
DESENVOLVIMENTO DE UMA FERRAMENTA EDUCACIONAL PARA O ENSINO DE GEOMETRIA PLANA

DEVELOPMENT OF AN EDUCATIONALA TOOL FOR THE TEACHING OF FLAT GEOMETRY

João Paulo Antunes, CARVALHO¹
Josué Antunes de, MACÊDO²

Resumo

O ensino da Matemática nas escolas enfrenta um grande desafio que é o desinteresse por parte dos estudantes que muitas vezes não encontram estímulos para se dedicar ao aprendizado, já que é uma disciplina com conteúdo considerado abstrato. Para minimizar essa situação, o uso das tecnologias digitais como auxiliar no processo de ensino e aprendizagem está sendo algo crescente nos dias atuais. Nesse contexto, esse trabalho apresenta a construção de uma ferramenta educacional em auxílio ao processo de ensino e aprendizagem de geometria plana. Sua funcionalidade consiste em desenhar figuras geométricas tais como quadrados, hexágonos retângulos circunferência entre outras. Além disso, a ferramenta desenvolvida é capaz de calcular área e perímetro de polígonos regulares. Seu potencial está na precisão nos desenhos. Foi utilizada a placa eletrônica denominada Arduino e alguns de seus periféricos, que tem como característica ser de baixo custo, permitindo aprender na prática, aproximando assim a Matemática da sua realidade. Sua programação é acessível para aqueles sem conhecimento de informática, pois utiliza a linguagem C++. Após um primeiro contato com os conceitos básicos já é possível programar a ferramenta para desenhar figuras geométricas. No desenvolvimento desta pesquisa também foi ministrado uma oficina para acadêmicos da licenciatura, capacitando-os a desenvolver suas próprias ferramentas em auxílio ao processo de ensino e aprendizagem. Além do mais, construiu-se um código único para polígonos regulares facilitando a utilização da ferramenta robótica.

¹ Graduando em Matemática pelo Instituto Federal de Educação, Ciência e Tecnologia do Norte de Minas Gerais (IFNMG); Januária (MG); Brasil. Email: carvalhojoao.jp@gmail.com; Orcid: <http://orcid.org/0000-0001-6443-6207>

² Doutor em Ensino de Ciências e Matemática (UNICSUL). Professor e Pesquisador do Instituto Federal do Norte de Minas Gerais (IFNMG). Professor do Programa de Pós-Graduação em Educação da Universidade Estadual de Montes Claros (UNIMONTES); Montes Claros (MG); Brasil. E-mail: josue.macedo@ifnmg.edu.br; Orcid: <https://orcid.org/0000-0001-7737-7509>.

Palavras-chave: Robótica; Arduino; Ensino e aprendizagem da Matemática

Abstract

The teaching of mathematics in schools faces a big challenge, which is the lack of interest for the part of students who often do not encounter incentives to dedicate themselves to learning, whereas it is a discipline with content considered abstract. In order to minimize this situation, the use of digital technologies as an assistant in teaching and learning process is increasing today. In that context, this work presents the construction of an educational tool to help the teaching and learning process of flat geometry. Its functionality consists in drawing geometric figures such as squares, hexagons, rectangles, circumference, among others. Moreover, the developed tool is able to calculate the area and perimeter of regular polygons. Its potential lies in the accuracy of the drawings. It was used the electronic board named Arduino and some of its peripherals, which is characterized by the low cost, allowing to learn in practice, approaching thereby mathematics to its reality. Its programming is accessible for those without computer knowledge, because it uses the C++ language. After a first contact, it is already possible to program the tool to draw geometric figures. In the development of this research, a workshop was also held for academics of licentiate, enabling them to develop their own tools in assistance to the teaching and learning process. Furthermore, a unique code for regular polygons was built, facilitating the use of the robotic tool.

Key words: Robotics; Arduino; Scratch.

Introdução

A Matemática é vista por alguns estudantes como uma disciplina difícil de se aprender. Conforme Santos, França e Santos (2007), mudar esse pensamento requer experiência do professor, utilizar metodologias que de fato venha contribuir na construção do conhecimento.

Dessa forma, esse trabalho apresenta o resultado de uma pesquisa no qual se desenvolveu uma ferramenta educacional, a fim de auxiliar o professor no processo de ensino e aprendizagem de geometria plana.

Nesse sentido, o objetivo desse trabalho é apresentar a construção de um objeto educacional de baixo custo, permitindo aos professores de Matemática acesso a uma ferramenta que pode contribuir na melhoria de suas aulas.

A ferramenta educacional em questão é caracterizada por um carrinho que pode ser controlado remotamente, cuja funcionalidade é construir figuras geométricas. Por meio de um pincel fixado ao seu centro ele desenha triângulos, pentágonos, circunferências, retângulos entre outros. Além disso é capaz de medir perímetros de

polígonos regulares, como também as áreas dos polígonos cuja quantidade de lados seja par.

Ao longo do desenvolvimento foi elaborada uma oficina e ministrada aos acadêmicos dos cursos de licenciaturas em Física e Matemática do Instituto Federal do Norte de Minas Gerais, Campus Januária, capacitando-os para construção da ferramenta educacional.

O recurso tecnológico foi constituído utilizando a plataforma Arduino e seus periféricos, alimentados por pilhas, com baixo custo de construção e além disso, sua programação é acessível àqueles que não possuem conhecimentos de informática.

A relevância está no fato de contribuir com os professores de Matemática, na medida em que apresenta uma ferramenta inovadora que pode auxiliar no processo de ensino e aprendizagem.

Esse trabalho é composto além dessa introdução de outras quatro seções. A próxima seção trata do *Aporte teórico* que deu base para a construção do objeto educacional. Em seguida os *Procedimentos metodológicos* descrevem detalhadamente a construção da ferramenta. A seção *Resultados e discussão*, aborda alguns resultados obtidos na aplicação do produto educacional. As Considerações finais abordam as contribuições da ferramenta educacional e suas possibilidades no ensino de geometria plana. O artigo é finalizado com as Referências.

Aporte teórico

O ensino de geometria plana nem sempre é um trabalho fácil para muitos dos professores de Matemática, pois os recursos disponíveis são limitados. Na maioria das vezes recorrem a lista de exercícios e fórmulas prontas, que nem sempre produzem resultados satisfatórios, conforme ressalta Lorenzato (1995).

Teixeira e Mussato (2020), enfatizam que os professores do Ensino Fundamental não dispõem de recursos didáticos eficientes, além de não dominarem o conteúdo de Geometria, de tal forma a promoverem uma transposição didática eficiente em sala de aula. Isso reflete no desempenho dos estudantes.

Macêdo e Pedroso (2020), salientam que nos dias atuais, o professor necessita ser um pesquisador, ou seja, aquele profissional reflexivo, questionador, investigador, que alia o ensino à pesquisa, articula teoria e prática pedagógica, pois este fato estimula a reflexão na ação, tornando o professor um pesquisador na prática.

Nesse aspecto, é relevante fazer uso dos aparatos tecnológicos disponíveis, desde que de fato possa contribuir no processo de ensino e aprendizagem de geometria plana, pois a Base Nacional Comum Curricular (BNCC), ressalta que “a Geometria envolve o estudo de um amplo conjunto de conceitos e procedimentos necessários para resolver problemas do mundo físico e de diferentes áreas do conhecimento” (BRASIL, 2018, p. 271). Assim com a variedades de recursos tecnológicos disponíveis, selecionamos a Plataforma Arduino para ser utilizado em nosso trabalho.

O Arduino³ é um recurso tecnológico lançado em 2005, caracterizado por ser uma plataforma aberta de prototipação eletrônica e de baixo custo. Tem como um grande potencial a facilidade de sua utilização, no qual qualquer pessoa interessada e curiosa pode começar a aprender e desenvolver seus próprios projetos (MCROBERTS, 2015).

Devido ao potencial do Arduino e como também da necessidade de produzir recurso metodológico para os professores de Matemática, pois as aulas tradicionais nem sempre conseguem atrativas aos estudantes, ressalta-se a importância de desenvolver ferramentas auxiliaadoras do processo de ensino e aprendizagem de geometria plana. Nesse sentido, desenvolveu-se uma ferramenta educacional que permite instigar a construção do conhecimento do estudante, pois,

[...] recursos didáticos como malhas quadriculadas, ábacos, jogos, livros, vídeos, calculadoras, planilhas eletrônicas e *softwares* de geometria dinâmica têm um papel essencial para a compreensão e utilização das noções matemáticas. Entretanto, esses materiais precisam estar integrados a situações que levem à reflexão e à sistematização, para que se inicie um processo de formalização. (BRASIL, 2018, p 276)

Essa ferramenta tem como característica trabalhar os conteúdos da geometria plana de maneira visual, o que para Boaler (2016) é uma abordagem eficaz, pois quando se trata o conteúdo dessa forma, os estudantes têm uma compreensão mais profunda sobre o assunto.

Uma segunda característica dessa ferramenta educacional, é que ela pode ser operada pelo próprio estudante, desde que tenha um primeiro contato com a ferramenta, com intuito de se familiarizar com os conceitos básicos de programação, ou seja, o aparato tecnológico se torna manipulável.

³ O site oficial dos desenvolvedores do Arduino está disponível em: <https://www.arduino.cc/>. Acesso em 31 out. 2020.

O Arduino já está em ambientes escolares, sendo que o estudante é o agente construtor do seu próprio conhecimento.

A Robótica Pedagógica ou Robótica Educacional (RE) vem se constituindo numa forma interdisciplinar de promoção do aprendizado de conceitos curriculares. Na aula com RE o aluno pensa, manuseia, constrói, executa, vê o que dá certo, depura o que está errado e reexecuta, ou seja, é o esmiuçar da teoria através da prática. (ALVES, 2013, p.163.)

A ferramenta robótica construída com Arduino tem um grande potencial, pois torna o conteúdo de geometria ilustrativo. O Arduino é uma plataforma eletrônica de prototipagem de fácil uso em termos de *Hardware* e *Software*.

O Arduino é facilmente programável e pode ser utilizado para automação de dispositivos eletrônicos, acionamento de motores e *leds*, monitoramento de sensores, construção de protótipos de soluções tecnológicas e um mundo de possibilidades (CANTU; SANTOS, 2013, p. 811).

Devido às suas funcionalidades, o Arduino tem se tornado uma alternativa cada vez mais viável em projetos de prototipagem, em áreas residenciais, escolares e até em empresas. Seu custo financeiro e a possibilidade de trabalhar com a plataforma sem muito conhecimento prévio, são as suas maiores vantagens. A Fig. 1 mostra uma placa de um dos modelos do Arduino.

Figura 1 – Placa do Arduino Uno



Fonte: www.arduino.cc

Segundo Baião (2016), o Arduino surgiu em 2005 em Ivrea, Itália, no instituto de *Design* e integração da cidade, com o objetivo de incentivar o uso das tecnologias e com um custo inferior. Por sua popularidade e por ser uma plataforma livre, é muito fácil atualmente encontrar 'placas-clone', que são versões do Arduino criado por diversas pessoas.

Desde o início do Arduino Project, em 2005, mais de 500 mil placas Arduino foram vendidas no mundo todo. Sem dúvida, o número de placas-clone não oficiais supera o de placas oficiais, sendo provável que mais de um milhão de placas Arduino e suas variantes tenham sido comercializadas (MCROBERTS, 2015, p. 24).

Por ser tão popular, o material de estudo sobre Arduino é abundante. Possui também uma comunidade ativa que compartilha projetos e ideias. A placa Arduino consiste em um conjunto de *hardware* e *software* integrado que pode ser modificado. A placa utiliza a linguagem *processing*, que é baseada na linguagem C/C++.

A linguagem de programação *Processing*⁴ foi criada pelo Instituto de Tecnologia de Massachusetts (MIT), sendo inicialmente concebida para aprender como codificar dentro do contexto das artes visuais. Trata-se de uma biblioteca gráfica de código aberto e ambiente de desenvolvimento integrado (IDE) construída para as artes eletrônicas, novas mídias artísticas e comunidades de *design* visual com o objetivo de ensinar a não programadores os fundamentos da programação de computador em um contexto visual. É utilizada no Arduino por ser uma linguagem de fácil acesso.

A linguagem C foi uma das primeiras linguagens de programação, servindo de base para a criação de muitas outras. É usada hoje principalmente para ensino e também em sistemas mais antigos. Já o C++ é uma biblioteca da linguagem C, trazendo recursos novos, sendo considerada uma evolução do C (JOHANN; SANTOS, 2004).

A próxima seção trata da metodologia utilizada no desenvolvimento da ferramenta educacional descrita nesse trabalho.

Procedimentos metodológicos

A Ferramenta foi desenvolvida por dois acadêmicos bolsistas dos Programas Institucionais de Bolsas de Iniciação Científica e Tecnológica, com apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), com auxílio de acadêmicos voluntários, sobre a orientação de um professor do Instituto Federal de Educação, Ciência e Tecnologia do Norte de Minas Gerais (IFNMG), Campus Januária.

No primeiro momento foi realizada uma pesquisa biobibliográfica, com intuito de identificar os materiais necessários para a construção da ferramenta educacional, como também sobre a utilização de sua programação em linguagem C++.

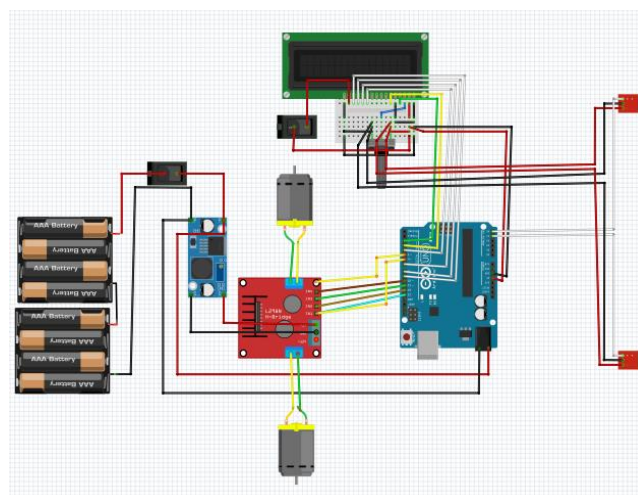
⁴ Site dos desenvolvedores: <https://processing.org/>. Acesso em 31 out. 2020.

Para a montagem da ferramenta educacional foram utilizados os seguintes itens: Arduino modelo uno, dois motores de corrente contínua (CC), modelo N20 6 volts 30 rotações por minuto (RPM), duas rodas 34 mm específico para os motores, roda boba esférica, ponte H L298N, regulador de tensão LM2596, duas chaves de duas posições, 8 pilhas AA, dois suportes para 4 pilhas, uma chapa MDF branca 3 mm, display LCD 16x2, potenciômetro 10kΩ, Protoboard 170 pontos, barra de pinos, dois sensores seguidor de linha, cabo *jumper* e pincel.

O segundo passo foi iniciar a montagem da ferramenta, no qual o MDF depois de cortado de forma retangular, foi utilizado como estrutura principal, recebendo o Arduino e seus periféricos.

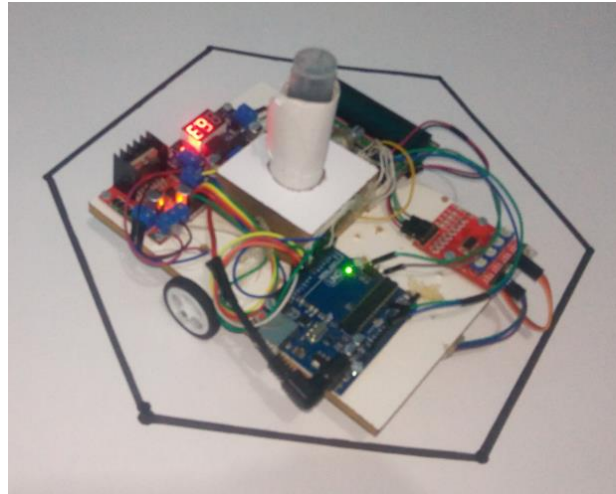
Próximo ao centro do MDF foi realizado um furo para receber o pincel marcador, que serviu também de suporte para o eixo imaginário dos motores. A disposição de montagem do circuito elétrico pode ser vista na Fig. 2

Figura 2 – Circuito elétrico



Fonte: Arquivo Pessoal

A ferramenta educacional pode ser vista na Fig. 3, com todos os componentes relatados anteriormente.

Figura 3 – Ferramenta educacional

Fonte: Arquivo Pessoal

Na terceira etapa, realizou-se a elaboração e programação dos códigos no *software* Arduino. Foram desenvolvidos dois códigos, dos quais um foi aplicado na construção de polígonos regulares e o segundo para o cálculo de áreas e perímetros.

A construção do código de polígonos regulares começou a partir de sua própria definição. De acordo com Dolce e Pompeo (2019), um polígono diz-se regular se tiver todos os seus lados iguais e todos os seus ângulos iguais. Partindo-se dessa informação, construiu-se um código único para todos os polígonos regulares, que pode ser visto na Fig. 4. O que muda de um polígono para outro é o ângulo externo, quantidade de lados e o seu tamanho.

Para a elaboração desse código foi necessário o desenvolvimento de duas funções afins, sendo que a primeira relaciona a distância do seguimento com o tempo necessário a ser inserida no *software* que é caracterizado por uma equação do primeiro grau. O processo para se chegar a essa função, se deu em atribuir um tempo aleatório e em seguida medir a distância percorrida, repetir o processo pela segunda vez com outro tempo diferente do primeiro. Assim usando os conceitos da matemática elementar, encontrou-se uma função afim a partir de dois pontos conhecidos. Chegou-se assim a uma função que relaciona a distância do seguimento com o tempo necessário para percorrer, que é dada pela Eq. (1):

$$t = 258,06x + 32 \quad (1)$$

na qual x é a distância em centímetros e t é o tempo em milésimo de segundo necessário a ser inserido no *software*

A Segunda função afim, relaciona o giro em graus ao tempo, que está representada pela Eq. 2:

$$t = 32,66 * x + 67,31 \quad (2)$$

em que x é o ângulo em graus desejado e t é o tempo em milésimo de segundo.

O processo de elaboração da Eq. 2 se deu de forma idêntica à Eq. 1, em que atribuímos um tempo qualquer, e medimos quantos graus ele girou, da mesma forma, repetindo o processo pela segunda vez com outro tempo, diferente do primeiro. Contudo essa função não permite construir alguns polígonos com uma precisão de 100,0%, mas os resultados obtidos são muito bons, servindo perfeitamente para as atividades didáticas.

Tomou-se a Eq. 2 como fonte norteadora para a elaboração do banco de dados dentro do código. Esse banco contém o tempo necessário para realizar o giro com precisão, relacionados aos ângulos em graus, especificamente para cada polígono regular que se deseja construir.

Figura 4 – Código para polígonos regulares

```

#define MOTOR_DIREITO_FRENTE 12
#define MOTOR_DIREITO_TRAZ 13
#define MOTOR_ESQUERDO_FRENTE 10
#define MOTOR_ESQUERDO_TRAZ 11
#define VELOCIDADE_DO_MOTOR_DIREITO 5
#define VELOCIDADE_DO_MOTOR_ESQUERDO 6

int lados3 =3700; // 120 GRAUS
int lados4 =2755; // 90 GRAUS
int lados5 = 2230; // 72 GRAUS
int lados6 =1845 ; // 60 GRAUS
int lados7 = 1570; // 51 GRAUS
int lados8 = 1380; // 45 GRAUS
int lados9 = 1220; // 40 GRAUS

int vezes = 1;
int TAMANHO_DO_SEGUIIMENTO_EM_CM = 15;
int QUANTIDADE_DE_LADOS_DO_POLIGONO =4;
int VALOR_DO_ANGULO_EXTERNO= lados4 ;

void setup(){
  pinMode(MOTOR_DIREITO_FRENTE, OUTPUT);
  pinMode(MOTOR_DIREITO_TRAZ, OUTPUT);
  pinMode(MOTOR_ESQUERDO_FRENTE, OUTPUT);
  pinMode(MOTOR_ESQUERDO_TRAZ, OUTPUT);
  pinMode(VELOCIDADE_DO_MOTOR_DIREITO, OUTPUT);
  pinMode(VELOCIDADE_DO_MOTOR_ESQUERDO, OUTPUT);
}

void loop() { while(vezes<=QUANTIDADE DE LADOS DO POLIGONO){
  analogWrite(VELOCIDADE_DO_MOTOR_DIREITO,255);
  analogWrite(VELOCIDADE_DO_MOTOR_ESQUERDO,255);
  digitalWrite(MOTOR_DIREITO_FRENTE,HIGH);
  digitalWrite(MOTOR_DIREITO_TRAZ,LOW);
  digitalWrite(MOTOR_ESQUERDO_FRENTE,HIGH);
  digitalWrite(MOTOR_ESQUERDO_TRAZ,LOW);
  delay(258*(TAMANHO_DO_SEGUIIMENTO_EM_CM)+33.54);

  analogWrite(VELOCIDADE_DO_MOTOR_DIREITO,255);
  analogWrite(VELOCIDADE_DO_MOTOR_ESQUERDO,255);
  digitalWrite(MOTOR_DIREITO_FRENTE,HIGH);
  digitalWrite(MOTOR_DIREITO_TRAZ,LOW);
  digitalWrite(MOTOR_ESQUERDO_FRENTE,LOW);
  digitalWrite(MOTOR_ESQUERDO_TRAZ,HIGH);
  delay(VALOR_DO_ANGULO_EXTERNO);

  analogWrite(VELOCIDADE_DO_MOTOR_DIREITO,0);
  analogWrite(VELOCIDADE_DO_MOTOR_ESQUERDO,0);
  vezes++;
}}

```

banco de dados

caixa de entrada

controla a repetição, de acordo com o número de lados do polígono

↑

siga para frente

giro em seu próprio centro

Fonte: Arquivo Pessoal

Foi elaborado uma caixa de entrada, no qual são inseridos os dados do polígono regular que se deseja construir, tais como quantidade de lados e tamanho

do seguimento em centímetros. O ângulo já está no banco de dados, que se altera de acordo com o número de lados.

O segundo código foi desenvolvido para o cálculo de área e perímetro. Sua lógica inicial se caracteriza por um carrinho seguidor de linha, no qual tem a capacidade de seguir por cima de uma linha preta (fita isolante ou pincel grosso), por meio do uso de sensores de infravermelho que identifica a linha. Partindo dessa característica, foi incrementado alguns termos tais como um contador de tempo, tal recurso foi utilizado para contabilizar o tempo gasto ao contornar uma figura geométrica, em seguida, implementou-se a função inversa da primeira função afim, dada na Eq. 1 por $t = 258,06x + 32$. Assim após alguns testes na prática, realizando os ajustes matemáticos, obteve-se como resultado a Eq. 3:

$$x = 0.004t - 0.5 \quad (3)$$

Além disso, foi necessário fazer uma subtração do tempo gasto ao contornar os vértices, que é um valor constante, devido ao fato que a soma dos ângulos externos de um polígono regular é 360 graus. Assim após várias observações e análises, chegou-se à conclusão que o tempo gasto nos vértices e foi de aproximadamente 1000 milésimo de segundos.

Logo, para esse modelo a equação ficou como a expressa na Eq. 4:

$$x = 0.004(t - 1000) - 0.5 \quad (4)$$

Nessa fase, concluiu-se o código do cálculo do perímetro para polígonos regulares. O próximo passo foi implementar o uso de uma fórmula matemática que calcula a área que é apresentada por

$$A = \frac{P \cdot a}{2} \quad (5)$$

em que a área (A) é dada por perímetro (P), vezes a apótema (a), dividido por dois.

Nota se que no caso particular dos polígonos com quantidade de lados pares, a apótema coincide com a metade da distância entre os lados que são paralelos. Assim, o código para o cálculo de área de polígono regular de quantidade de lados pares pode ser visto nas Fig. 5

Figura 5 – Código para cálculo de área e perímetro

<pre>#include <LiquidCrystal.h> LiquidCrystal liquid_crystal(2,3,4,7,8,9); #define motor_direito_frente 13 #define motor_direito_traz 12 #define motor_esquerda_frente 10 #define motor_esquerda_traz 11 #define motor_direito_porta_velocidade 5 #define motor_esquerdo_porta_velocidade 6 int parar_motores = 0; int velocidade_motor_direita = 255; int velocidade_motor_esquerda= 255; signed long int tempo_inicial_perimetro, tempo_inicial_area; #define porta_sensor_direita 14 #define porta_sensor_esquerda 15 bool sensor_direita, sensor_esquerda;</pre>	<pre>void setup() { Serial.begin(9600); pinMode(motor_direito_frente, OUTPUT); pinMode(motor_direito_traz, OUTPUT); pinMode(motor_esquerda_frente, OUTPUT); pinMode(motor_esquerda_traz, OUTPUT); pinMode(motor_direito_porta_velocidade,OUTPUT); pinMode(motor_esquerdo_porta_velocidade,OUTPUT); pinMode(porta_sensor_direita, INPUT); pinMode(porta_sensor_esquerda, INPUT); liquid_crystal.clear(); liquid_crystal.begin(16,2); }</pre>
<pre>void frente (){ analogWrite(motor_direito_porta_velocidade, velocidade_motor_direita); analogWrite(motor_esquerdo_porta_velocidade, velocidade_motor_esquerda); digitalWrite(motor_direito_frente, HIGH); digitalWrite(motor_direito_traz, LOW); digitalWrite(motor_esquerda_frente, HIGH); digitalWrite(motor_esquerda_traz, LOW);} void direita(){ digitalWrite(motor_direito_traz, HIGH); digitalWrite(motor_direito_frente, LOW); digitalWrite(motor_esquerda_frente, HIGH); digitalWrite(motor_esquerda_traz, LOW); analogWrite(motor_direito_porta_velocidade, velocidade_motor_direita); analogWrite(motor_esquerdo_porta_velocidade, velocidade motor esquerda); }</pre>	<pre>void esquerda (){ analogWrite(motor_direito_porta_velocidade, velocidade_motor_direita); analogWrite(motor_esquerdo_porta_velocidade velocidade_motor_esquerda); digitalWrite(motor_direito_frente, HIGH); digitalWrite(motor_direito_traz, LOW); digitalWrite(motor_esquerda_traz, HIGH); digitalWrite(motor_esquerda_frente, LOW);} void parar(){ analogWrite(motor_direito_porta_velocidade, parar_motores); analogWrite(motor_esquerdo_porta_velocidade parar_motores); digitalWrite(motor_direito_frente, LOW); digitalWrite(motor_direito_traz, LOW); digitalWrite(motor_esquerda_frente,LOW); digitalWrite(motor_esquerda_traz, LOW);}</pre>
<pre>bool print_perimetro_check = false; signed long int perimetro; signed long int print_perimetro (signed long int tempo) { if(!print_perimetro_check) { perimetro =0.0086*(tempo-6100); liquid_crystal.setCursor(0,0); liquid_crystal.print(perimetro); liquid_crystal.print(" CM"); liquid_crystal.setCursor(0,1); liquid_crystal.print("Calculando area"); print_perimetro_check = true; delay(10000); }}}</pre>	<pre>bool print_area_check = false; signed long int area; signed long int print_area (signed long int tempob){ if(!print_area_check) { area = ((perimetro)/4)*(0.004*(tempob-9700 -tempo_inicial_perimetro)-1); liquid_crystal.clear(); liquid_crystal.setCursor(0,0); liquid_crystal.print(perimetro); liquid_crystal.print(" CM"); liquid_crystal.setCursor(0,1); liquid_crystal.print(area); liquid_crystal.print(" CM2"); print_area_check = true;}}</pre>

<pre>bool print_perimetro_check = false; signed long int perimetro; signed long int print_perimetro (signed long int tempo) { if(!print_perimetro_check) { perimetro =0.0086*(tempo-6100); liquid_crystal.setCursor(0,0); liquid_crystal.print(perimetro); liquid_crystal.print(" CM"); liquid_crystal.setCursor(0,1); liquid_crystal.print("Calculando area"); print_perimetro_check = true; delay(10000); }} </pre>	<pre>bool print_area_check = false; signed long int area; signed long int print_area (signed long int tempob){ if(!print_area_check) { area = ((perimetro)/4)*(0.004*(tempob-9700 -tempo_inicial_perimetro)-1); liquid_crystal.clear(); liquid_crystal.setCursor(0,0); liquid_crystal.print(perimetro); liquid_crystal.print(" CM"); liquid_crystal.setCursor(0,1); liquid_crystal.print(area); liquid_crystal.print(" CM2"); print_area_check = true;}}</pre>
<pre>void loop() { sensor_direita = digitalRead(porta_sensor_direita); sensor_esquerda = digitalRead(porta_sensor_esquerda); if(!print_perimetro_check) { if(sensor_direita == false && sensor_esquerda == false){ frente (); } else { if(sensor_direita == true && sensor_esquerda == false){ direita(); } else { if(sensor_direita == false && sensor_esquerda == true){ esquerda (); } else { if(sensor_direita == true && sensor_esquerda == true){ parar(); tempo_inicial_perimetro = millis(); print_perimetro(tempo_inicial_perimetro);}}}}</pre>	<pre>} else { if(sensor_direita == false && sensor_esquerda == false){ frente(); } else { parar(); tempo_inicial_area = millis(); print_area(tempo_inicial_area); delay(20000); } } </pre>

Fonte: Arquivo Pessoal

Por último, realizou-se uma oficina com o uso da ferramenta educacional para os acadêmicos dos cursos de Licenciaturas em Matemática e Física, com seis participantes. Foram formados dois grupos com três integrantes cada, para melhor custo benefício, pois os bolsistas e os estudantes da oficina tiveram que arcar com as despesas dos materiais.

A oficina foi dividida em dois momentos. No primeiro realizou-se a montagem da ferramenta, no qual os estudantes eram os agentes construtores. O processo de montagem foi realizado em dois encontros com duração de 1h cada. Os acadêmicos se empenharam bastante, utilizando as ferramentas, tais como furadeira, máquina de solda, pistola de cola quente, cola rápida, estiletes, chaves, como podem ser vistos na Fig. 7.

Figura 7 – Oficina



Fonte: Arquivo Pessoal

O segundo momento dedicou-se à elaboração do código para desenho dos polígonos regulares, como também a utilização de códigos já construídos, tais como o de cálculo de área e perímetro.

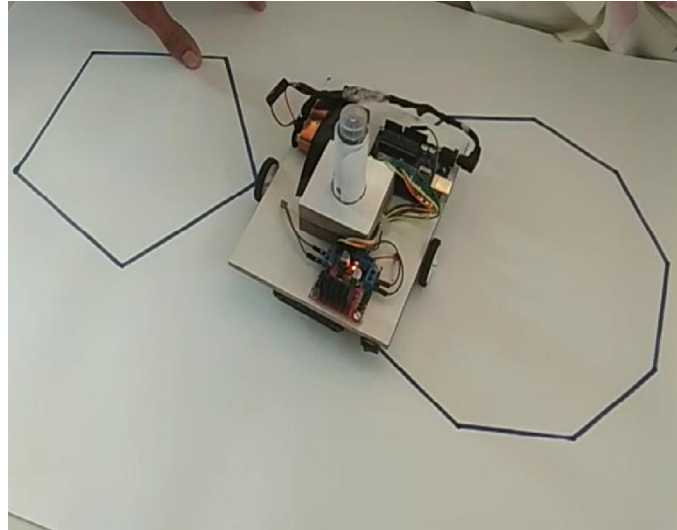
Todo esse processo de construção se inspirou na metodologia de ensino denominada construcionismo (PAPERT, 1986). Essa metodologia se baseia em um aprendizado com uma abordagem mais prática. Os livros e cadernos são deixados em segundo plano para uma abordagem completamente voltada ao objeto de estudo de forma palpável.

Resultados e discussão

A ferramenta educacional, é caracterizada por um carrinho controlado a distância, na qual o professor pode programá-lo para desenhar figuras geométricas. Sua precisão é alta na maioria das figuras desenhadas, mesmo com interferência do

atrito e a variação de energia, ainda assim obtém-se bons resultados como pode ser visto na Fig. 8.

Figura 8 – ferramenta educacional



Fonte: Arquivo Pessoal

O desenvolvimento de um código único para construção de polígonos regulares facilitou a sua utilização durante uma aula de geometria plana, mas a ferramenta não se limitou apenas a esse tipo de figura geométrica. Existe uma infinidade de possibilidades que pode ser desenhada. Um exemplo prático é a Fig. 9. Seu código fonte é simples, pois nessa figura existe um padrão em seus ângulos e seguimentos, o que facilita sua programação. Após uma interação com a programação em linguagem C++ voltada para a ferramenta educacional, outras figuras geométricas podem ser construídas tranquilamente.

Figura 9 – Figura geométrica



Fonte: Arquivo Pessoal

Durante a oficina os estudantes foram os agentes construtores da ferramenta. Eles mesmo fizeram a montagem, fazendo as devidas ligações do Arduino aos seus respectivos periféricos. Com isso eles tiveram o conhecimento sobre a função de cada módulo utilizado na ferramenta.

No segundo momento obteve-se o primeiro contato com a programação, e logo após já conseguiram realizar a construção de alguns polígonos, como pode ser visto na Fig. 10.

Figura 10 – Oficina



Fonte: Arquivo Pessoal

Além disso observaram a importância de alguns conceitos matemáticos na programação, pois eles tiveram que construir funções afim para ser utilizada na programação de qualquer figura geométrica.

Considerações finais

A ferramenta educacional é de grande potencial, pois tal recurso é algo a mais que o professor pode utilizar para abordar os conteúdos de geometria plana, deixando suas aulas mais interativas aos seus respectivos estudantes. Permite ainda levá-los a construir seus próprios conhecimentos, pois eles podem observar alguns padrões nas figuras desenhadas e tirar suas próprias conclusões orientado por um professor, ou seja, os estudantes são o próprio agente construtor de seu conhecimento, tornando a aula com aprendizado mais relevante.

Desse modo os objetivos foram alcançados, que era construir uma ferramenta tecnológica que de fato venha auxiliar o professor no processo de ensino e aprendizagem de geometria, partindo de recurso tecnológico já existentes e de baixo custo e como também sem precisar de um conhecimento aprofundado na área.

O desenvolvimento dessa ferramenta educacional, realizada com o Arduino, pode ser utilizado para auxiliar o processo de ensino aprendizagem de geometria plana, nos diferentes níveis de ensino, especialmente na educação básica, possibilitando aos estudantes irem além dos resultados obtidos apenas com atividades puramente teóricas.

Com a realização das atividades proporcionadas pela ferramenta educacional, os estudantes podem construir conceitos a partir da visualização e manipulação das figuras construídas, vivenciando a matemática na prática. Assim, entende-se que as potencialidades dessa ferramenta contribuem para que os estudantes estabeleçam conexões entre a construção das figuras e a compreensão das propriedades dessas figuras, indo além das construções puramente com lápis e papel.

O uso de recursos tecnológicos por si só não pode ser considerado uma solução, mas inevitavelmente traz contribuições no processo de ensino aprendizagem, como pode ser observado com a ferramenta educacional apresentada nesse trabalho.

Referências

ALVES, Rafael Machado *et al.* Uso do hardware livre Arduino em ambientes de ensino-aprendizagem. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, p. 162-187, 2013.

BAIÃO, Emerson Rodrigo. **Desenvolvimento de uma metodologia para o uso do Scratch for Arduino no ensino médio**. Campinas, 2016. 101f. Dissertação (Mestrado em Educação). Universidade Estadual de Campinas, Programa de Pós-Graduação em Educação, 2016.

BOALER, Jo et al. Seeing as understanding: The importance of visual mathematics for our brain and learning. **Journal of Applied & Computational Mathematics**, v. 5, n. 5, p. 1-6, 2016.

BRASIL. **Base nacional comum curricular**. Ministério da Educação, Secretaria de Educação Básica. Brasília, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_sit_e.pdf. Acesso em 31 out. 2020.

- CANTÚ, Evandro; SANTOS, Luciano Marcos dos. Usando a linguagem *Scratch* e a plataforma Arduino para implementar uma abordagem metodológica baseada em aprender fazendo. In: Nuevas Ideas en Informática Educativa TISE 2013: Memorias do Congresso Internacional de Informática Educativa, 18, 2013, Porto Alegre, **Anais...** Porto Alegre (RS): Pontifícia Universidade Católica do Rio Grande do Sul – (PUCRS), 2013, p. 811–813.
- DOLCE, Osvaldo; POMPEO, José Nicolau. **Fundamentos de matemática elementar: Geometria plana**. v. 9. 9 ed. Atual Editora: São Paulo, 2019.
- JOHANN, Marcelo de Oliveira; SANTOS, Glauco Borges Valim dos. **Curso de introdução à programação em C++**. Apostila do Curso de Extensão da Universidade Federal do Rio Grande do Sul, Porto Alegre, 2004.
- LORENZATO, Sergio. Por que não ensinar geometria? **A Educação Matemática em Revista**, Ano III, n. 4, 1º semestre, Blumenau: SBEM, 1995
- MACÊDO, Josué Antunes de; PEDROSO, Luciano Soares. Os saberes docentes e a formação para a pesquisa na área de ciências. In.: **Formação inicial e continuada de professores: políticas e desafios**. [org]. Reginaldo Peixoto. 1.ed. Curitiba, PR: Bagai, 2020.
- MCROBERTS, Michael. **Arduino básico**. 2. ed. São Paulo: Novatec Editora, 2015.
- PAPERT, Seymour. **Constructionism**: a new opportunity for elementary science education. Massachusetts Institute of Technology, Media Laboratory, Epistemology and Learning Group, 1986.
- SANTOS, Josiel Almeida; FRANÇA, Kleber Vieira; SANTOS, Lúcia Silveira Brum dos. **Dificuldades na aprendizagem de Matemática**. Monografia de Graduação em Matemática. São Paulo: UNASP, 2007.
- TEIXEIRA, Alcinda Souza Muniz; MUSSATO, Solange. Contribuições do *software* GeoGebra nas aulas com sólidos geométricos de faces planas nos anos iniciais do ensino fundamental. **REAMEC-Rede Amazônica de Educação em Ciências e Matemática**, v. 8, n. 3, p. 449-466, 2020.

Recebido em: 08/04/2020

Aprovado em: 09/11/2020