

**INSTITUTO DE HACKERS**  
**OFICINA DE PROGRAMAÇÃO BASEADA EM BLOCOS**



**Leandro Delgado de Souza**  
**Elisângela Valevein Rodrigues**

**PROFEPT**

MESTRADO PROFISSIONAL EM  
EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA

---

**INSTITUTO FEDERAL**  
**Paraná**

**Curitiba – PR**  
**2019**

Dados da Catalogação na Publicação  
Instituto Federal do Paraná  
Biblioteca do Campus Curitiba

S729i Souza, Leandro Delgado de  
Instituto de hackers: oficina de programação baseada em blocos / Leandro Delgado de Souza; orientadora, Elisângela Valevein Rodrigues. – Curitiba: Instituto Federal do Paraná, 2019. - 21 p. : il. color.

1. Pensamento computacional. 2. Educação profissional e tecnológica. 3. Aprendizagem situada I. Rodrigues, Elisângela Valevein. II. Título.

CDD 23. ed. - 006



LEANDRO DELGADO DE SOUZA

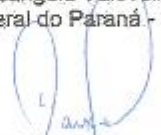
**INSTITUTO DE HACKERS OFICINA DE PROGRAMAÇÃO BASEADA EM BLOCOS**

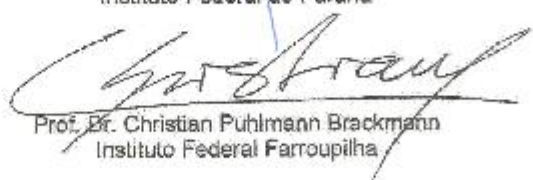
Produto Educacional apresentado ao Programa de Pós-Graduação em Educação Profissional e Tecnológica (ProfePT), do Instituto Federal do Paraná – Campus Curitiba, como requisito parcial para obtenção do título de Mestre em Educação Profissional e Tecnológica.

Aprovado e validado em 07 de outubro de 2019.

**COMISSÃO EXAMINADORA**

  
Prof.ª Dr.ª Elisângela Valévin Rodrigues  
Instituto Federal do Paraná - Orientador

  
Prof.ª Dr.ª Márcia Valéria Paixão  
Instituto Federal do Paraná

  
Prof. Dr. Christian Puhlmann Brackmann  
Instituto Federal Farroupilha

## SUMÁRIO

<b>INTRODUÇÃO</b> .....	6
<b>PILARES DO PENSAMENTO COMPUTACIONAL</b> .....	8
<b>PROGRAMAÇÃO POR BLOCOS</b> .....	9
<b>DESCRIÇÃO DA PROPOSTA DE ENSINO</b> .....	9
<b>Quadro 1. Sequência didática da aula 1- Estrutura Sequencial</b> .....	11
<b>Quadro 2. Sequência didática da aula 2</b> .....	12
<b>Quadro 3. Sequência didática da aula 3 e 4 – Estrutura de Decisão</b> .....	15
<b>Quadro 4. Sequência didática da aula 5 e 6 – Estrutura de Repetição</b> .....	18
<b>REQUISITOS PARA APLICAÇÃO DA OFICINA</b> .....	19
<b>RELAÇÃO ENTRE ATIVIDADES DE PILARES DO PENSAMENTO COMPUTACIONAL</b> .....	21
<b>CONSIDERAÇÕES FINAIS</b> .....	22
<b>REFERÊNCIAS</b> .....	23

## RESUMO

A educação profissional e tecnológica contemporânea tem suas bases na relação entre a educação e o mundo do trabalho, com objetivo de promover a concepção de estudantes autônomos, críticos, capazes de construir seu conhecimento, com base em atividades práticas, teóricas e interdisciplinares. A programação, por suas características, desperta a criatividade, o raciocínio lógico, a construção passo a passo de soluções de problemas reais, transportados para o mundo computacional. Tendo essa base e relação entre educação profissional e tecnológica com a programação de computadores, este produto educacional é uma proposta de ensino indicada a professores da área de computação que trabalhem ou desejam ensinar lógica de programação em suas aulas, ou ministrar cursos ou oficinas de programação. Esta proposta de ensino pode ser ministrada no formato de oficinas com duração de 20 horas aulas, para alunos de ensino médio técnico integrado. A linguagem base utilizada é a Blockly, para servir de base para outras aplicações utilizarem a linguagem de blocos como ferramenta para a construção de programas, animações, histórias e games. Os softwares utilizados são o Scratch e a Hora do Código. Espera-se que essa oficina sirva como base orientadora aos docentes da área que trabalhem com o ensino de programação, ou desejem trabalhar com ensino de programação em suas aulas, e que os estudantes tenham contato com o mundo da programação, desmistificando essa área tão abrangente e complexa, e desenvolvam o pensamento computacional, tão importante para os dias atuais.

Palavras-chave: Educação Profissional. Pensamento Computacional. Programação.

## INTRODUÇÃO

Este produto educacional trata-se de uma proposta de ensino de programação, utilizando lógica de blocos, ministrada na forma de oficina de curta duração. A construção da oficina teve base na aprendizagem situada, criada por Jean Lave. Essa prática de aprendizagem envolve o contexto, podendo ser o contexto vivido, presenciado ou proposto aos estudantes (LAVE, 2013). A mesma autora, em 2015, afirma que o aprendizado ocorre quando o estudante se apropria de conteúdos previamente conhecidos para resolver problemas, situações e criar soluções a partir destes.

Um exemplo é uma pessoa que está em um supermercado, e precisa escolher entre dois produtos semelhantes. Sabe-se, por meio de aulas de biologia, que excesso de sódio não faz bem para o corpo. Dessa forma, essa pessoa olha a tabela nutricional dos produtos e faz uma comparação. Em um produto A, existem 50mg de sódio em 100g de produto, já no produto B, existem 30mg de sódio em 50g de produto. Dessa forma, a pessoa pode fazer duas escolhas. Uma seria a que possui o menor valor declarado de sódio, a outra seria calcular a proporção de sódio na mesma quantidade de produto. Utilizando-se de conceitos matemáticos, a pessoa percebe que o produto A possui 50mg/100g, e o produto B possui 60mg/100g, dessa forma assim escolhendo, de forma correta, o produto A, pois é o que possui menor teor de sódio por igual porção.

Esse exemplo ilustra o aprendizado, pois este ocorreu no momento de solução de uma situação problema, dentro do contexto da pessoa, uma simples compra no supermercado, baseado em conhecimentos previamente adquiridos.

A proposta de ensino de programação utilizando-se lógica de blocos baseada na aprendizagem situada é uma tentativa de inovação do ensino de programação, principalmente quando tratam-se de cursos que não são provenientes de áreas tecnológicas, como técnico em informática, mas que possuem disciplinas de informática ou computação em sua grade curricular, corroborando para a prerrogativa de que o pensamento computacional é uma habilidade que deve estar disponível para todos.

Muitas pesquisas relatam a dificuldade de estudantes, que não são de áreas tecnológicas, em aprender programação. Quando se trata de estudantes do ensino médio, essa dificuldade pode ser ainda maior, pois para alguns, aprender programação não faz sentido, ou não percebem a possível aplicação ou não entendem o motivo de estarem aprendendo isso. A observação dessa realidade, motivou o desenvolvimento de uma oficina para tornar o ensino de programação como algo que atraísse os alunos e os mesmos pudessem compreender o sentido do aprendizado, utilizando-se assim, a estratégia de ensino da aprendizagem situada.

O público desta proposta de ensino são estudantes do ensino técnico integrado ao ensino médio, dos Institutos Federais de Ciência e Tecnologia.

A educação profissional técnica e tecnológica contempla os Institutos Federais de ciência e Tecnologia, o Colégio Pedro II e os Centros Federais de Educação Tecnológica do Rio de Janeiro e Minas Gerais, além da Universidade Tecnológica Federal do Paraná, que fazem parte da Rede Federal de Educação Profissional, Científica e Tecnológica, instituídas a partir da lei 11.892 de 29 de dezembro de 2008.

A finalidade dos Institutos Federais é formar e qualificar cidadãos, para que estes possam atuar nos diversos ramos da economia, nos mais diversos cargos e competências que o mundo do trabalho possuem, atendendo arranjos produtivos e demandas regionais (BRASIL, 2008).

A lei de criação dos Institutos Federais aponta objetivos tais como:

- Ministrando educação profissional e técnica de nível médio [...];
- Realizando pesquisas aplicadas [...];
- Desenvolvendo atividades de extensão, [...], em articulação com o mundo do trabalho [...];
- Estimular e apoiar processos educativos que levem à geração de trabalho e renda e à emancipação do cidadão [...].

Ressalta-se que o objetivo da educação profissional técnica e tecnológica contemporânea não é o ensino de uma profissão, ou de técnicas para realização de um trabalho específico, mas sim proporcionar aos estudantes condições de compreender o mundo do trabalho, em toda sua complexidade, as dinâmicas sócio-produtivas, elevando o estudante a um nível de autonomia, criticidade e criatividade adequados e requisitados pelo mundo do trabalho, no exercício da profissão escolhida (BRASIL, 2010).

Assim, a educação proposta por essas instituições é capaz de promover autonomia, emancipação dos estudantes, bem como a capacidade de solução de problemas de suas atividades práticas com uso e base em seus conhecimentos teóricos adquiridos, constituindo assim uma formação integral dos estudantes. Com essa premissa, o desenvolvimento do pensamento computacional, proposto nessa proposta de ensino, vai ao encontro a esses objetivos da educação profissional, pois tem como objetivo estimular do pensamento computacional dos estudantes do ensino médio técnico integrado. Para atender ao objetivo proposto, a oficina conta com atividades contextualizadas e baseadas nos pilares do pensamento computacional, sendo abstração, algoritmo, reconhecimento de padrões e decomposição.

## **PILARES DO PENSAMENTO COMPUTACIONAL**

Segundo Brackmann *et al.* (2017) o pensamento computacional possui quatro pilares, a “decomposição”, em que ao identificar um problema complexo é possível quebrá-lo em pedaços menores e mais fáceis de gerenciar; o “reconhecimento de padrões”, em que os problemas menores podem ser analisados individualmente com maior profundidade, identificando, assim problemas similares já solucionados anteriormente; a “abstração” quando se foca apenas nos detalhes importantes, ignorando informações irrelevantes; e “algoritmos” quando passos ou regras simples são criados para resolverem cada um dos subproblemas encontrados. Dessa forma, de acordo com o autor, ao seguir os passos ou preceitos utilizados para se criar um código-fonte, “é possível também que se compreenda por sistemas computacionais e, seja utilizado na resolução de problemas complexos de forma eficiente, independentemente da carreira profissional que o estudante deseja seguir.”

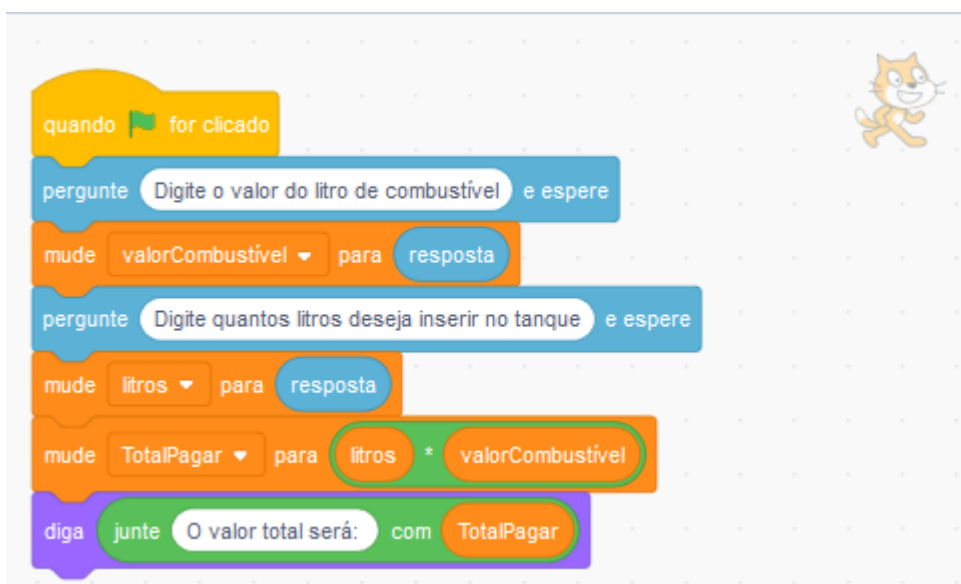
Para trabalhar o pensamento computacional escolheu-se a utilização dos softwares Scratch e Site Hora do Código como ferramenta de desenvolvimento dos códigos, pois ambos fazem uso da linguagem de programação por blocos, Blockly.



## PROGRAMAÇÃO POR BLOCOS

A programação de computadores utilizando linguagem de blocos não é tão recente, porém, ela emergiu com a crescente utilização de ferramentas como o Scratch, SNAP!,Blockly ou a Hour of Code. Essa tecnologia protege o usuário de possíveis erros de sintaxe, escrita da linguagem de forma incorreta, sendo que os ambientes apenas aceitam que os blocos corretos sejam conectados entre si, formando assim o corpo do programa. Essa linguagem de programação remete a um quebra-cabeça, sugerindo de forma visual o local adequado para o encaixe de cada componente(WEINTROP; WILENSKY, 2017).

A imagem a seguir ilustra um código escrito com linguagem de blocos.



**Figura 1.** Imagem extraída do Software de programação com blocos *Scratch*. Blocos encaixados.

**Fonte:** Os autores

## DESCRIÇÃO DA PROPOSTA DE ENSINO

Esta proposta de ensino trata-se de um oficina de curta duração, com uma carga horária de 20 horas-aula, dividida em 6 encontros, destinada a alunos do ensino médio técnico integrado para áreas não pertencentes ao eixo de computação. Os tópicos de programação abordados são as estruturas sequenciais, estruturas de decisão e estruturas de repetição, associados à atividades baseadas em conteúdos de Física (MRU), Química (Soluções),

Matemática (Porcentagens, custos), e atividades mais contextualizadas com atividades rotineiras ou de fácil abstração por parte dos estudantes, como compras ou viagens, além de gameficação, a reprodução de partes de jogos digitais conhecidos.

O software escolhido para a construção das soluções dos problemas propostos foi o Scratch, que se trata de um software de programação baseado em lógica de blocos, desenvolvido pelo Lifelong Kindergarten Group, do MIT Media Lab.

As aulas 1 e 2 são destinadas à introdução à programação, ambientação com a lógica de blocos e com o software Scratch e as primeiras atividades contextualizadas, utilizando estrutura sequencial, especificamente um deslocamento entre pontos/lugares, em que os estudantes deverão ser inseridos em um planejamento, escolha do destino, meio, custos e rotas, sendo que a contextualização é importante para a execução das atividades propostas em sequência.

### Quadro 1. Sequência didática da aula 1- Estrutura Sequencial

Atividade 1	Descrição
1.1	<p>Apresentação inicial sobre programação.</p> <ul style="list-style-type: none"> <li>• Introdução;</li> <li>• Objetivos;</li> <li>• Variáveis;</li> <li>• Entradas;</li> <li>• Saídas;</li> <li>• Processamento.</li> </ul>
1.2	<p>Apresentação dos Objetivos da oficina.</p> <ul style="list-style-type: none"> <li>• Introduzir a programação de computadores com blocos;</li> <li>• Ensinar de estruturas sequencias, de decisão e de repetição;</li> <li>• Promoção do pensamento computacional.</li> </ul>
1.3	<p>Contato inicial com programação de blocos</p> <ul style="list-style-type: none"> <li>• Atividade Star Wars. <a href="https://code.org/starwars">https://code.org/starwars</a></li> <li>• Atividade Sports. <a href="https://code.org/athletes">https://code.org/athletes</a></li> </ul>
1.4	<p>Ambientação com o software Scratch</p> <ul style="list-style-type: none"> <li>• Opção1 - Apresentação do software Scratch, utilizando o mangá disponível em: <ul style="list-style-type: none"> <li>○ <a href="https://www.researchgate.net/publication/263917806">https://www.researchgate.net/publication/263917806</a> <u>Programar um Jogo com Scratch - Um Gibi em Estilo Manga para Ensino de Programacao</u></li> </ul> </li> <li>• Outras Opções: <ul style="list-style-type: none"> <li>○ <a href="http://www.computacaonaescola.ufsc.br/?page_id=16">http://www.computacaonaescola.ufsc.br/?page_id=16</a></li> </ul> </li> </ul>



Figura 2. Atividade Star Wars

Fonte: os autores

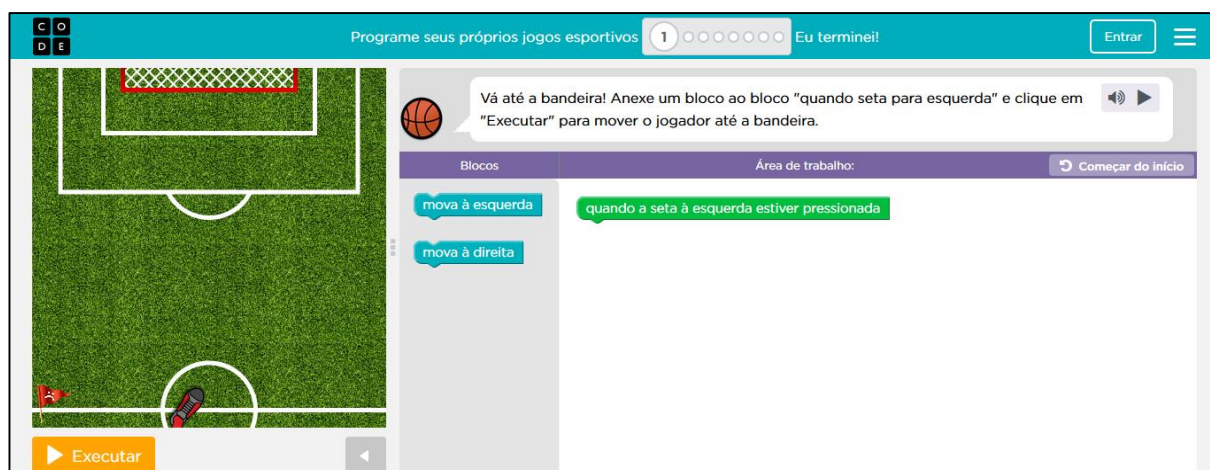


Figura 3. Atividade Sports

Fonte: os autores

## Quadro 2. Sequência didática da aula 2

Atividade 2	Descrição
2.1	<p>Atividade Movimento Retilíneo Uniforme (MRU) – conteúdo de física 1º ano do ensino médio.</p> <ul style="list-style-type: none"> <li>• <b>Contextualização:</b> inserir os estudantes em uma situação de planejamento de viagem, onde eles devem pensar o que é necessário para realizar uma viagem ou traslado com veículo particular, quilometragem a ser percorrida, valor do combustível, consumo do veículo;</li> <li>• <b>Descrição:</b> Para que o estudante entre no contexto sugere-se conduzir a contextualizar da seguinte forma: <ul style="list-style-type: none"> <li>○ 1ª Pergunta: Digamos que cada um de vocês irá fazer uma viagem de carro, o que é necessário para realizar essa viagem? <ul style="list-style-type: none"> <li>▪ Respostas esperadas/necessárias: Veículo; Origem; Destino; Distância; Dinheiro;</li> </ul> </li> <li>○ 2º Pergunta: Como calcula-se o custo dessa viagem, o que precisa-se para realizar esse cálculo? <ul style="list-style-type: none"> <li>▪ Respostas esperadas: Distância; Preço combustível; Consumo do veículo;</li> </ul> </li> <li>○ 3º Pergunta: Qual o cálculo do custo com base na distância, preço e consumo? <ul style="list-style-type: none"> <li>▪ Resposta: <math>(\text{Distância}/\text{Consumo}) \times \text{Preço Litro}</math>.</li> </ul> </li> <li>○ 4º Mostrar exemplo do Cálculo.</li> </ul> </li> </ul>
2.2	<p>1) 1.1 - Escrever um programa que recebe um valor em metros, e imprime o valor correspondente em km. 1 Quilometro = 1000 m.</p> <p>1.2 – Escrever um programa onde o usuário digita a quantidade de litros que uma pessoa abasteceu, o valor do litro de combustível e o programa exhibe o valor total a ser pago.</p>

	<p>1.3 – Um proprietário de veículo abasteceu o carro completamente de combustível até a trava da bomba. Percorrendo 400Km percebeu que precisaria abastecer novamente. Ao realizar o mesmo procedimento de encher o tanque até a trava de combustível foram colocados 45 litros de combustível no tanque. Com esses dados eles fez uma conta para saber quantos quilômetros por litro seu veículo fez de média (Km/L). Crie um programa para automatizar a solução desse problema. O usuário deve informar a Km percorrida e quantos litros de combustível foram utilizados, exibindo assim o consumo médio do veículo.</p>
2.3	<p>2) Escrever um programa que recebe a distância, em metros, a ser percorrida por um corredor e o tempo de duração da prova, em segundos. Calcular a velocidade média, em metros por segundo (m/s). <math>V = \text{distância}/\text{tempo}</math>.</p>
2.4	<p>3) Apresentar o site <a href="http://www.mapeia.com.br">www.mapeia.com.br</a>, demonstrar funcionalidades, cálculo de custo de combustível e tempo de duração da viagem. Explicação sobre os cálculos executados pelo site. Exercício: Criar um programa semelhante ao site Mapeia.</p> <p>Receber os seguintes valores:</p> <ul style="list-style-type: none"> <li>• do combustível a ser utilizado;</li> <li>• do consumo, km/l, do veículo;</li> <li>• distância da viagem;</li> </ul> <p>Imprimir os seguintes dados:</p> <ul style="list-style-type: none"> <li>• Custo total do combustível; (Litros * Preço do litro);</li> <li>• Tempo da viagem, considerando velocidade média de 80km/h. <math>\text{Tempo} = \text{Distância}/\text{Velocidade}</math>.</li> </ul>
2.5	<p>Atividade misturas químicas – Saturadas, Insaturadas e Saturadas com corpo de Fundo.</p> <p>Explicação sobre conceitos de:</p> <ul style="list-style-type: none"> <li>• Solução: Solução é toda mistura homogênea de duas ou mais substâncias, sendo formadas por um solvente, normalmente em maior quantidade, e um soluto, normalmente em menor quantidade.</li> <li>• Soluto: Composto que é diluído;</li> <li>• Solvente: Composto que dilui;</li> <li>• Soluções saturadas: solução que contém uma quantidade de soluto igual à solubilidade a uma dada temperatura. Na solução saturada o soluto dissolvido e o não dissolvido estão em equilíbrio dinâmico entre si. Na solução, para uma determinada temperatura há a quantidade limite que o solvente consegue dissolver daquele determinado soluto, sem formar corpo de fundo.</li> <li>• Soluções insaturadas: solução que contém uma</li> </ul>

	<p>quantidade de soluto inferior à solubilidade a uma dada temperatura.</p> <ul style="list-style-type: none"> <li>• Soluções saturadas com corpo de fundo - Supersaturadas: solução que contém uma quantidade de soluto superior a solubilidade a uma dada temperatura. A solução supersaturada é instável, e a mínima perturbação do sistema faz com que o excesso de soluto dissolvido precipite, tornando-se uma solução saturada com presença de corpo de fundo.</li> </ul>
2.6	<ul style="list-style-type: none"> <li>• <b>Contextualização:</b> Inserir os estudantes em situação de mistura de componentes em casa, com achocolatados, sal, gelatina, ou alguns outros componentes sólidos que possam ser diluídos em um solvente (água, leite).</li> <li>• <b>Descrição:</b> Para preparar um suco, utilizando sucos em pó, a embalagem possui uma recomendação. O conteúdo de uma embalagem faz 1 Litro de Suco. Essa recomendação trata-se da quantidade de solvente necessária para que o produto fique com sabor ideal segundo o fabricante. Porém, caso use a mesma quantidade para 100mL, ou 10L, a concentração ficará diferente, deixando o suco com sabor mais concentrado ou menos concentrado.</li> </ul>
2.7	<p>1) 1.1 - Crie um programa onde o usuário digita o número de pacotes de suco que ele irá utilizar para fazer uma bebida e o rendimento desse suco. Exemplo 1 Pacote faz 1 Litro. O sistema deverá informar quantos litros de água serão necessários para fazer o suco conforme a concentração ideal indicada.</p> <p>1.2 - Escreva um programa que lê a quantidade de soluto máximo para ser dissolvido em 1L de solvente e a quantidade de soluto que deseja-se misturar. O programa irá imprimir a quantidade de solvente necessária, para dissolver a quantidade de soluto, sem que a solução fique saturada com corpo de fundo e nem insaturada, ou seja, na concentração ideal (saturada).</p>

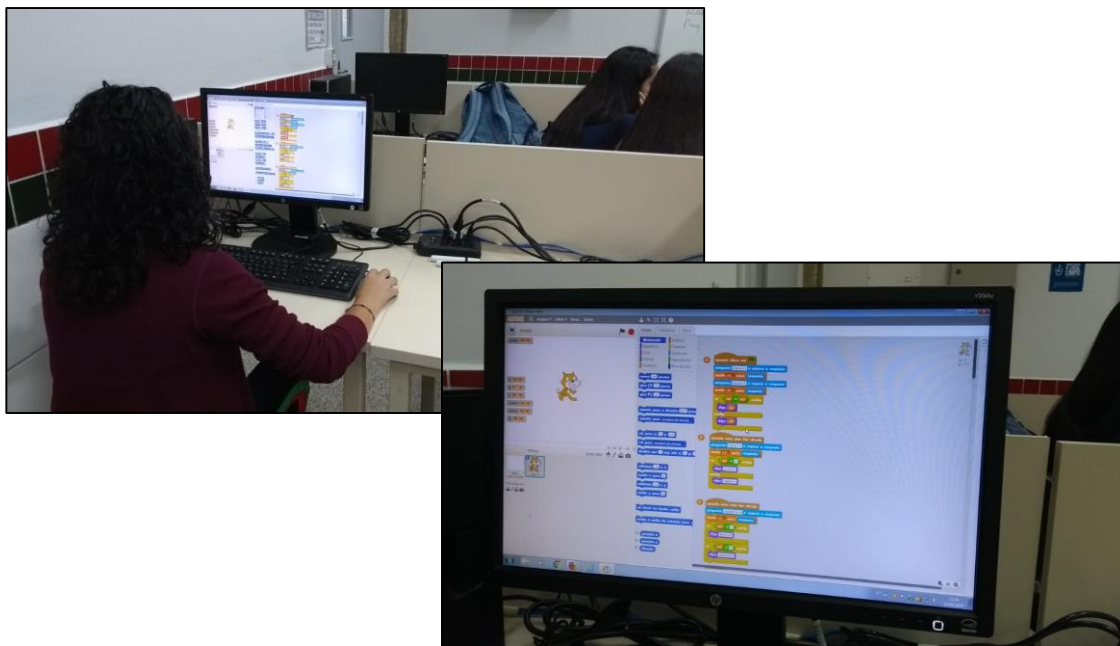


Figura 4. Primeiros Passos no Scratch.

Fonte: os autores

As aulas 3 e 4 são destinadas às atividades pertinentes a estrutura de decisão. Nessas aulas, as atividades contemplam contextos de situação de aprovação ou reprovação e aquisição de itens com desconto em lojas.

### Quadro 3. Sequência didática da aula 3 e 4 – Estrutura de Decisão

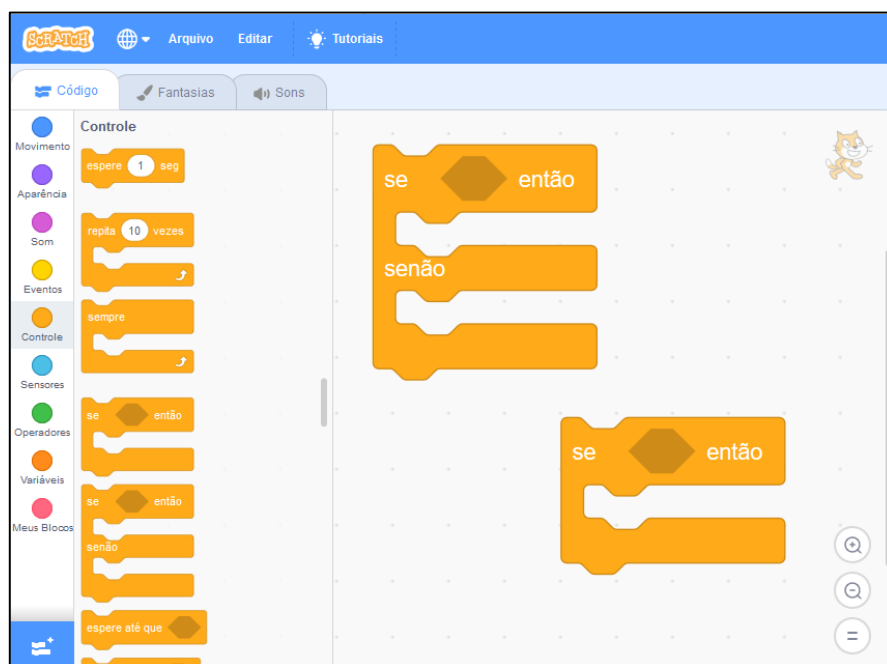
Atividade 3	Descrição
3.1	<p>Apresentação sobre decisões e estruturas de decisão</p> <ul style="list-style-type: none"> <li>• IF;</li> <li>• ELSE;</li> <li>• ELSE IF;</li> </ul> <p>Exemplificação de decisões:</p> <ul style="list-style-type: none"> <li>• Exemplos de decisões tomadas no dia a dia (que roupa usar, que comida ingerir, que time torcer);</li> <li>• Exemplos de escolhas a serem feitas (economizar ou gastar, comer de tudo ou fazer dieta);</li> <li>• Exemplo de conceitos no final do ano (A, B ou C aprovação, D reprovação);</li> </ul>
3.2	<ul style="list-style-type: none"> <li>• <b>Contextualização:</b> Inserir os estudantes na entrega dos boletins do final de ano, pensando que os conceitos finais determinam a aprovação ou reprovação em algum componente curricular.</li> <li>• <b>Descrição:</b> Os componentes curriculares são avaliados ao final do período letivo com conceitos A (aprendizado</li> </ul>

	pleno), B (parcialmente pleno, C (suficiente), sendo o estudante avaliado com esse conceito considerado APROVADO e D (insuficiente), sendo que o aluno que obtiver esse conceito final considera-se REPROVADO.
3.3	1) Criar um programa que leia o conceito final de um estudante na disciplina de matemática. Se o resultado for A, B ou C, o sistema deverá mostrar “APROVADO”, senão, o sistema deverá exibir a mensagem “REPROVADO”.
3.4	<ul style="list-style-type: none"> <li>• <b>Contextualização:</b> Inserir o estudante em uma situação de compra de itens em lojas. Pode ser qualquer tipo de produto ou loja, aquilo que for de maior familiaridade de cada estudante. Que ao se deparar com um tipo de promoção, precisa calcular os valores dos produtos com o desconto determinado pela cor da etiqueta que está no produto.</li> <li>• <b>Descrição:</b> Uma loja está com uma promoção em seus produtos, dando descontos de 15% até 50%, diferenciando os produtos com etiquetas coloridas presas nas etiquetas originais do produto. Neste caso, podem ser dados exemplos do cálculo do custo do produto depois de aplicado o desconto. Exemplo, um produto de R\$ 75,00 está com uma etiqueta verde. A etiqueta verde significa que o produto possui 30% de desconto. Ao aplicar o desconto, quanto ficaria o preço desse item?</li> </ul>
3.5	2) Sistema de verificação de descontos em lojas com base na etiqueta do produto. Uma loja possui etiquetas coloridas nas roupas, cada uma referente a um determinado desconto. Branco 50% de desconto, Verde 30% de desconto e Azul 15% de desconto. Crie um programa que leia o valor de um produto e a cor da etiqueta, dessa forma, o sistema deverá imprimir o valor final do produto e qual o valor do desconto dado.
<b>Atividade 4</b>	<b>Descrição</b>
4.1	<ul style="list-style-type: none"> <li>• <b>Contextualização:</b> Inserir o participante em jogos eletrônicos onde ações de destruir objetos ocorram ao clique do mouse sobre esses objetos.</li> <li>• <b>Descrição:</b> Um exemplo desse jogo é o DUCK HUNT, jogo clássico de “matar patos” de videogames dos anos 2000. Nesse jogo patos saíam de trás de um gramado e movimentavam-se pela tela por um determinado período de tempo e saíam da tela caso não fossem acertados pelo tiro do jogador. Ao passar das fases a velocidade e critérios de continuação eram alterados. Em caso de haver internet, o jogo pode ser exibido via Youtube.</li> </ul>
4.2	1) 1.1 - Criar um jogo em que atores apareçam em posições aleatórias da tela, e se o mouse for clicado sobre algum desses objetos a pontuação aumente, caso contrário, a pontuação será reduzida. O fim do jogo deve ser definido por um número limite



de cliques errados, ou cliques não realizados sobre algum ator.

1.2 – Alterar o jogo anterior para que os objetos saiam de uma posição pré-definida e movam-se semelhante aos do jogo mostrado, realizando de três a quatro movimentos antes de saírem da tela. No Scratch os movimentos de deslize ocorrem por um determinado período de tempo. Recomenda-se que cada movimento inicie com 2 segundos e com o passar de fases esse tempo vá subtraindo. Na aplicação do produto usou-se 50 fases, sendo que o tempo era subtraído a cada 5 fases que o usuário conseguia avançar. O jogo finaliza quando o jogador errava/perdia 8 objetos.

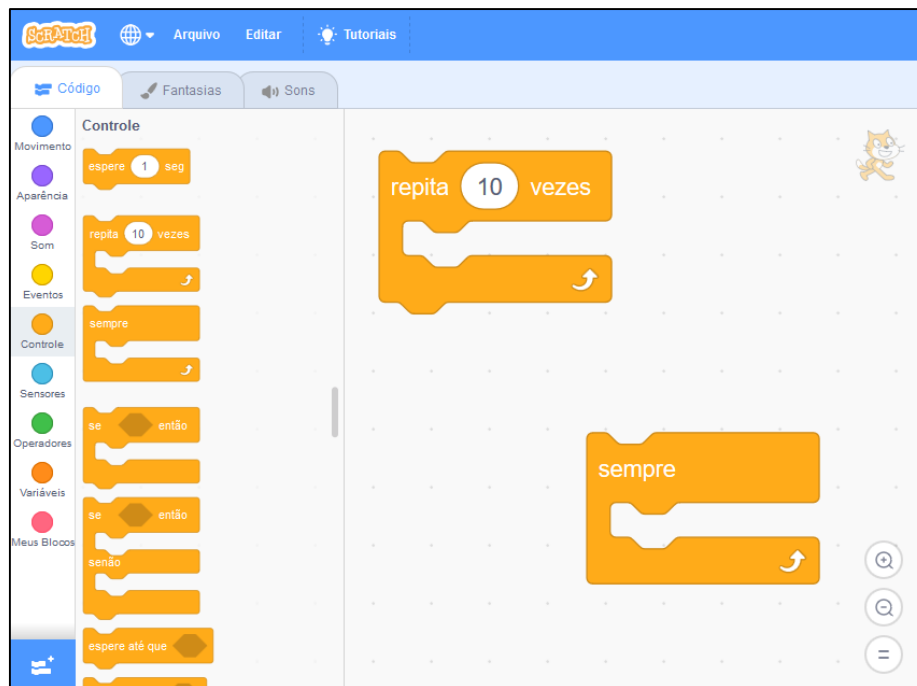


**Figura 5. Estruturas de Decisão Scratch**  
Fonte: os autores

As aulas 5 e 6 são destinadas às atividades relacionadas com estruturas de repetição. As aulas dessa estrutura objetivam a criação de jogos, sendo a “gameificação” uma das técnicas mais utilizadas na promoção do pensamento computacional, quando destinado à faixa etária do ensino fundamental e médio.

**Quadro 4. Sequência didática da aula 5 e 6 – Estrutura de Repetição**

<b>Atividade 5</b>	<b>Descrição</b>
5.1	Explicação sobre estruturas de repetição <ul style="list-style-type: none"> <li>• Looping;</li> <li>• Condição de parada.</li> </ul>
5.2	Exemplificação de atividades repetitivas <ul style="list-style-type: none"> <li>• Caixa de supermercado;</li> <li>• Atividades rotineiras;</li> <li>• Relógio.</li> </ul>
5.3	1) Aplicar a atividade da hora do código Labirinto clássico, disponível em: <a href="https://studio.code.org/hoc/1">https://studio.code.org/hoc/1</a> , acesso em 02/10/2018.
5.4	2) Aplicar a atividade da hora do código Frozen, disponível em: <a href="https://studio.code.org/s/frozen/stage/1/puzzle/1">https://studio.code.org/s/frozen/stage/1/puzzle/1</a> , acesso em: 02/10/2018.
<b>Atividade 6</b>	<b>Descrição</b>
6.1	<ul style="list-style-type: none"> <li>• <b>Contextualização:</b> Mostrar um jogo de guitarra, onde objetos vão caindo na tela e em um determinado momento, se clicado a tecla correspondente ele vai somando pontos, se as teclas forem pressionadas sem que o objeto esteja no local correto, perde-se ponto. Se a pontuação zerar acaba o programa, que termina, exceto no caso anterior, ao final de uma música pré-determinada.</li> </ul>
6.2	1) Criar um jogo semelhante ao mostrado na contextualização, com objetos de preferência dos estudantes.



**Figura 6. Estrutura de Repetição no Scratch**  
**Fonte: os autores**

Cada uma das aulas possuem um objetivo específico e atendem a um ou mais pilares do pensamento computacional. O desenvolvimento do pensamento computacional proporciona ao estudante a decomposição de problemas em partes menores, reconhecimento de padrões dos problemas, abstração de situações e capacidade de desenvolvimento de algoritmos que automatizem a solução das atividades propostas.

Para aplicação da oficina são necessários alguns softwares e recursos, abordados na sessão a seguir.

## REQUISITOS PARA APLICAÇÃO DA OFICINA

A proposta de ensino possui uma parte inicial destinada à ambientação dos estudantes do ensino técnico integrado ao ensino médio com a linguagem de programação por blocos e uma ambientação com o software Scratch.

Para a utilização dos softwares mencionados faz-se necessário o atendimento dos seguintes requisitos:

1. **Hardware:** Laboratório de informática, equipado com computadores ou notebooks, em número suficiente para o número de alunos da turma. (Laboratório ideal para se trabalhar com 20 estudantes na turma).

2. **Software:** As aplicações da hora do código estão disponíveis no site <https://hourofcode.com/br/learn>. (Necessário acesso à internet).
3. As atividades no Scratch podem ser realizadas tanto na versão on-line quanto off-line:
  - a. **Versão Offline:** Scratch – Offline versão mais atualizada. <https://scratch.mit.edu/download>. Necessário a instalação do software Adobe Air (<https://get.adobe.com/br/air/>).
  - b. **Versão On-line:** <https://scratch.mit.edu/projects/editor/?tutorial=getStarted> (Necessário acesso à internet).

## RELAÇÃO ENTRE ATIVIDADES DE PILARES DO PENSAMENTO COMPUTACIONAL

O quadro 5 relaciona as atividades da oficina com os pilares do pensamento computacional. É importante frisar que as atividades podem possuir uma maior ênfase em algum dos quatro pilares, porém as atividades podem relacionar-se em menor grau com os outros pilares do pensamento computacional. No quadro as células marcadas com “**X**” (x em negrito).

**Quadro 5. Sequência didática da aula 4 e 5 – Estrutura de Repetição**

Atividade/Pilar	Abstração	Reconhecimento de padrões	Decomposição	Algoritmo
1.1 – Apresentação	<b>X</b>			
1.2 – Objetivos				
1.3 – Star Wars/Sports			X	<b>X</b>
1.4 – Mangá – Scratch	<b>X</b>			X
2.1 – Contextualização MRU	<b>X</b>		X	
2.2 – Conversão m -> cm	<b>X</b>			X
2.3 – Velocidade	<b>X</b>		X	X
2.4 – Mapeia.com.br	<b>X</b>	X	X	X
2.5 – Misturas Químicas	<b>X</b>	X	X	
2.6 – Contextualização	<b>X</b>	X	X	
2.7 – Solução	<b>X</b>	X	X	X
3.1 – IF – ELSE	<b>X</b>	X	X	
3.2 – Contextualização				
3.3 – Boletim Escolar	X			<b>X</b>
3.4 – Contextualização				
3.5 – Liquidação em Loja	X		<b>X</b>	X
4.1 – Contextualização				
4.2 – Jogo	X	X	<b>X</b>	X
5.1 – Repetição	<b>X</b>	X		
5.2 – Exemplos	<b>X</b>	X		
5.3 – Labirinto		<b>X</b>		X
5.4 – Frozen		<b>X</b>		X
6.1 – Contextualização				
6.2 – GuitarHero	X	<b>X</b>	X	X

## CONSIDERAÇÕES FINAIS

A programação de computadores não é mais uma habilidade exclusiva de quem pretende seguir na área de computação, trata-se de uma habilidade acessível e essencial no mundo tecnológico atual.

Contudo, programar não é uma tarefa trivial, e há vários aspectos no processo de ensino/aprendizagem, além da interação entre os atores, estudantes e professores.

Apesar disso, há inúmeras iniciativas de ensino de programação, desde séries iniciais, fundamental I, por exemplo, em que técnicas, ferramentas e metodologia são específicos para a faixa etária correspondente à estudantes do ensino médio, reconhecendo que não se pode trabalhar da mesma forma com todas as faixas etárias. Outro ponto importante é o objetivo do ensino, pode ser muito diverso, dessa forma, a maneira de ensinar também deverá ser bastante diversificada.

Esse produto aborda o ensino baseado no contexto, pois inserir o aluno em um contexto, em que todas as atividades que irá desenvolver façam sentido para ele, possibilita a compreensão e a aprendizagem para resolver problemas de situações reais do cotidiano. Outro aspecto positivo da aprendizagem contextualizada, é que foge da abstração de exercícios tradicionais de programação, pois fazem sentido ao estudante, que pode verificar a aplicação das técnicas e procedimentos na prática.

Reconhecendo a linguagem de programação por blocos como uma ferramenta intuitiva, e comprovada em diversos estudos como facilitador do processo de ensino/aprendizagem de lógica de programação, a opção por esta foi ao encontro aos objetivos da proposta de ensino, que é desenvolver o pensamento computacional dos estudantes do ensino técnico integrado ao ensino médio.

## REFERÊNCIAS

BRACKMANN, C. et al. **Pensamento Computacional Desplugado: Ensino e Avaliação na Educação Primária Espanhola.** . In: VI CONGRESSO BRASILEIRO DE INFORMÁTICA NA EDUCAÇÃO. Recife, Pernambuco, Brasil: 27 out. 2017Disponível em: <<http://www.br-ie.org/pub/index.php/wcbie/article/view/7487>>. Acesso em: 27 ago. 2018

BRASIL. Lei nº 11892. . 29 dez. 2008.

BRASIL. **Concepções e Diretrizes**, 2010.

LAVE, J. A prática da aprendizagem. In: **Teorias Contemporâneas da Aprendizagem**. Porto Alegre: Penso, 2013. p. 235–245.

WEINTROP, D.; WILENSKY, U. Comparing Block-Based and Text-Based Programming in High School Computer Science Classrooms. **ACM Transactions on Computing Education**, v. 18, n. 1, p. 1–25, 27 out. 2017.