interesse dos alunos pela área da programação e tecnologia. Para isso, antes do início da primeira aula será aplicado um questionário (Apêndice A) com os alunos a fim de verificar o conhecimento e os interesses deles em computação e tecnologia em geral. Graças a isso, pretende-se saber se houve mais interesse por parte deles após a finalização da oficina.

Ao fim da última aula propõe-se a aplicação de uma lista de atividades objetivas sobre os temas abordados na oficina a fim de avaliar o aprendizado dos alunos. Além disso, será aplicado outro questionário (Apêndice B) a fim de verificar se os alunos percebem que seus conhecimentos relacionados à informática e tecnologia foram afetados e se o seu interesse pela área aumentou após participar da oficina.

Como forma de avaliação da metodologia de ensino proposta dentro da oficina, ela foi apresentada aos professores, que por unanimidade a avaliaram como uma boa estratégia que pode ser implementada para avaliar seus resultados.

## **5 CONSIDERAÇÕES FINAIS**

Este trabalho propôs uma metodologia de oficina de ensino de programação para crianças entre 7 e 14 anos em escolas, a partir de um estudo exploratório sobre metodologias que estão sendo utilizadas no ensino de programação e os conteúdos que estão sendo apresentados para os alunos nas iniciativas de ensino em estudos de caso. A coleta dos artigos foi realizada por meio do motor de busca “Google Acadêmico”.

Foram selecionados 30 artigos, seguindo os seguintes critérios de inclusão:

- Artigos que relatam experiência em estudos de caso realizados no Brasil;



- Artigos de 2012 até 2023 para análise de metodologia de ensino no passar dos anos;
- Artigos escritos na língua portuguesa;
- Artigos que relatam uso de ferramentas para o ensino de programação.

Percebeu-se que, dos 30 artigos selecionados, 7 fazem uso da ferramenta Scratch no ensino, devido a sua interface visual e o uso da programação em blocos para construir projetos fazendo com que ela se torne acessível e atraente para as crianças, levando os alunos a aprenderem de forma mais fácil, além de sua biblioteca de recursos e comunidade online ativa, que oferece suporte adicional para as crianças.

Com relação aos conteúdos, destacamos como o método de ensino mais utilizado representando 30% dos artigos, o tema “Conceitos de algoritmos e estruturas de uma linguagem de programação”, que teve destaque devido ao fato de ser uma forma simples de introduzir os conceitos básicos antes do primeiro contato com a programação.

Baseado nisso, apresentamos uma proposta de metodologia de uma oficina modelo para o ensino de programação para crianças. Essa proposta se baseia em uma aplicação progressiva de conceitos teóricos de programação, dinâmicas para exemplificar os conceitos abordados, além de inserir desafios crescentes para manter os alunos motivados.

Acreditamos que o ensino de programação para crianças deve ser ampliado e fortalecido. A importância de promover habilidades de programação desde a infância torna-se cada vez mais evidente em um mundo cada vez mais digital. Além disso, é fundamental investir recursos e formação adequada para os educadores, a fim de garantir a qualidade e eficácia do ensino de programação.

Esperamos que este trabalho contribua para a disseminação do ensino de programação para as crianças, inspirando outros educadores e instituições de ensino a explorarem esse campo que já se provou muito promissor. A programação, aliada à criatividade e ao pensamento crítico, abrirá novas possibilidades e oportunidades para as crianças, preparando-as para os desafios do século XXI.

Como trabalho futuro, sugere-se um estudo para a aplicação da oficina em escolas públicas para análise do método no ambiente de aprendizagem junto a professores e alunos. A análise desses resultados terá como objetivo verificar a eficácia da abordagem proposta e quais são as possíveis melhorias que podem ser aplicadas.

## REFERÊNCIAS

ARIMOTO, Maurício; OLIVEIRA, Weldrey. **Dificuldades no Processo de Aprendizagem de Programação de Computadores: um Survey com Estudantes de Cursos da Área de Computação**. In: WORKSHOP SOBRE EDUCAÇÃO EM COMPUTAÇÃO (WEI), 27., 2019, Belém. Porto Alegre: Sociedade Brasileira de Computação, 2019. p. 244-254.

BEECHER, K. **Computational thinking a beginner's guide to problem-solving and programming**. Swindon: BCS, 2017

**Biblioteca Digital de Monografias da UFPA**. Disponível em: <https://bdm.ufpa.br:8443/>. Acessado em: 31 de março de 2023.

**Biblioteca Digital da SBC**. Disponível em: <https://sol.sbc.org.br/index.php/anais>. Acesso em 31 de março de 2023.

CODE.ORG. **Learn Code**. Disponível em: <<http://learn.code.org/hoc/1>>. Acesso em: 22 de novembro de 2022

CodeCombat. **CodeCombat** (fr wiki.wiki). Disponível em:< <https://pt.frwiki.wiki/wiki/CodeCombat>>. Acesso em 24 de maio de 2023.

CodeCombat - **Coding games to learn Python and JavaScript | CodeCombat (CodeCombat site oficial)**. Disponível em:< <https://codecombat.com/>>. Acesso em 22 de novembro de 2022.

CODECOMBAT. **Foto recortada de uma cena do jogo CodeCombat**. 2017. Figura 3. Disponível em: <https://discourse.codecombat.com/t/user-interface-weird-ui-objects-obscuring-map/12809> Acesso em: 09 de maio de 2023.

CORE. Disponível em: <https://core.ac.uk/>. Acessado em: 31 de março de 2023.

**FERREIRA, Pensamento Computacional e Programação Scratch: uma revisão de literatura do SBIE.** Disponível em: <https://sol.sbc.org.br/index.php/encompif/article/view/15942/15783> Acessado em: 11 de abril de 2023.

**GIL, A. C. Métodos e técnicas de pesquisa social.** 5.ed. São Paulo: Atlas, 1999.

**GOMES, P. Codecademy ensina programação de graça.** Disponível em <<https://porvir.org/codecademy-ensina-programacao-de-graca-pelo-mundo/>>. Acesso em: 22 de novembro de 2022.

**GOMES, Marcos César Pires. Os benefícios do ensino de linguagem de programação no currículo regular,** 2015.

**HARDNETT, Gaming for Middle School Students: Building Virtual Worlds.** Disponível em: [https://static.aminer.org/pdf/PDF/000/591/506/cybertech\\_itest\\_an\\_nsf\\_funded\\_initiative\\_to\\_a\\_ttract\\_and\\_prepare.pdf](https://static.aminer.org/pdf/PDF/000/591/506/cybertech_itest_an_nsf_funded_initiative_to_a_ttract_and_prepare.pdf) Acessado em: 11 de abril de 2023.

**KAFAL, Y. B.; BURKE, Q. Computer programming goes back to school. PhiDelta Kappan, SAGE Publications Sage CA: Los Angeles, CA, v. 95, n. 1, p.61–65, 2013.**

**KELLEHER, C. and Pausch, R. (2005). Lowering the barriers to programming: A taxonomy of programming environments and languages for novice programmers. ACM Computing Surveys (CSUR), 37(2):83–137. Publisher: ACM New York, NY, USA.**

**OLIVEIRA, Ensino de lógica de programação no ensino fundamental utilizando o Scratch: um relato de experiência.** Disponível em: <https://sol.sbc.org.br/index.php/wei/article/view/10978/10848> Acessado em: 11 de abril de 2023.

ORO, Neuza et al. **Olimpíada de Programação de Computadores para Estudantes do Ensino Fundamental: A interdisciplinaridade por meio do Software Scratch**. In: Anais do WoPENTAGRAM. Foto encontrada através da busca pela logomarca da Code

PENTAGRAM. Foto encontrada através da busca pela logomarca da CodeCademy. Figura 5. Disponível em: <https://www.pentagram.com/work/codecademy-1> Acesso em: 09 de maio de 2023.

PRADO, M. E. B. B. **LOGO – Linguagem de Programação e as Implicações Pedagógicas**. Campinas: UNICAMP, 2000

PNG WING. Foto encontrada através da busca pela logomarca da Code.org. Figura 4. Disponível em: <https://www.pngwing.com/pt/search?q=Code.org> Acesso em: 09 de maio de 2023.

**Revista Brasileira de Informática na Educação**. Disponível em: <http://www.br-ie.org/pub/index.php/rbie>. Acesso em: 31 de março de 2023.

**Revista Brasileira de Informática na Educação**. Disponível em: <http://ojs.sector3.com.br/index.php/rbie/index>. Acesso em: 31 de março de 2023.

**Repositório Institucional da Ufal**. Disponível em: <https://www.repositorio.ufal.br/jspui/community-list>. Acesso em: 31 de março de 2023.

**Repositório Institucional da Universidade Tecnológica Federal do Paraná (RIUT)**. Disponível em: <http://riut.utfpr.edu.br/jspui/>. Acessado em: 31 de março de 2023.

**Repositório Institucional UnB**. Disponível em: <https://repositorio.unb.br/>. Acesso em: 31 de março de 2023.

SANTOS, R. P.; Costa, H. A. X. (2006). **Análise de metodologias e ambientes de ensino para algoritmos, estrutura de dados e programação aos iniciantes em Computação e Informática.** INFOCOMP Journal of Computer Science, 5(1):41–50.

SCRATCH. **ABOUT Scratch (Scratch Documentation Site).** Disponível em: <<https://scratch.mit.edu/about>>. Acesso em: 22 de novembro de 2022.

SCRATCH. PROJECTS (Scratch. adivinhe o número by mcpmyngrid) Disponível em: < <https://scratch.mit.edu/projects/230259854/> >

SCRATCH. The Guessing Game. Disponível em: < [https://scratched.gse.harvard.edu/sites/default/files/scratch\\_exercise6.pdf](https://scratched.gse.harvard.edu/sites/default/files/scratch_exercise6.pdf) >

SEER.UFRGS. **Foto recortada de um exemplo de exercício de desenhar flores utilizando Superlogo.** 2018. Figura 2. Disponível em: <https://seer.ufrgs.br/index.php/renote/article/view/85933/49313> Acesso em: 09 de maio de 2023.

SELBY, C.; woollard, J. **Computational thinking: the developing definition**, 2013.

SERRANO, F. **Geração Geek. Revista Exame Informática.** São Paulo: Editora Abril, 2014.

**Simpósio Brasileiro de Informática na Educação.** Disponível em: <http://www.br-ie.org/pub/index.php/sbie/index>. Acesso em: 31 de março de 2023.

WAGNER GASPAR. **Foto recortada de um exercício de cálculo de idade utilizando Scratch.** 2021. Figura 1. Disponível em: <https://wagnergaspar.com/exercicio-1-calculando-a-idade-do-usuario-no-scratch/> Acesso em: 09 de maio de 2023.

WING, J. **Computational thinking. Communications of the acm**, v. 49, n. 3, 2006, p. 33-35.

ZANETTI, Humberto Augusto Piovesana; BORGES, Marcos Augusto Francisco; RICARTE, Ivan Luiz Marques. **Pensamento Computacional no Ensino de Programação: uma revisão sistemática da literatura. In: Congresso Brasileiro de Informática na Educação, 5, 2016, Uberlândia, Minas Gerais. Anais [...]. Disponível em: <https://www.brie.org/pub/index.php/sbie/article/view/6677/456> 6. Acesso em: 31 de março de 2023. <https://doi.org/10.5753/cbie.sbie.2016.21>**

## **APÊNDICE A – QUESTIONÁRIO 1 – INÍCIO DA OFICINA**

### Questionário 1 - Início da oficina

#### Dados do aluno

Nome: \_\_\_\_\_ Idade: \_\_\_\_\_ Sexo: \_\_\_\_\_  
Turma: \_\_\_\_\_ E-mail: \_\_\_\_\_

#### 1. Você gosta de Tecnologia?

( ) Muito ( ) Mais ou menos ( ) Pouco ( ) Não gosto

Por quê? \_\_\_\_\_  
\_\_\_\_\_

#### 2. Responda com Sim ou Não:

Você já fez algum curso de informática? Sim ( ) Não ( )  
Você já fez algum curso relacionado à programação? Sim ( ) Não ( )

#### 3. Em casa, você utiliza a tecnologia para:

( ) Estudar ( ) Jogar ( ) Redes sociais ( ) Fazer pesquisa

( ) Outras atividades. Quais? \_\_\_\_\_

#### 4. Você tem interesse por Computação? ( ) Sim ( ) Não

Por quê? \_\_\_\_\_  
\_\_\_\_\_

#### 5. Você gosta do laboratório de informática? ( ) Sim ( ) Não

Por quê? \_\_\_\_\_  
\_\_\_\_\_

### APÊNDICE B – QUESTIONÁRIO 2 – FINAL DA OFICINA

## Questionário 2 – Final da oficina

### Dados do aluno

Nome: \_\_\_\_\_ Idade: \_\_\_\_\_ Sexo: \_\_\_\_\_

Turma: \_\_\_\_\_ E-mail: \_\_\_\_\_

### 1. Responda com Sim ou Não:

Você gostaria de fazer algum curso de informática?      Sim ( )      Não ( )

Você gostaria de fazer algum curso de programação?      Sim ( )      Não ( )

### 2. Após a sua participação na oficina você teve mais interesse por Computação?

( ) Sim      ( ) Não.

Por  
quê? \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

### 3. Você tem interesse em continuar a fazer aulas de programação Scratch?

( ) Sim      ( ) Não

Por  
quê? \_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

### 4. Você fez alguma pesquisa sobre Computação/Scratch/Programação durante a aplicação da oficina?

( ) Sim      ( ) Não

Qual? \_\_\_\_\_

### 5. Você tem alguma sugestão para a oficina?

( ) Sim      ( ) Não