

# **CRIATIVIDADE E RESOLUÇÃO DE PROBLEMAS: O ENSINO DA MATEMÁTICA POR MEIO DA LÓGICA PROGRAMACIONAL**

Lais Santos Brasil

*Orientada por*

Marcelo Oliveira Dias



**PPGEduCIMAT**

## Apresentação

Bem-vindo(a) ao nosso caderno de atividades!

Este caderno foi cuidadosamente elaborado para proporcionar uma experiência de aprendizado envolvendo elementos básicos da linguagem de programação no ensino de geometria, ele é resultado de uma pesquisa de dissertação e busca contribuir com a promoção de uma educação enriquecedora e interativa. Nele, você encontrará uma proposta de intervenção com uma sequência didática para estimular o raciocínio lógico, criatividade e capacidade de resolução de problemas.

Esperamos que você aproveite cada momento e divirta-se enquanto aprende. Bons estudos!

Lais Santos Brasil.

Marcelo Oliveira Dias.

## Sumário

Apresentação	1
1. A aproximação da resolução de problemas com o pensamento computacional	3
2. Fundamentos da Programação	5
3. Proposta de aplicação	11
<b>3.1. Estrutura da sequência didática proposta</b>	<b>12</b>
3.2. Atividades desplugadas	13
<b>Atividade 1 - Algo movimento</b>	<b>13</b>
<b>Atividade 2 - Algoritmo do café</b>	<b>15</b>
Atividade 3 - Cálculo de área de polígonos regulares	19
Atividade 4 - Cálculo de área de um polígono qualquer com o Teorema de Pick	20
<b>4. Códigos em Python</b>	<b>22</b>
<b>5. Referências</b>	<b>25</b>

## 1. A aproximação da resolução de problemas com o pensamento computacional

Com o avanço das tecnologias digitais e o aumento da conectividade nas ações do cotidiano, tem-se crescido os debates sobre o uso dessas tecnologias digitais e a sua promoção por uma educação para a cidadania digital (CETIC, 2022). Além disso, neste ínterim, a Base Nacional Comum Curricular (BNCC) tem como uma das habilidades a serem atendidas pelos estudantes do Ensino Médio na área de Matemática e suas tecnologias, o uso da linguagem de programação. No entanto, no cenário atual, segundo a base de microdados da Cetic (2020), 92,4% das escolas brasileiras não possuem salas de recurso de Robótica. Por isso, esta pesquisa optou em trabalhar com atividades desplugadas<sup>1</sup>, por serem uma possibilidade para integrar esses estudantes de diferentes classes sociais ao mundo digital.

Na BNCC, a inclusão da linguagem de programação é obrigatória na disciplina de Matemática e suas tecnologias na etapa do Ensino Médio. E nesta pesquisa, mostrou-se a relação existente entre a resolução de problemas matemáticos a partir das fases de Polya com as habilidades fundamentais do pensamento computacional.

**Tabela 1:** relação entre PC e RP

Relação existente entre PC e RP			
Habilidade	Definição	Definição	Fase de Polya
ABSTRAÇÃO	Organização das informações disponíveis. Implica em filtrar dados,	Compreensão do problema, quais informações usar ou não.	COMPREENSÃO DO PROBLEMA

<sup>1</sup> Segundo Brackmann (BRACKMANN, 2017) são consideradas atividades desplugadas, aquelas atividades que introduzem conceitos computacionais por meio de uma aprendizagem cinestésica (cartas, jogos, atividades...), desde que sejam offlines.

	considerando o que é ou não é necessário.		
DECOMPOSIÇÃO	Dividir os problemas em partes menores para simplificar a resolução.	Desenvolvimento de um possível caminho que deverá ser trabalhado, isolando as principais partes do problema.	ESTABELECENDO UM PLANO
ALGORITMO	As instruções foram descritas quando o problema foi decomposto.	Realização do plano que foi estabelecido na fase anterior.	EXECUÇÃO DO PLANO
RECONHECIMENTO DE PADRÓES	Reconhecer as características em comum entre o problema e a solução, a fim de encontrar uma solução eficaz.	Para a resolução do problema proposto e reflexão da solução proposta.	RETROSPECTO

Fonte: dados da pesquisa

Com a conexão estabelecida entre esses dois campos, torna-se viável abordar os fundamentos da linguagem de programação nas aulas de Matemática.

Além disso, existem possíveis benefícios para os estudantes a partir da aprendizagem transdisciplinar entre Matemática e Ciência da Computação. Segundo a pesquisa realizada por Tew et al. apud Brackmann (2008, p. 44)

após o aprofundamento de conhecimento de seus estudantes em conceitos da Computação, alunos que tinham dificuldades em outras disciplinas, começaram a correlacionar esses conceitos com os temas trabalhados nas outras disciplinas e tiveram um rendimento superior, inclusive comparável aos melhores alunos.

Para além de uma aplicação instrumental do uso das TICs na sala de aula, em conformidade com Dias (2021), é preciso que reflitamos sobre os desafios trazidos por essas tecnologias para as salas de aula, possibilitando a alfabetização tecnológica para aqueles que normalmente não possuem acesso ao mundo digital possam atuar como cidadãos críticos.

## 2. Fundamentos da Programação

Para esta seção, utilizaremos como base os livros Manzano (2000 e 2019). Para que um computador funcione, será necessário ser programado. E a programação será o processo de criar instruções para que o computador execute uma tarefa. Durante essa criação será necessário conhecer alguma linguagem de programação, pois serão a partir delas que são controladas as ações de um computador.

Nesta seção, exploraremos os conceitos fundamentais do português estruturado ou portugol, que é uma linguagem de projeto. Utiliza-se para familiarizar as pessoas a escrita de um programa de computador, mas sem se preocupar com a simbologia necessária de alguma linguagem específica. Segundo Manzano (p. 95, 2019),

Ela possibilita, em um primeiro momento, rapidez na escrita do projeto de código do programa sem se preocupar com os rigores técnicos e sintáticos particulares de cada uma das linguagens de programação existentes, além de servir como instrumento de documentação de código de programa, facilitando os trabalhos de manutenção e de auditoria. Em um segundo momento possibilita

facilidade na tradução desse código preliminar em uma linguagem de programação que, de preferência, tenha características estruturadas.

Como é uma linguagem que pode se adaptar de acordo com o idioma de cada lugar, os comandos vão depender da região. Por isso, segue uma tabela com alguns comandos mais conhecidos.

**Tabela 2:** alguns comandos do portugol

Comando	Aplicabilidade
EXECUTE	Executa um bloco de código ou uma função específica.
ENTÃO	Utilizado dentro de uma estrutura condicional (SE), determinando o que deve ser executado se a condição for verdadeira.
ESCREVA	Exibe uma mensagem ou valor na tela.
FIM	Marca o término do programa.
FIM_SE FIM_ENQUANTO	Indica o término de um bloco de código. É usado para encerrar funções e estruturas como se, enquanto e repita.
FUNÇÃO	Declara uma função que pode ser chamada em diferentes partes do programa.
INÍCIO	Marca o início do programa.
LEIA	Obtém um valor digitado pelo usuário.
REPITA	Executa repetidamente um bloco de código até que uma condição seja verdadeira.
SE	Estrutura condicional usada para verificar se uma condição é verdadeira e executar um bloco de código.
SENÃO	Define o bloco de código que será executado caso a condição do SE não seja verdadeira.
VAR	Declaração de variáveis no programa.

Fonte: adaptado de Manzano (2019)

Também existem ferramentas responsáveis para auxiliar na escrita das instruções que serão solicitadas a um computador e envolvem operações matemáticas, conhecidos como os Operadores Aritméticos. A tabela a seguir, apresenta alguns operadores que podem ser utilizados em expressões aritméticas.

**Tabela 3:** operadores aritméticos

Operação	Operador aritmético	Descrição	Exemplo
Adição ( $a + b$ )	+	Soma de dois valores	resultado $\leftarrow 1 + 2$
Subtração ( $a - b$ )	-	Subtrai o segundo valor do primeiro	resultado $\leftarrow 10 - 4$
Multiplicação ( $a \cdot b$ )	*	Multiplica dois valores	resultado $\leftarrow 7 * 2$
Divisão ( $a \div b$ )	/	Divide o primeiro valor pelo segundo	resultado $\leftarrow 10 / 5$
Resto da divisão ( $a$ )	%	Calcula o resto da divisão entre dois números inteiros	resultado $\leftarrow 10 \% 3$
Potenciação ( $a^b$ )	^	Eleva o primeiro número à potência do segundo	resultado $\leftarrow 2 ^ 4$

Fonte: adaptado de Manzano (2019)

Além dos seus comandos e seus operadores aritméticos, é preciso compreender as etapas de ação de um computador. As três entradas principais, são:

- Entrada de dados;
- Processamento desses dados;
- Saída dos resultados.

Sendo assim, é preciso sinalizar quando o algoritmo começará e como receberá os dados disponíveis. A declaração das variáveis é muito importante, pois serão esses dados que poderão ser armazenados na memória do computador.

var

idade: inteiro

nome: caractere

altura: real

Na segunda etapa, será o momento em que escreverá o código que deseja ser executado. Será o momento de transformações dos dados que foram inseridos na etapa anterior, podendo utilizar as estruturas de controle como se, se não, para, enquanto etc.

inicio

escreva ("Digite seu nome: ")

leia (nome)

escreva ("Digite sua idade: ")

leia (idade)

se idade >= 18 entao

escreva ("Você é maior de idade.")

senao

escreva ("Você é menor de idade.")

fim\_se

fim\_algoritmo

Por fim, tem-se a última etapa destinada a saída de dados, será o momento em que o computador enviará os dados processados. Para facilitar a compreensão de como escrever um problema utilizando esta linguagem, será apresentado um exemplo de como escrever um programa que verifique se um número é ímpar ou par.

## EXEMPLO

Utilizando a fase abstração de Polya, precisa-se entender o que significa um número ser par ou ímpar. Por definição, todo número será par quando ele der resto zero da divisão por 2. Após compreender o problema, algumas possíveis fases do plano são:

- Solicitar ao usuário um número inteiro;
- Ler o número digitado;
- Verificar se o número é divisível por 2;
- Se o resto da divisão por 2 for igual a 0, então o número é par. Caso contrário, o número é ímpar;
- Exibir a resposta ao usuário.

A execução do plano será o algoritmo desenvolvido.

algoritmo "ParOuImpar"

var

numero: inteiro

início

escreva ("Digite um número inteiro: ")

leia (numero)

resto <- numero / 2

se resto == 0, escreva ("O número ", numero, " é PAR.")

senão

escreva ("O número ", numero, " é ÍMPAR.")

fim\_se

fim\_algoritmo

Por fim, na última fase - retrospecto - testa-se o algoritmo com diferentes valores para garantir que ele esteja funcionando corretamente.

Além da escrita, é importante se atentar às estruturas lógicas de um programa, aos tipos de dados e à noção de variável. Existem três tipos de estruturas lógicas (linear, estruturada e modular).

**Tabela 4:** estruturas lógicas de um algoritmo

Tipos de estruturas	
Linear	Esta técnica é a mais parecida com o modelo tradicional da resolução de problemas, em que os elementos estão ordenados por uma só propriedade, de forma que cada um deles é executado passo a passo de cima para baixo.
Estruturada	Utiliza estruturas de controle como decisões e laços. Permite maior flexibilidade e controle do fluxo do programa.
Modular	Divide o código em módulos independentes. Facilita a manutenção e reutilização de partes do programa.

Fonte: adaptado de Manzano (2019)

Ao começar qualquer programa, é importante definir as variáveis que serão informações a serem tratadas por um computador, e estas informações podem ser caracterizadas por três tipos de dados: numéricos, caracteres ou lógicos. E o conceito de variável é utilizado tanto na Matemática quanto na Computação, mas há diferenças importantes entre os dois contextos.

**Tabela 5:** diferença do conceito de variável nas duas áreas

	Matemática	Computação
Definição	Representa um valor desconhecido ou um número que pode mudar	Representa um espaço na memória do computador que armazena dados.

	dentro de uma equação.	
Tipo	Geralmente um número real, inteiro etc..	Pode armazenar diferentes tipos de dados (inteiros, reais, caracteres, booleanos etc.).
Alteração do seu Valor	O valor é fixo dentro de uma equação específica.	O valor pode ser alterado durante a execução do programa.
Uso	Resolver equações e expressar relações matemáticas.	Armazenar e manipular dados durante a execução do código.

Fonte: autoria própria

### 3. Proposta de aplicação

Após apresentar ao longo deste documento algumas justificativas sobre a inclusão de fundamentos da lógica programacional no Ensino de Matemática e apoiar, quando possível, o uso de tecnologias digitais em sala de aula, será apresentada a seguir uma sequência didática que possui como objeto de aprendizagem o cálculo de área de polígonos regulares e não regulares por meio da resolução de problemas utilizando a escrita estruturada do pseudocódigo.

A sequência didática foi elaborada pelos autores e possui 4 atividades, sendo duas delas criações de outros autores que foram vinculadas a essa sequência didática. Pode ser desenvolvida ao longo de pelo menos duas aulas compostas por dois tempos de 50 minutos cada. A primeira aula será destinada a familiarização dos estudantes com os fundamentos da escrita estruturada e na segunda será o desenvolvimento dos algoritmos para escrever uma solução para os problemas solicitados. O público-alvo das atividades propostas é constituído por turmas do Ensino Médio, mas dependendo de como o professor aplique essas atividades, a sequência pode ser utilizada para anos anteriores. Por último, a sequência didática a seguir visa trabalhar as seguintes habilidades da Base Nacional Comum Curricular (Brasil, 2018).

### 3.1. Estrutura da sequência didática proposta

Aula 1					
Objetivos	Etapas	Estratégias	Recursos	Local	Carga horária
	<ol style="list-style-type: none"> <li>1. Introdução sobre a relação entre a matemática e a computação;</li> <li>2. Atividade “Algo movimentos”;</li> <li>3. Atividade “Algoritmo do café”;</li> <li>4. Finalização.</li> </ol>	<ol style="list-style-type: none"> <li>1. Aula expositiva sobre os fundamentos da programação;</li> <li>2. Atividades.</li> </ol>	Papel, cartas, projetor.	Sala de aula	1h 40min

Aula 2					
Objetivos	Etapas	Estratégias	Recursos	Local	Carga horária
	<ol style="list-style-type: none"> <li>1. Introdução a lógica de programação;</li> <li>2. Cálculo de área de polígonos regulares;</li> <li>3. Cálculo de área de polígonos não-regulares</li> </ol>	<ol style="list-style-type: none"> <li>1. Retomada dos fundamentos dados na aula anterior;</li> <li>2. Atividades.</li> </ol>	Papel quadriculado ou pontilhado, projetor.	Sala de aula	1h 40min

### 3.2. Atividades desplugadas

Neste momento, apresentamos para você uma sequência de atividades que introduzem a escrita de soluções algorítmicas por meio de pseudocódigos. Para a realização desta atividade, será necessário a impressão do baralho que será

disponibilizado, da folha com os problemas<sup>2</sup>, ou apenas o compartilhamento por meio de uma apresentação em slide que todos possam visualizar.

### **3.3. Aula 1**

Nesta aula, orientamos que o(a) professor(a) disponibilize pelo menos três momentos: o primeiro para a apresentação dos fundamentos da escrita algorítmica e a sua aproximação com a resolução de problemas matemáticos, o segundo para a familiarização dos estudantes por meio da aplicação das atividades e o terceiro para a finalização da aula e feedback.

#### 3.3.1. Antes das atividades

Sugerimos que o material necessário seja impresso antes da aplicação da atividade. E que 30 minutos iniciais sejam disponibilizados para a familiarização dos estudantes em relação importância dos conhecimentos computacionais, tendo como foco a escrita de máquina. Caso seja possível, também sugerimos que o(a) professor(a) faça uma introdução falando da relação entre a matemática e a computação.

## Atividade 1 - Algo movimento

O link de acesso da atividade é:  
[https://www.computacional.com.br/download\\_algocards.php?id=por\\_color](https://www.computacional.com.br/download_algocards.php?id=por_color)

A atividade escolhida é conhecida como “Algo Movimento” e foi desenvolvida por Brackmann (2019). Ao acessar o link acima, o leitor se deparará com o baralho necessário para realizar a atividade. Por ser uma atividade desplugada, não será necessário o uso de recursos digitais, como computadores ou celulares, ao longo de sua aplicação, o único material necessário será um baralho que deve ser entregue a cada dupla de estudantes. Tem-se como objetivo que o estudante identifique e desenvolva um comando claro e objetivo, além de reconhecer o que é esperado como entrada e saída.

**Figura 1:** baralho algo movimento

---

<sup>2</sup> Está disponível nas páginas 23 e 24 deste documento.



Fonte: Brackmann (2019)

Esta atividade consiste em dividir os alunos em duplas de modo em que alguns serão “robôs” e outros os “programadores”, os estudantes que forem o programadores deverão utilizar o baralho indicado para instruir os estudantes que forem os robôs a realizarem alguma ação. A seguir, temos algumas recomendações do autor da atividade:

- Deve ser realizada em uma sala em que os pisos tenham delimitações claras, como pisos de porcelanato ou pisos cerâmicos;
- Podem ser usados objetos adicionais para atingir a meta solicitada.

#### Sugestão de aplicação

Faça o robô desenhar um quadrado em que o seu lado seja igual a dois pisos.

Neste momento, espera-se que os estudantes fragmentem a situação, pensando em quais comandos deverão ser realizados para que o robô execute o movimento exato.

- Inicie o movimento em um dos vértices do piso;
- Ande para frente;
- Ande para frente;
- Gire à esquerda;
- Ande para frente;
- Ande para frente;
- Gire à esquerda;

- Ande para frente;
- Ande para frente;
- Gire à esquerda;
- Ande para frente;
- Ande para frente;
- Gire à esquerda.

Depois que a ação for realizada, é possível solicitar aos estudantes uma reflexão sobre a solução desenvolvida e se existem outras maneiras de otimizá-la. Como, por exemplo:

4 vezes a junção dos seguintes comandos (ande para frente, ande para frente e gire à esquerda).

Depois da atividade, reúna a turma para um momento de reflexão. Recomendamos que disponibilize uns 15 minutos para indagá-los sobre alguns elementos presentes na atividade. A seguir, temos algumas sugestões de perguntas que podem ser feitas, para que aguace a curiosidade dos estudantes.

- Quais padrões ou sequências você consegue identificar na solução realizada pelo robô?
- Como poderia ser a escrita de um pseudocódigo para descrever o mesmo movimento solicitado?
- Como a identificação de padrões pode ajudar a otimizar a solução de um problema?
- Qual é a importância de um comando claro e objetivo?

## Atividade 2 - Algoritmo do café

Na segunda atividade, não é amplamente documentado na literatura acadêmica, mas é uma técnica utilizada por profissionais da computação, no entanto, é possível associá-lo ao Edsger Dijkstra pelo estilo de ensino em que utiliza analogias para explicar problemas da computação e pela organização dos processos lógicos, como, por exemplo, os seus famosos problemas dos filósofos jantando, ou o problema do barbeiro dorminhoco (APT, 2022).

Sendo assim, os estudantes são convidados a montar um passo a passo do ato de fazer um café. Diferente da anterior, será pedido que individualmente, cada estudante crie as instruções necessárias para que um robô consiga realizar essa tarefa. Indicamos que reforce a definição do que é um algoritmo para que eles compreendam a importância de realizar uma boa sequência de passos lógicos com instruções claras.

O objetivo desta atividade é que os estudantes aprendam a planejar de forma estruturada uma tarefa cotidiana, utilizando conceitos do pensamento computacional (fragmentação em etapas, decisões, repetições e instruções claras). Neste momento, enfatize a importância da ordem dos passos e se não tem nenhum passo essencial faltando. Esta atividade servirá para aproxima-los ainda mais da escrita, mas caso não seja possível a escrita em pseudocódigo, pode ser feita utilizando a escrita materna.

A seguir, temos dois algoritmos diferentes para a atividade.

ALGORITMO	ALGORITMO
<pre> INICIAR FAZER_CAFE_NA_CAFETEIRA  // Passo 1: Verificar recursos disponíveis SE NÃO HÁ ÁGUA ENTAO     ADICIONE ÁGUA NO RESERVATÓRIO FIM SE  SE NÃO HÁ PÓ_DE_CAFE     ADICIONAR PÓ DE CAFÉ NO FILTRO FIM SE  SE CAFETEIRA ESTÁ SUJA     ENTAO LIMPE A CAFETEIRA FIM SE  // Passo 2: Preparação LIGAR CAFETEIRA ESPERAR 5 MINUTOS  //Passo 3: Verificar se o café ficou pronto SE CAFÉ PRONTO     ENTAO SERVIR_CAFE(Xícara)     DESLIGAR CAFETEIRA SENÃO EXIBIR "Erro: Algo deu errado! </pre>	<pre> INICIAR FAZER_CAFE_NO_FOGAO  // Passo 1: Adicionar água à panela ADICIONAR ÁGUA(500ml)  //Passo 2: Ligar o fogão e aquecer a água LIGAR FOGÃO ESPERAR ATÉ ÁGUA FERVER  // Passo 3: Adicionar pó de café ao filtro COLOCAR FILTRO(Porta-filtro) EXIBIR "Adicionando pó de café ao filtro." ADICIONAR PO_DE_CAFE(3 colheres)  // Passo 4: Coar o café DERRAMAR_AQUA(Filtro, Lento)  // Passo 5: Servir o café SERVIR_CAFE(Xícara)  // Passo 6: Desligar o fogão e limpar DESLIGAR FOGÃO LIMPAR_PANELA() </pre>

Verifique a cafeteira."  
FIM SE  
TERMINAR ALGORITMO

LIMPAR\_FILTRO()  
TERMINAR ALGORITMO

### 3.3.2. Depois das atividades

Depois da atividade 2, reúna a turma para um momento reflexivo. Retome sobre a importância da construção de instruções claras e utilize momentos em que tiveram instruções equivocadas, como, por exemplo, confundir a sua esquerda com a esquerda do robô, pedir para fazer uma ação diferente da esperada etc.

Para este momento, sugerimos as seguintes perguntas:

- O que aconteceria se nos esquecêssemos de colocar o passo de adicionar água no pseudocódigo?
- O que aconteceria se colocássemos “ligar o fogão” antes de colocar a água?
- Como podemos melhorar este algoritmo para que ele funcione para qualquer pessoa ou robô?

Estas perguntas têm como objetivo incentivar os estudantes a pensarem como programadores, garantindo que entendam a importância de um algoritmo claro, organizado e lógico.

## 3.4. Aula 2

Nesta aula, o foco será escrita algorítmica para problemas matemáticos. A unidade temática trabalhada será a geometria, mas o(a) professor(a) pode adaptar para que seus estudantes aprendam a fazer outros algoritmos com problemas de outras unidades

temáticas. É indicado que os conteúdos geométricos referentes ao cálculo de área já tenham sido trabalhados nas aulas anteriores.

### 3.4.1. Antes das atividades

Neste momento, comece a aula apresentando os elementos da escrita em pseudocódigo. Se possível, pegue um exemplo de solução que eles já fizeram nas aulas de Matemática que envolva Geometria e mostre como seria a escrita desta solução neste tipo de linguagem.

Sugerimos o seguinte exemplo.

O problema	
Encontrar a área de um trapézio em que possui base maior igual a 8m, base menor igual a 5m e altura igual a 3m.	
Solução matemática	Solução pseudocódigo
<p>Fórmula para calcular a área de um trapézio:</p> $A = \frac{(B+b) \cdot h}{2}$ <p>sendo, B = 8, b = 5 e h = 3.</p> <p>Substituindo os valores, temos</p> $A = \frac{(8+5) \cdot 3}{2}$ $A = \frac{13 \cdot 3}{2}$ $A = \frac{39}{2}$ $A = 19,5 \text{ m}^2$ <p>Resposta</p> <p>A área do trapézio será <math>19,5 \text{ m}^2</math></p>	<pre> INICIAR CALCULAR_AREA_TRAPEZIO ALGORITMO  // Entrada de dados EXIBIR "Digite a base maior do trapézio:" LER base_maior  EXIBIR "Digite a base menor do trapézio:" LER base_menor  EXIBIR "Digite a altura do trapézio:" LER altura  // Cálculo da área area ← ((base_maior + base_menor) * altura) / 2  // Saída do resultado EXIBIR "A área do trapézio é:", area, "m<sup>2</sup>"  TERMINAR ALGORITMO </pre>

O(a) professor(a) pode utilizar outras situações em que os estudantes possam comparar as duas soluções. Além disso, caso queira revisar os conceitos da Geometria necessários para a resolução das próximas atividades, pode ser interessante fazer uma breve retomada antes de iniciar as atividades.

#### 3.4.2. Durante as atividades

Disponibilize a folha com os problemas impressos ou compartilhe-os por meio de um projetor para que os estudantes possam visualizar as atividades. Caso seja possível, disponibilize papel quadriculado para que os estudantes possam simular o Teorema de Pick. Também será disponibilizado um modelo inicial<sup>3</sup> de algoritmo para os alunos que tiverem dificuldade em como começar.

#### Atividade 3 - Cálculo de área de polígonos regulares

Agora os estudantes montarão um algoritmo em que resolva o problema da área de polígonos regulares por meio da apótema, caso seja necessário, disponibilize a relação de apótema com o cálculo da área. É indicado que ao longo do desenvolvimento deles, seja perguntado sobre as características de um programa, o que deve ser a variável, o que deve ser a instrução por exemplo.

A seguir disponibilizamos sugestões de perguntas para a mediação:

- Como estruturar o passo a passo deste algoritmo?
- Como calcular o perímetro de um polígono regular?
- Quais entradas o programa precisa receber?
- Quais são as variáveis no seu código? Ex: apótema e o perímetro.
- A sequência de passos desenvolvidos pode ser ajustada?
- Quais são as atribuições das variáveis? Elas possuem atribuições? Ex: inteiro, decimal.

---

<sup>3</sup> Está disponível na página 25 deste documento.

Apresentamos um exemplo de solução para o problema da atividade.

```
INICIAR ALGORITMO CALCULAR_AREA_POLIGONO
```

```
//Passo 1: Definir as variáveis
```

```
VAR
```

```
    apotema: real
```

```
    perimetro: real
```

```
// Passo 2: Entrada de dados
```

```
EXIBIR "Digite o valor do perímetro do polígono:"
```

```
LER perimetro
```

```
EXIBIR "Digite o valor da apotema do polígono:"
```

```
LER apotema
```

```
// Passo 3: Cálculo da área
```

```
area ← (perimetro * apotema) / 2
```

```
// Passo 4: Exibir o resultado
```

```
EXIBIR "A área do polígono é:", area, "unidades de área"
```

```
TERMINAR ALGORITMO
```

#### Atividade 4 - Cálculo de área de um polígono qualquer com o Teorema de Pick

Por último, os estudantes montarão um algoritmo que resolva o problema de área de polígonos quaisquer, cujos vértices estejam sobre uma grade de coordenadas inteiras, usando o Teorema de Pick. O objetivo é que os estudantes implementem um programa que receba os valores de fronteira e interiores, e encontre a área do polígono.

A seguir, disponibilizamos sugestões de perguntas para a mediação:

- Quais são as instruções presentes no seu algoritmo? Ex: print e o cálculo de área
- A execução sequencial está adequada?

- O que acontece se mudarmos a forma do polígono?

Apresentamos um exemplo de solução para o problema da atividade.

```
INICIAR ALGORITMO CALCULAR_AREA_POLIGONO_PICK

// Passo 1: Entrada de dados
EXIBIR "Digite o número de pontos internos (I):"
LER pontos_internos

EXIBIR "Digite o número de pontos na fronteira (F):"
LER pontos_fronteira

// Passo 2: Cálculo da área usando o Teorema de Pick
area ← pontos_internos + (pontos_fronteira / 2) - 1

// Passo 3: Exibir o resultado
EXIBIR "A área do polígono é:", area, "unidades de área"

TERMINAR ALGORITMO
```

### 3.4.3. Depois das atividades

Depois da aplicação das atividades, reserve um momento de discussão sobre o que foi aprendido.

A seguir, disponibilizamos algumas perguntas que podem ajudar.

- O que foi fácil e o que foi mais difícil na atividade?
- Os algoritmos poderiam ser aplicados em outros problemas matemáticos?
- Existem formas de otimizar os cálculos?

Caso exista a possibilidade, para fortalecer a interdisciplinaridade das atividades, sugerimos a participação de um professor de informática para a reprodução desses algoritmos em alguma linguagem específica, permitindo aos alunos simularem os códigos e aperfeiçoarem suas soluções por meio dos resultados.

## 4. Códigos em Python

Caro(a) professor(a), caso você tenha disponibilidade em utilizar ou mostrar as calculadoras funcionando em algum compilador<sup>4</sup> para os seus estudantes, disponibilizamos o código das atividades a seguir em Python:

### 4.1. Atividade dos polígonos regulares

```
apotema = input ("Coloque aqui o valor da apótema:")

apotema = float (apotema)

perimetro = input ("Coloque aqui o valor do perímetro:")

perimetro = float (perimetro)

area = (perimetro * apotema)/2

print ("A área do polígono é:", area)
```

### 4.2. Atividade do Teorema de Pick

```
fronteira = input ("Coloque aqui a quantidade de pontos que estão na fronteira do
polígono:")

fronteira = int (fronteira)

internos = input("Coloque aqui a quantidade de pontos internos do polígono:")

internos = int (internos)

area = internos +(fronteira/2)- 1
```

<sup>4</sup> Indicamos o seguinte compilador online: [https://www.tutorialspoint.com/online\\_python\\_compiler.php](https://www.tutorialspoint.com/online_python_compiler.php)

```
print ("A área do polígono é:", área)
```

Nome:

Data:

Turma:

## Aula 1

### Atividade 2 - Algoritmo do café

Desenvolva um algoritmo escrito em pseudocódigo ou em língua cotidiana que represente o processo de preparo de um café. O algoritmo deve seguir uma sequência lógica de passos, desde a obtenção dos ingredientes até a finalização da bebida pronta para consumo.

Nome:

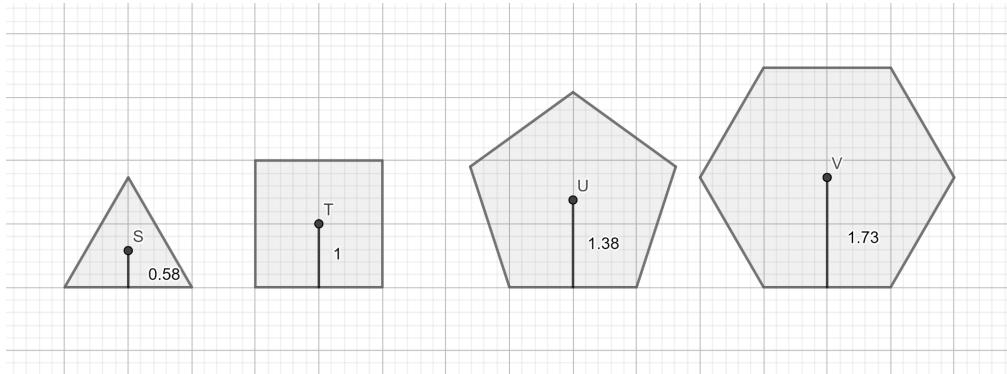
Data:

Turma:

## Aula 2

### Atividade 3 - Cálculo de área de polígonos regulares

Retomando aos conceitos iniciais de geometria, desenvolva um algoritmo que encontre o valor da área de qualquer polígono regular.



Instrução para os valores Entrada e de Saída:

- A entrada deve conter a apótema e o perímetro do polígono desejado.
- Na saída, apresente a mensagem "A área do polígono é" seguida do valor da variável.

### Atividade 4 - Cálculo de área de um polígono qualquer com o Teorema de Pick

Para o cálculo de área de um polígono qualquer, uma possível solução seria o uso do Teorema de Pick. Considere um polígono  $P$  no plano cartesiano, se os vértices de  $P$  têm todos coordenadas inteiras, então a fórmula de pick para a sua área será:  $A = i + \frac{f}{2} - 1$ , em que  $i$  e  $f$  representam a quantidade de pontos com coordenadas inteiras no interior e na aresta do polígono respectivamente. Tomando como base o teorema enunciado, desenvolva um algoritmo que retorne a área das figuras a seguir.

Instrução para os valores Entrada e de Saída:

- A entrada deve conter apenas os pontos de fronteira e os pontos internos;
- Na saída apresente a mensagem "O valor da área do polígono é" seguida do valor da variável.



## Modelo inicial de algoritmo

```
INICIAR ALGORITMO NOME_DO_ALGORITMO

// Passo 1: Entrada de dados

EXIBIR "Mensagem para o usuário solicitando dados"

LER variáveis_necessárias

// Passo 2: Processamento (Cálculos, Condições, Laços, etc.)

REALIZAR operações necessárias

SE condição_então
    FAZER algo
SENÃO
    FAZER outra coisa
FIM_SE

PARA cada_item DE lista FAZER
    EXECUTAR ação
FIM_PARA

ENQUANTO condição FOR verdadeira FAZER
    EXECUTAR ação
FIM_ENQUANTO

// Passo 3: Saída de dados

EXIBIR "Resultado:", resultado

TERMINAR ALGORITMO
```

## 5. Referências

APT, Krzysztof R. Edsger Dijkstra, The Man Who Carried Computer Science on His Shoulders. In: Edsger Wybe Dijkstra: His Life, Work, and Legacy. 2022. p. 373-398. Disponível em: <https://arxiv.org/pdf/2010.00506.pdf> Acesso em: 15 de janeiro de 2025.

BRACKMANN, CHRISTIAN PUHLMANN. DESENVOLVIMENTO DO PENSAMENTO COMPUTACIONAL ATRAVÉS DE ATIVIDADES DESPLUGADAS NA EDUCAÇÃO BÁSICA/ CHRISTIAN PUHLMANN BRACKMANN. – 2017.

BRACKMANN, Christian. AlgoCards. Disponível em: <[www.computacional.com.br](http://www.computacional.com.br)>. 2019. Acesso em: 20 de fevereiro de 2025.

CGI.br/NIC.br, Centro Regional de Estudos para o Desenvolvimento da Sociedade da Informação (Cetic.br), Pesquisa sobre o uso das tecnologias de informação e comunicação nas escolas brasileiras - TIC Educação 2020

DIAS, Marcelo de Oliveira. Reformas curriculares recentes no Brasil e em Portugal: processos, tensões e perspectivas sobre a adoção de tecnologias no ensino de matemática. Pedro & João Editores, 2021. 131p. Disponível em: <https://pedroejoaoeditores.com.br/produto/reformas-curriculares-recentes-no-brasil-e-em-portugal-processos-tensoes-e-perspectivas-sobre-a-adocao-de-tecnologias-no-ensi-no-de-matematica/> Acesso em: 1 de junho de 2024.

MANZANO, José Augusto Navarro Garcia. Estudo Dirigido: algoritmos - Editora Érica, 2000.

MANZANO, José Augusto Navarro Garcia. Algoritmos: lógica para desenvolvimento de programação de computadores / José Augusto N. G. Manzano, Jayr Figueiredo de Oliveira. - 29. ed. - São Paulo: Érica, 2019.