



PROGRAMA DE MESTRADO PROFISSIONAL EM ENSINO DE CIÊNCIAS E MATEMÁTICA

PRODUTO EDUCACIONAL

Sequência de ensino: ROTEIRO PARA O ENSINO DE ALGUNS
CONCEITOS DE MATRIZES UTILIZANDO O ARDUINO

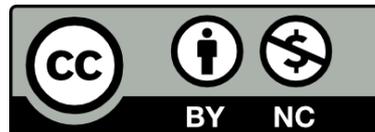
Fábio Anderson de Assumpção Silva

Mariana Pelissari Monteiro Aguiar Baroni

São Paulo (SP)

2017

Este trabalho está licenciado sob uma Licença Creative Commons Atribuição-
NãoComercial 4.0 Internacional. Para ver uma cópia desta licença, visite
<http://creativecommons.org/licenses/by-nc/4.0/>.



Produto Educacional apresentado como requisito à obtenção do grau de Mestre em Ensino de Ciências e Matemática pelo Programa de Mestrado Profissional em Ensino de Ciências e Matemática do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo, campus São Paulo. Defesa realizada em 25/04/2017.

AUTORES

Fábio Anderson de Assumpção Silva: Licenciado em Matemática pela Universidade de São Paulo (USP), Licenciado em Pedagogia pela Faculdade de Pinhais (FAPI), Aperfeiçoamento em Licenciatura em Matemática pela Universidade do Porto (U.PORTO), Especialista no Ensino de Matemática pela Universidade Estadual de Campinas (UNICAMP), Especialização em Educação Infantil pelo Instituto Brasileiro de Formação (IBF), MBA em Gestão Escolar pelo Centro Universitário das Faculdades Metropolitanas Unidas (FMU) e Mestre em Ensino de Ciências e Matemática pelo Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP). Atualmente é Professor do Colégio See Saw Panamby e do Colégio Visconde de Porto.

Mariana Pelissari Monteiro Aguiar Baroni: É Licenciada em Matemática pela Universidade Estadual Paulista (UNESP-2002), Mestre em Computação Aplicada pelo Instituto Nacional de Pesquisas Espaciais (INPE-2005), Doutora em Computação Aplicada pelo Instituto Nacional de Pesquisas Espaciais (INPE-2009) com período sabbático na Université Libre de Bruxelles (ULB) e realizou estágio de pós-doutoramento em Matemática Aplicada na Universidade Federal do ABC (UFABC-2010). Atualmente é professora do Instituto Federal de Educação, Ciência e Tecnologia de São Paulo (IFSP). Tem experiência na área de Matemática Aplicada e Ensino de Matemática, atuando nos seguintes temas: física computacional, simulação e análise de sistemas espaço-temporais, solução numérica das equações diferenciais parciais lineares e não-lineares, análise de padrões espaço-temporais, dinâmica de fluidos em meios porosos, avaliação da aprendizagem e novas tecnologias para o ensino de matemática.

Nível de Ensino:

Médio

Título:

Roteiro para o ensino de alguns conceitos de matrizes utilizando o Arduino

Duração:

1 aula de 50 min para cada atividade aberta

Conteúdo abordado:

Conceitos Iniciais de Matrizes (Linha , coluna, posição, diagonal principal e secundária, entre outros);

Operação de adição/subtração de Matrizes;

Operação de multiplicação de escalar por Matriz;

Operação de multiplicação de Matrizes.

Tipo de Atividade:

Prática, expositiva e em grupo

Material Utilizado (para um kit):

1 Placa Arduino

9 Led's

9 Resistores de 680Ω

Jumpers (diversos)

1 Protoboard

1 Notebook/Computador

1 Braço Robótico de 4 servomotores

Divisão dos Grupos:

Os estudantes devem ser divididos em duplas e cada dupla deve receber um kit com as peças e equipamentos descritos anteriormente. O ideal é que o professor efetue a atividade na sala de informática e que todos os computadores/notebooks já tenham o software do Arduino instalado. Além disso, em algumas atividades, os estudantes são orientados a realizarem pesquisas, e o acesso à internet é fundamental.

Objetivos:

- Fazer com que o estudante aprenda os conceitos iniciais de matrizes;
- Levar o estudante a perceber os conceitos de adição, subtração e multiplicação de matrizes e multiplicação de um escalar por uma matriz através da movimentação do braço mecânico;
- Criar uma relação entre o que é aprendido na escola e o que os estudantes vivenciam;
- Tornar o conteúdo de matrizes prazeroso e contextualizado;
- Tornar esta prática interdisciplinar e lúdica;
- Oportunizar aos estudantes expor suas ideias e opiniões sobre os levantamentos e análises que fizeram da experiência;
- Tornar a aula mais dinâmica e participativa, levando o estudante ao protagonismo do processo;
- Levar o estudante a "fazer" Matemática através da Investigação e Experimentação;
- Uma maior interação entre os estudantes e as NTIC;
- Despertar a curiosidade e o senso investigativo nos estudantes.

A.1. PRÉ-AULA

Para a aplicação das atividades deste roteiro é necessário que o professor se familiarize com o Arduino e com a linguagem C++. Para isso, o professor deve fazer a leitura antecipada deste material por completo, pois o mesmo oferece uma introdução inicial ao Arduino e sua linguagem de programação. A leitura e execução das atividades propiciará a familiarização necessária para a execução das atividades em aula juntamente com os estudantes, lembrando-se que todo o material disponibilizado é *open-source*, ou seja, permite as alterações necessárias por parte do usuário e que “a prática leva a perfeição”.

Após esta familiarização inicial, o professor pode colocar a atividade em ação em sala de aula, dividindo os estudantes de sua turma em duplas e cada dupla tendo um kit com o material necessário para o desenvolvimento da atividade.

Uma possibilidade que pode ser explorada pelo professor também é estimular os estudantes a aprenderem a linguagem de programação C++ durante os meses, ou até mesmo anos, que antecedem a programação de apresentação de conceitos de matrizes. Diversas atividades podem ser encontradas na *internet* sobre o ensino de matemática de diferentes conceitos utilizando linguagem de programação. Tal empenho depende também do trabalho em conjunto dentro da escola, entre pais, professores e a coordenação pedagógica, a fim de que novas tecnologias para o ensino possam ser inseridas no ambiente escolar.

Quaisquer dúvidas sobre a utilização ou aperfeiçoamento que o professor executar, por favor, comunique ao e-mail fabioassumpção@gmail.com, para que assim possamos cada vez mais agregar e compartilhar conhecimentos.

Este roteiro apresenta-se da seguinte forma: na seção A1, o professor tem uma introdução básica sobre o Arduino e sua linguagem de programação com alguns exemplos básicos de programação e montagens de circuitos para o acionamento de *led* e de servomotor; na seção A2 estão as atividades propostas para o ensino dos conceitos iniciais de matrizes juntamente com as orientações necessárias para a execução das atividades; na seção A3 orientamos sobre como avaliar o processo de ensino e aprendizagem e finalizamos na seção A4 com as referências.

A.1.1. Conhecendo o Arduino

Utilizaremos o circuito conhecido como Arduino e sua linguagem de programação para propiciar ao estudante uma aprendizagem dos conceitos de matrizes através da investigação e experimentação matemática. Todas as informações fornecidas aqui foram extraídas da página oficial do Arduino Genuíno (<http://www.arduino.cc>).

O Arduino nasceu no *Interaction Design Institute Ivrea* como uma ferramenta de fácil manuseio para estudantes sem nenhum conhecimento em eletrônica ou programação e sua arquitetura é totalmente *open-source* (plataforma livre que permite ao usuário modificá-la sem prévia autorização do fabricante). Isto faz com que esta plataforma de trabalho cresça através das contribuições de usuários do mundo inteiro, pois a maioria dos usuários disponibilizam os projetos e programações, servindo de base para um novo projeto de outro usuário.

Existem mais de vinte modelos de placas de Arduino, tais como o Arduino Uno, Arduino Mega, e Arduino Nano. A diferença básica entre elas é a quantidade de entradas/saídas de dados, capacidade de armazenamento, *Clock* e o seu tamanho.

Utilizamos uma placa Arduino Uno (Fig. A.1) para a execução das atividades aqui propostas.

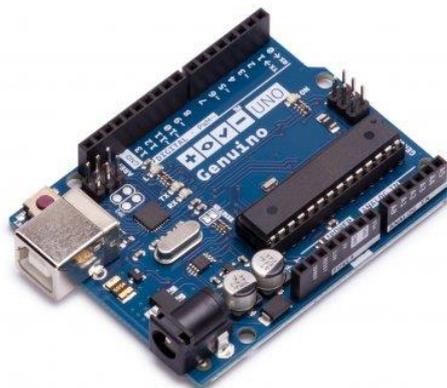


Figura A.1 – Arduino Uno

Fonte: Extraído de Arduino Products (2016).

A Figura A.2 apresenta as portas de acesso ao Arduino Uno, que podem ser utilizadas para as atividades descritas neste roteiro.

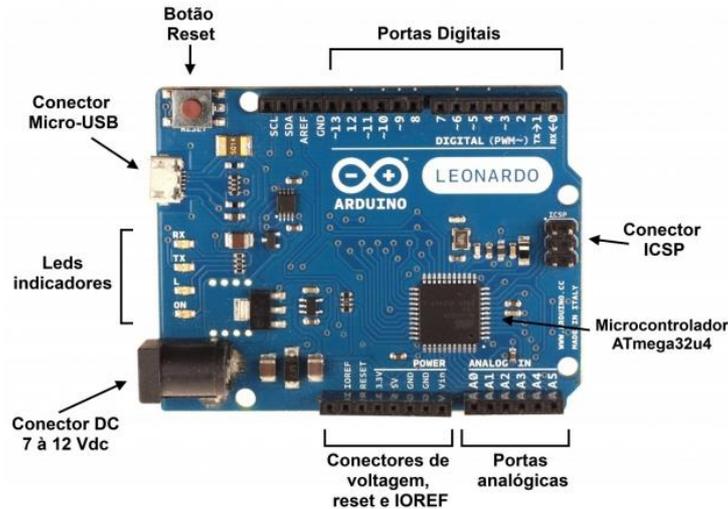


Figura A.2 – Portas de Acesso ao Arduino UNO.
Fonte: Extraído de Blog Filipe Flop (2016).

O Quadro A.1 a seguir, apresenta as especificações técnicas do Arduino UNO.

Quadro A.1 – Especificações Técnicas

Microcontrolador	ATmega328P
Tensão operacional	5V
Tensão de entrada (recomendado)	7-12V
Tensão de entrada (limite)	6-20V
Digital pinos I / O	14 (dos quais 6 fornecer saída PWM)
PWM Digital pinos I / O	6
Pinos de entrada analógica	6
Corrente DC por I / O Pin	20 mA
Corrente DC 3.3V para Pin	50 mA
Memória flash	32 KB (ATmega328P) dos quais 0,5 KB utilizado por bootloader
SRAM	2 KB (ATmega328P)
EEPROM	1 KB (ATmega328P)
Velocidade do relógio	16 MHz
LED_BUILTIN	13
Comprimento	68,6 mm
Largura	53,4 mm
Peso	25 g

Fonte: Extraído de: Arduino Genuino Uno (2016)

A linguagem utilizada na programação do Arduino é a linguagem C++ e o Arduino possui um software próprio que pode ser baixado no site do Arduino Genuino. Segue um roteiro para instalação:

- 1- Acessar o site www.arduino.cc e ir no ícone DOWNLOAD;
- 2- Efetuar o download do programa de acordo com o seu sistema operacional
- 3- Após o download do programa Arduino, executar a instalação do mesmo;
- 4- Com o programa instalado, conecte o computador à placa Arduino com um cabo USB (Fig. A.3);



Figura A.3 – Conexão do Arduino com o computador através do cabo USB.

- 5- Clicar no ícone do Arduino criado na área de trabalho do seu computador e abrirá a tela de início do Arduino (Fig. A.4) onde são digitadas as linhas de comando.

```
string_led_matriz | Arduino 1.6.7
Arquivo Editar Sketch Ferramentas Ajuda
string_led_matriz
int ledPin2 = 2;
int ledPin3 = 3;
int ledPin4 = 4;
int ledPin5 = 5;
int ledPin6 = 6;
int ledPin7 = 7;
int ledPin8 = 8;
int ledPin9 = 9;
int ledPin10 = 10;
String a11 = "a11";
String a12 = "a12";
String a13 = "a13";
String a21 = "a21";
String a22 = "a22";
String a23 = "a23";
String a31 = "a31";
String a32 = "a32";
String a33 = "a33";
String dip = "dip";
```

Figura A.4 – Tela de Início do Arduino.

As duas principais partes (funções) de um programa desenvolvido para o Arduino são:

- `setup()`: onde devem ser definidas algumas configurações iniciais do programa. É executado uma única vez.
- `loop()`: função principal do programa. Fica executando as linhas de comando ininterruptamente, até o usuário finalizar a execução do programa.

Todo programa para o Arduino deve ter estas duas funções.

6- Após efetuar a conexão do Arduino ao computador, basta copiar as linhas de comando do código-fonte e carregar o programa na placa Arduino.

7- Para efetuar o carregamento do programa, devemos clicar no botão  localizado no menu (Figura A.5).

**Para carregar o programa,
clique neste botão**

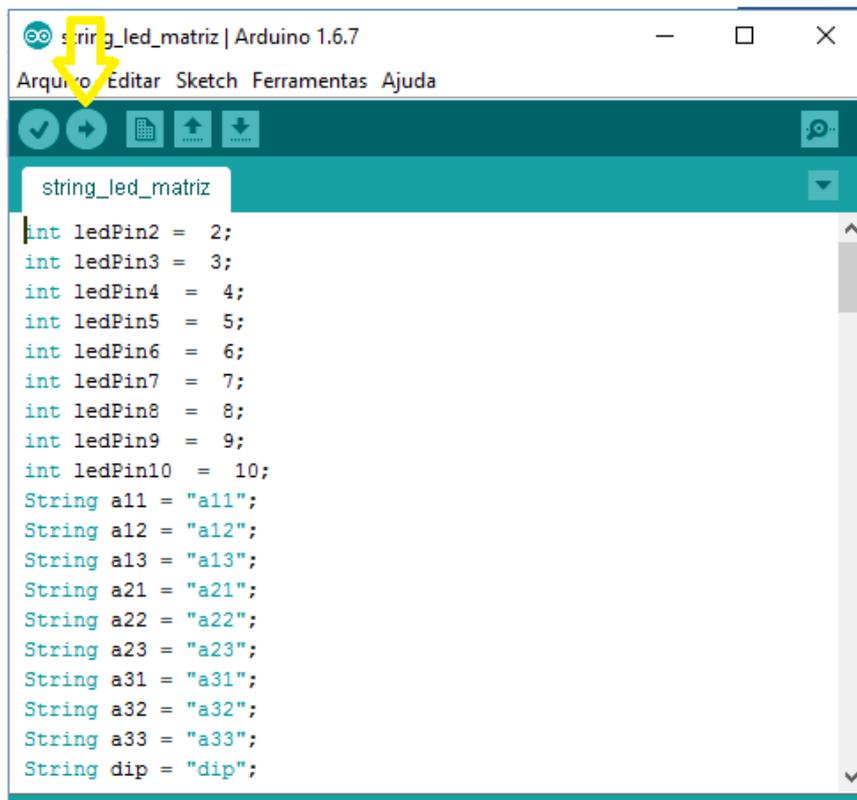


Figura A.5 – Localização de botão de carregamento de programa na tela de início do Arduino.

8- Após este passo, o programa já está pronto para rodar na placa do Arduino.

No sitio do Arduino/Genuino (<http://www.arduino.cc>), podemos buscar todas estas informações como também participar de fóruns sobre programação, dispositivos

e programas já desenvolvidos para esta plataforma, sempre lembrando que todas estas informações são *open-source*, ou seja, os usuários podem utilizá-las livremente e alterá-las como achar necessário.

A.1.2. Código fonte de experimentos com o Arduino

Como dissemos anteriormente, o Arduino tem uma estrutura de Hardware e software *open-source*, o que permite que os usuários disponibilizem os códigos-fontes criados por eles, como também, usufruam de outros códigos-fontes criados e disponibilizados livremente na internet.

No site do fabricante, conseguimos localizar os mais diversos tipos de montagem, desde o princípio básico de funcionamento do Arduino (instalação, configuração, softwares) como também os mais diversos programas e configurações, o que permite a iniciação de qualquer usuário que queira aprender como utilizar o Arduino.

A seguir descrevemos um código fonte básico para o acionamento de um *led* através do Arduino (Fig. A.6) e para a movimentação do braço mecânico (Fig. A.7). Este código é base para os códigos desenvolvidos neste roteiro de atividades, e pode ser adaptado para outras atividades de ensino.

Esta programação para o acionamento de um *led* e acionamento de servo motores, como também diversos outros programas, podem ser encontrados no sítio do Arduino/Genuino.

Código-fonte para acionamento de um led.

```

/* LED piscante
 * -----
 * liga e desliga um LED conectado a um pino digital
 * em intervalos de 2 segundos. Idealmente usa-se o pino 13 da
 * placa do Arduino porque ele já possui um resistor ligado a ele
 *
int ledPin = 13;          // LED conectado ao pino digital 13
void setup()
{
  pinMode(ledPin, OUTPUT); // configura pino digital como saída
}
void loop()
{
  digitalWrite(ledPin, HIGH); // liga o LED
  delay(1000);                // temporiza 1 segundo
  digitalWrite(ledPin, LOW);  // desliga o LED
  delay(1000);                // aguarda mais um segundo

```

```
}
```

Código-fonte para movimentação de um braço mecânico utilizando um servo motor.

```
#include <Servo.h>      // Biblioteca Servo Motor
Servo motor;
void setup()
{
  motor.attach(3);      // Pino 3 saída do comando
}
void loop()
{
  motor.write(160);     // Movimenta o servo motor para a posição de 160°
  delay(1000);         // Aguarda 1000ms
  motor.write(16);     // Movimenta o servo motor para a posição de 16°
  delay(1000);         // Aguarda 1000ms
}
```

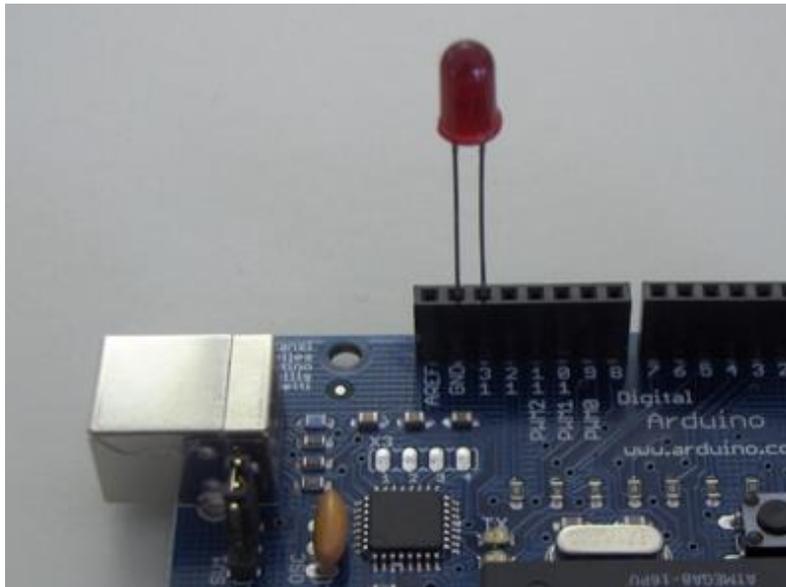


Figura A.6 – Led conectado ao Arduino no pino 13.
Fonte: Extraído de Arduino Products (2016).

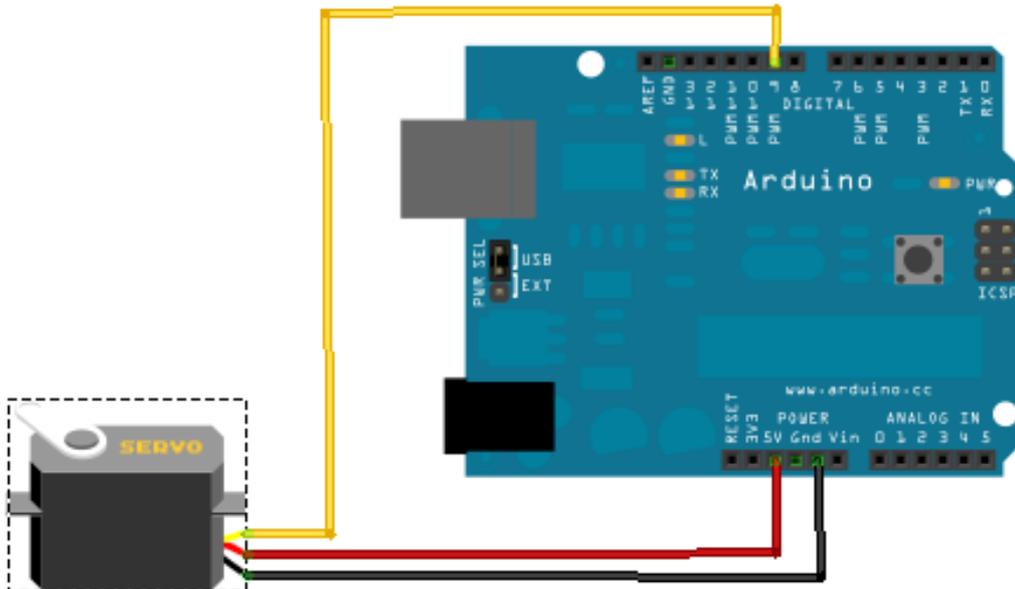


Figura A.7 – Servo motor conectado ao Arduino.
Fonte: Extraído de Arduino Products (2016).

Essa estrutura de funcionamento do Arduino, através de um software *open-source*, possibilita que os estudantes, através das suas investigações, criem um plano de ação, aplique-o na resolução do problema, faça a análise do resultado obtido e verifique se a situação problema inicial foi resolvida. Caso não a tenha resolvido, o estudante pode retornar para qualquer etapa da atividade e trilharem um novo caminho a partir do ponto problemático, o que possibilita a criação do seu próprio aprendizado, não estando atrelado a uma só opção para solução do problema.

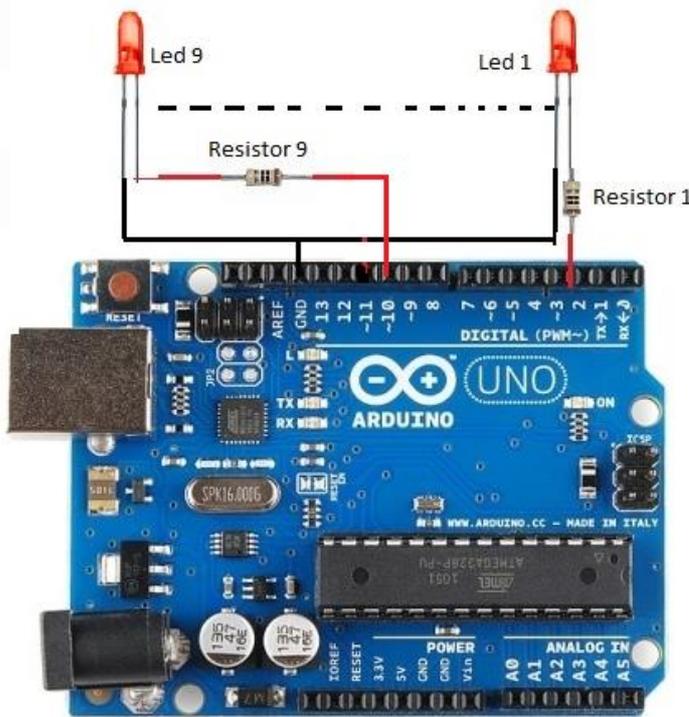
A próxima seção apresenta as atividades com Arduino utilizadas neste roteiro de ensino.

A.2. APLICAÇÃO DAS ATIVIDADES

A.2.1. Conceitos iniciais de matrizes utilizando um painel de led's

Nesta primeira atividade, o estudante será apresentado ao Arduino e terá que pesquisar sua funcionalidade e como utilizá-lo como uma matriz de *led's* (Figuras A.8 e A.9), possibilitando um processo investigativo dos conceitos teóricos básicos de matrizes, tais como linha, coluna, diagonal principal, diagonal secundária e seus elementos (estes conceitos foram os escolhidos por serem visualmente possíveis em uma matriz de *led's*), além dos tipos de matrizes.

O professor deve montar os kits e entrega-los montados para que os estudantes realizem as experimentações.



O pino GND (terra) é comum para todos os led's. Ligar um resistor de 680 ohms em cada pino (2 ao 10) do Arduino em série com um led. Depois, dispor estes led's no formato de uma matriz 3x3, lembrando que o led 1 que esta ligado no pino 2 é o a11, o led2 ligado no pino 3 é o a12 e assim sucessivamente até formar a matriz.

Figura A.8 – Esquema para ligar os led's da matriz 3x3 com Arduino.

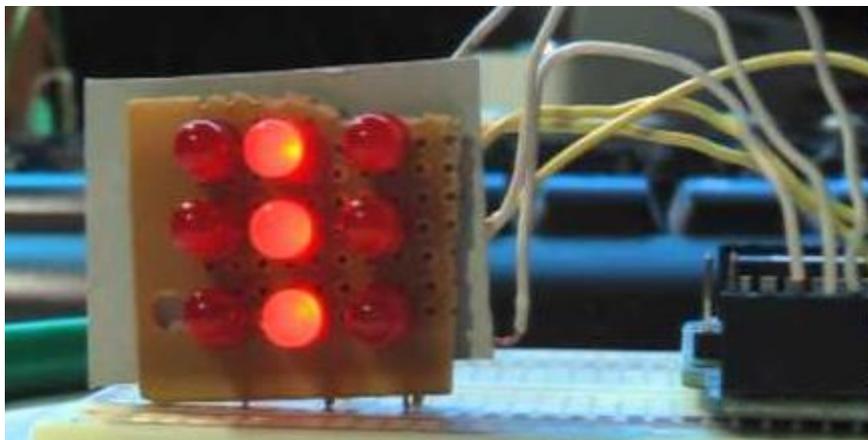


Figura A.9 – Matriz de led's 3x3.
Fonte: Extraído de Arduino 3x3 LED Matrix display (2016).

Nesta atividade não é interessante que o professor monte o experimento juntamente com os estudantes uma vez que a montagem envolve o conhecimento de posição de elementos em uma matriz. Entretanto, após o experimento, o professor pode solicitar que os estudantes explorem o sistema e montem diferentes formatos de matrizes, introduzindo assim, os conceitos de tipos de matrizes. Além disso, caso o professor ache interessante introduzir os conceitos já na montagem do experimento, este propicia o entendimento de posição de elementos de uma matriz.

A seguir, apresentamos o código fonte que deve ser digitado no *software* do Arduino e feito o carregamento. Após esta etapa, o Arduino está pronto para controlar o painel de *led's* e assim, o professor poderá entregar o kit para os estudantes efetuarem as experimentações. Esta atividade fica ainda mais interessante quando os estudantes realizam em dupla, proporcionando assim o debate e outras experimentações entre os estudantes.

Código-Fonte Matriz de Led Arduino 3X3

```
int ledPin2 = 2;
int ledPin3 = 3;
int ledPin4 = 4;
int ledPin5 = 5;
int ledPin6 = 6;
int ledPin7 = 7;
int ledPin8 = 8;
int ledPin9 = 9;
int ledPin10 = 10;
String a11 = "a11";
String a12 = "a12";
String a13 = "a13";
String a21 = "a21";
String a22 = "a22";
String a23 = "a23";
String a31 = "a31";
String a32 = "a32";
String a33 = "a33";
String dip = "dip";
String dis = "dis";
String rst = "rst";
String str;
char c;
char matriz[20];
int x=0;
void setup()
{
  Serial.begin(9600);
  pinMode(ledPin2, OUTPUT);
  pinMode(ledPin3, OUTPUT);
  pinMode(ledPin4, OUTPUT);
  pinMode(ledPin5, OUTPUT);
  pinMode(ledPin6, OUTPUT);
  pinMode(ledPin7, OUTPUT);
  pinMode(ledPin8, OUTPUT);
  pinMode(ledPin9, OUTPUT);
  pinMode(ledPin10, OUTPUT);
}
void loop()
{
  if(Serial.available())
```

```
{
do{
c=Serial.read();
matriz[x]=c;
Serial.print(matriz[x],DEC);
x++;
delay(1); //Delay para o Arduino não perder o dado da Serial
}while(c!='\n');
matriz[x-1]='\0';
Serial.print(matriz);
str=matriz;
if (str==a11)
{
digitalWrite(ledPin2, HIGH);
}
else
if(str==a12)
{
digitalWrite(ledPin3, HIGH);
}
else
if(str==a13)
{
digitalWrite(ledPin4, HIGH);
}
else
if(str==a21)
{
digitalWrite(ledPin5, HIGH);
}
else
if(str==a22)
{
digitalWrite(ledPin6, HIGH);
}
else
if(str==a23)
{
digitalWrite(ledPin7, HIGH);
}
else
if(str==a31)
{
digitalWrite(ledPin8, HIGH);
}
else
if(str==a32)
{
digitalWrite(ledPin9, HIGH);
}
else
```

```

    if(str==a33)
    {
        digitalWrite(ledPin10, HIGH);
    }
    else
        if(str==dip)
        {
            digitalWrite(ledPin2, HIGH);
            digitalWrite(ledPin6, HIGH);
            digitalWrite(ledPin10, HIGH);
        }
    else
        if(str==dis)
        {
            digitalWrite(ledPin4, HIGH);
            digitalWrite(ledPin6, HIGH);
            digitalWrite(ledPin8, HIGH);
        }
    else
        if(str==rst)
        {
            digitalWrite(ledPin2, LOW);
            digitalWrite(ledPin3, LOW);
            digitalWrite(ledPin4, LOW);
            digitalWrite(ledPin5, LOW);
            digitalWrite(ledPin6, LOW);
            digitalWrite(ledPin7, LOW);
            digitalWrite(ledPin8, LOW);
            digitalWrite(ledPin9, LOW);
            digitalWrite(ledPin10, LOW);
        }
    delay(100);
    x=0;
}
}

```

Após digitar estas linhas de comando no software do Arduino e carregar na placa, o Arduino já está pronto para controlar o painel de *led's* e para isso, basta clicar no ícone monitor serial localizado no canto superior direito da tela.

Sempre que for necessária a inserção de dados via teclado, o serial monitor do Arduino deve ser iniciado.

O professor deve indicar alguns comandos iniciais para que os estudantes digitem no computador e observem os resultados no painel de *led's*, e assim, iniciar o processo experimental e investigativo sobre matrizes.

O professor deve orientar o estudante a efetuar os seguintes comandos e anotar o observado no painel de *led's*:

- 1) Digitar no teclado do computador “a11”.
- 2) Digitar no teclado do computador “a12” e assim sucessivamente até posição “a33”.
- 3) Digitar “rst” para apagar todos os *led*’s.
- 4) Digitar “dip” para descobrir o que é a diagonal principal dentro desta matriz (a11, a22, a33).
- 5) Digitar “rst” para apagar todos os *led*’s.
- 6) Digitar “dis” para descobrir o que é a diagonal secundária (a13, a22, a31).

A partir deste ponto, o professor pode propor desafios utilizando o painel, tais como jogo da velha e batalha naval.

O professor, nesta primeira etapa, foi um mediador, e agora deve estimular os estudantes e ajudá-los a formalizar estes conhecimentos:

- O que é uma matriz?
- Quais os tipos de matrizes?
- O que são linhas, colunas, diagonal principal e diagonal secundária?
- Onde visualizamos em nosso cotidiano painéis de *led*’s?
- Onde vemos aplicações de matrizes em nosso cotidiano?

Ao final desta atividade aberta envolvendo a matriz de *led*’s 3x3, o estudante desenvolve a conceituação inicial de linha, coluna, diagonal principal, diagonal secundária e a localização dos elementos dentro de uma matriz.

A.2.2. Conceitos de operações com Matrizes utilizando a movimentação de um braço mecânico

A.2.2.1. Adição de matrizes

Nesta segunda etapa, o professor deve ensinar os conceitos de adição de matrizes através do movimento de um braço robótico (Fig. A.9).

Para implementar a utilização do braço robótico, a inclusão de um esquadro em cada uma de suas articulações seria uma ideia viável e complementar a visualização que já é feita através do prompt de comando do próprio software do Arduino.

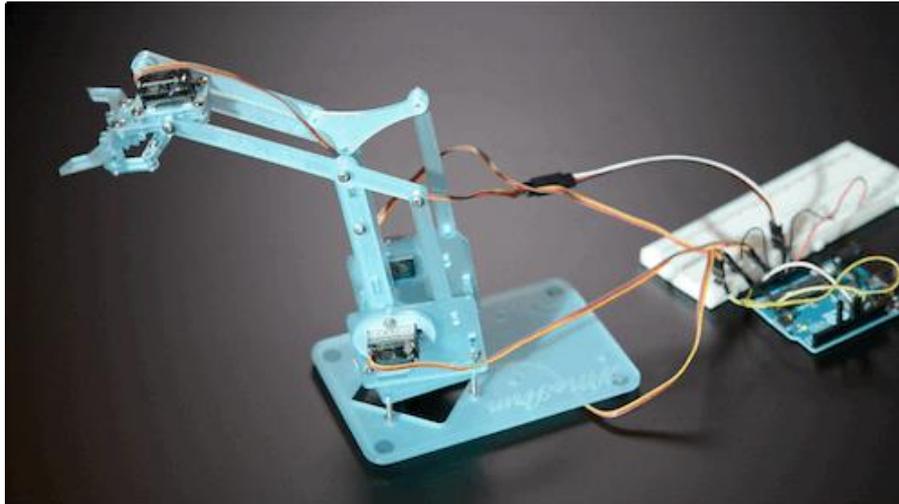


Figura A.9 – Braço Robótico Arduino.
Fonte: Extraído de Vitrônica (2016).

Da mesma forma que no exercício anterior, os estudantes recebem o dispositivo já programado e cabe aos estudantes conectá-lo ao computador através da porta USB e começar a fase de exploração do dispositivo.

A matriz configurada no Arduino para a movimentação do braço robótico é a seguinte:

$$A = \begin{pmatrix} \text{Base do braço} & \text{Altura do braço} \\ \text{Inclinação do braço} & \text{Acionamento da garra} \end{pmatrix},$$

ou seja:

A11 – Base do braço;

A12 – Altura do braço;

A21 – Inclinação do braço;

A22 – Acionamento da garra .

Através do teclado do computador, os estudantes devem inserir a primeira matriz, $A = \begin{pmatrix} 45 & 5 \\ 60 & 0 \end{pmatrix}$, no Arduino. Esta matriz executa um movimento no braço robótico e o movimenta para a posição:

A11 = A base gira no sentido horário e para na posição 45°;

A12 = O braço sobe e para na posição de 5°;

A21 = O braço inclina 60° para frente;

A22 = A garra se abre até a posição 0°.

E esta matriz é armazenada na memória do dispositivo.

A Figura A.10 apresenta um exemplo de movimentação do braço robótico.

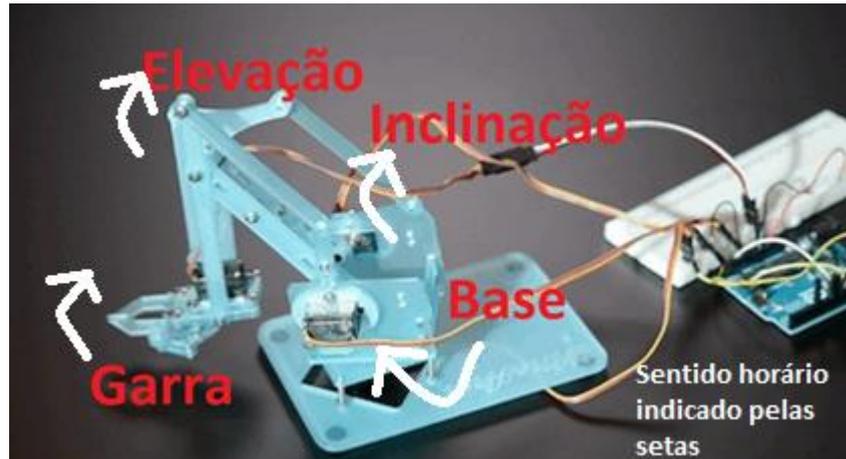


Figura A.10 – Braço Robótico Arduino: um exemplo de movimentação.
Fonte: Extraído de Vitrônica (2016).

Ao inserir a segunda matriz, $B = \begin{pmatrix} 5 & 100 \\ 30 & 10 \end{pmatrix}$, no Arduino, o braço robótico prossegue o movimento de onde estava, pois, a segunda matriz B foi adicionada a primeira matriz A, resultando em uma nova matriz $C = \begin{pmatrix} 50 & 105 \\ 90 & 100 \end{pmatrix}$ e o braço robótico realiza a seguinte movimentação:

A11 = A base gira mais 5° no sentido horário, chegando na posição 50°

A12 = O braço sobe mais 100° e chega na altura de 105°

A21 = O braço inclina mais 30° para frente e chega na posição 90°

A22 = A garra se abre mais 100° e chega na posição 100°

Cabe aos estudantes descobrirem qual operação de matrizes que movimentou o braço e como se deu esse movimento.

Após término da experimentação de adição de matrizes, o professor pode induzir os estudantes a conceituarem a ideia da multiplicação de uma matriz por um escalar, através da adição sucessiva de matrizes idênticas, e assim, quando chegar nesta etapa da atividade, este conceito já estará formalizado.

Para realizarmos a operação de subtração de matrizes, o professor pode utilizar-se da adição de matrizes e iniciar os seguintes procedimentos:

Os estudantes, através do teclado do computador, devem inserir a primeira matriz, $A = \begin{pmatrix} 10 & 5 \\ 12 & 20 \end{pmatrix}$, no Arduino via teclado do computador que movimenta o braço robótico para posição desejada como na adição de matrizes:

A11 = A base gira 10° no sentido horário;

A12 = O braço sobe uma altura de 5°;

A21 = O braço inclina 12° para frente;

A22 = A garra se movimenta para a posição 20°.

Sendo a matriz A armazenada na memória do dispositivo.

Ao inserir a segunda matriz,

$$B = \begin{pmatrix} -8 & -4 \\ -10 & -10 \end{pmatrix}$$

no Arduino, o braço robótico prossegue o movimento de onde estava, pois, a segunda matriz B foi adicionada a primeira A, resultando em uma nova matriz $C = \begin{pmatrix} 2 & 1 \\ 2 & 10 \end{pmatrix}$, pois o braço robótico efetuou o seguinte movimento:

A11 = A base gira 8° no sentido anti-horário e chega na posição 2°

A12 = O braço desce 4° e chega na altura de 1°

A21 = O braço recua 10° e chega na posição 2°

A22 = A garra se fecha 10° e chega na posição 10°

Essa operação pode ser utilizada para experimentar a operação de subtração de matriz, e os estudantes poderão concluir que a operação realizada é esta, cabendo ao professor/pesquisador apenas mediar esta ideia, demonstrando e confirmando que::

$$A + B = \begin{pmatrix} 10 & 5 \\ 12 & 20 \end{pmatrix} + \begin{pmatrix} -8 & -4 \\ -10 & -10 \end{pmatrix} = \begin{pmatrix} 10 & 5 \\ 12 & 20 \end{pmatrix} - \begin{pmatrix} 8 & 4 \\ 10 & 10 \end{pmatrix} = \begin{pmatrix} 2 & 1 \\ 2 & 10 \end{pmatrix}.$$

E que o mesmo ocorreria se fosse digitada matrizes onde algum dos elementos tem sinal negativo:

$$A + B = \begin{pmatrix} -10 & 5 \\ 12 & -10 \end{pmatrix} + \begin{pmatrix} -8 & -4 \\ 10 & 10 \end{pmatrix} = \begin{pmatrix} -18 & 1 \\ 22 & 0 \end{pmatrix}.$$

Caso o professor queira trabalhar a subtração de matrizes isoladamente da adição de matrizes, elaboramos também um código fonte para que o mesmo possa trabalhar as duas operações isoladamente.

Durante todas as etapas do processo, os estudantes são orientados pelo seu professor/pesquisador a fazer anotações sobre suas dúvidas, hipóteses, experimentações e descobertas em seu diário de bordo.

Com as observações, anotações, investigações e experimentações sobre o que ocasionou a movimentação do braço robótico e através de seus conhecimentos prévios de operações com números reais, os estudantes são capazes de construir o conhecimento de adição e subtração de matrizes;

A avaliação de todo o processo é feita continuamente, através da observação do professor/pesquisador sobre o andamento das experimentações e investigações

dos estudantes, como também com as atividades e exercícios aplicados após a experimentação.

A matriz configurada inicialmente no Arduino serve de base para o seu funcionamento em todas as atividades aqui propostas.

Nesta etapa, recomendamos que o professor monte todo o experimento antes de entregar aos estudantes. No entanto, recomendamos também que após a atividade, o professor estimule os estudantes a investigarem e elaborarem a movimentação do braço mecânico com outras operações de matrizes (por exemplo, com o cálculo da matriz inversa).

A seguir, apresentamos o código fonte que deve ser digitado no *software* do Arduino e feito o carregamento para adição de matrizes. Após esta etapa, o Arduino está pronto para controlar o braço mecânico e assim, o professor poderá entregar o kit para os estudantes efetuarem as experimentações. Esta atividade fica ainda mais interessante quando os estudantes realizam em duplas ou trios, proporcionando assim o debate e outras experimentações entre os estudantes.

Código-Fonte de Adição de Matrizes

```
#include <Servo.h> //BIBLIOTECA DE FUNÇÕES COM MOTORES SERVOS
// DECLARAÇÃO DOS MOTORES USADOS NO PROJETO...
Servo servoa11;
Servo servoa12;
Servo servoa21;
Servo servoa22;
float provisorio = 0;
float tecla = 0;
float provisorio2 = 0;
float tecla2 = 0;
float provisorio3 = 0;
float tecla3 = 0;
float provisorio4 = 0;
float tecla4 = 0;
int posicao11 = 0;
int posicao12 = 0;
int posicao21 = 0;
int posicao22 = 0;
int opcao;
void direita(float valor) //FUNÇÃO RESPONSÁVEL POR GIRAR A BASE PARA A DIREITA...
{
  servoa11.write(valor);
  delay(1300); //GIRA POR ATÉ ATINGIR UM ÂNGULO DE APROX.90 GRAUS E PARA...
```

```

//parado();
}
void setup() //FUNÇÃO ONDE SÃO ESPECIFICADAS AS CONFIGURAÇÕES DO PROJETO...
{
//ATRIBUIÇÃO DE PINOS PARA OS MOTORES
servoa11.attach(10);
servoa12.attach(11);
servoa21.attach(12);
servoa22.attach(13);
Serial.begin(9600);
}
void loop()
{
Serial.println("Digite a opcao no menu");
Serial.println("1 - posição a11 base");
Serial.println("2 - posição a12 altura");
Serial.println("3 - posição a21 inclinação");
Serial.println("4 - posição a22 garra");
while (Serial.available() == 0) { }
opcao = Serial.parseInt();
switch (opcao) {
case 1:{
Serial.println("digite o valor da a11 (em graus)");
while (Serial.available() == 0) { }
delay (10000);
tecla = Serial.parseFloat();
float resultante = tecla + provisorio;
if (resultante > 180) {
resultante = 180;
}
if (resultante < 0) {
resultante = 0;
}
Serial.print("valor da a11: ");
Serial.println(resultante);
servoa11.write(resultante);
provisorio = resultante;
break;}
case 2:{
Serial.println("digite o valor do a12 (em graus)");
while (Serial.available() == 0) { }
delay (10000);
tecla2 = Serial.parseFloat();
float resultante2 = tecla2 + provisorio2;
if (resultante2 > 180) {
resultante2 = 180;
}
if (resultante2 < 0) {
resultante2 = 0;
}
}
}
}

```

```

Serial.print("valor do a12: ");
Serial.println(resultante2);
servoa12.write(resultante2);
provisorio2 = resultante2;
break;}
case 3:{
Serial.println("digite o valor de a21 (em graus)");
while (Serial.available() == 0) { }
delay (10000);
tecla3 = Serial.parseFloat();
float resultante3 = tecla3 + provisorio3;
if (resultante3 > 180) {
    resultante3 = 180;
}
if (resultante3 < 0) {
    resultante3 = 0;
}
Serial.print("valor da a21: ");
Serial.println(resultante3);
servoa21.write(resultante3);
provisorio3 = resultante3;
break;}
case 4:{
Serial.println("digite o valor de a22 (em graus)");
while (Serial.available() == 0) { }
delay (10000);
tecla4 = Serial.parseFloat();
float resultante4 = tecla4 + provisorio4;
if (resultante4 > 180) {
    resultante4 = 180;
}
if (resultante4 < 0) {
    resultante4 = 0;
}
Serial.print("valor de a22? ");
Serial.println(resultante4);
servoa22.write(resultante4);
provisorio4 = resultante4;
break;}
}
}

```

A.2.2.2. Subtração de matrizes

Na atividade de adição de matrizes, já aplicou-se o conceito de subtração de matrizes, mas caso o professor queira trabalha-lo separadamente, elaboramos uma atividade de subtração de matrizes.

Nesta atividade, o professor deve seguir os mesmos passos descritos na atividade anterior, desde a programação do Arduino até o início da aula.

A matriz inicial é $A = \begin{pmatrix} 90 & 70 \\ 30 & 50 \end{pmatrix}$, sendo que a matriz digitada posteriormente pelos estudantes será subtraída desta. Caso o professor queira ter outra matriz inicial, basta modificar as linhas de comando *float* b11, b12, b21, b22 e inserir os valores que achar conveniente.

Código-Fonte Subtração de Matrizes

```
#include <Servo.h> //BIBLIOTECA DE FUNÇÕES COM MOTORES SERVOS
// DECLARAÇÃO DOS MOTORES USADOS NO PROJETO...
Servo servoa11;
Servo servoa12;
Servo servoa21;
Servo servoa22;
float b11 = 90;
float tecla = 0;
float b12 =70;
float tecla2 = 0;
float b21 = 30;
float tecla3 = 0;
float b22 = 50;
float tecla4 = 0;
int posicao11 = 0;
int posicao12 = 0;
int posicao21 = 0;
int posicao22 = 0;
int opcao;
void direita(float valor) //FUNÇÃO RESPONSÁVEL POR GIRAR A BASE PARA A DIREITA...
{
  servoa11.write(valor);
  delay(1300); //GIRA POR ATÉ ATINGIR UM ÂNGULO DE APROX.90 GRAUS E PARA...
  //parado();
}
void setup() //FUNÇÃO ONDE SÃO ESPECIFICADAS AS CONFIGURAÇÕES DO PROJETO...
{
  //ATRIBUIÇÃO DE PINOS PARA OS MOTORES
  servoa11.attach(10);
  servoa12.attach(11);
  servoa21.attach(12);
  servoa22.attach(13);
  Serial.begin(9600);
}
void loop()
```

```

{
Serial.println("Digite a opcao no menu");
Serial.println("1 - posição a11 base");
Serial.println("2 - posição a12 altura");
Serial.println("3 - posição a21 inclinação");
Serial.println("4 - posição a22 garra");
while (Serial.available() == 0) { }
opcao = Serial.parseInt();
switch (opcao) {
case 1:{
Serial.println("digite o valor da a11 (em graus)");
while (Serial.available() == 0) { }
delay (10000);
tecla = Serial.parseFloat();
float resultante = b11 - tecla;
if (resultante > 180) {
resultante = 180;
}
if (resultante < 0) {
resultante = 0;
}
Serial.print("valor da a11: ");
Serial.println(resultante);
servoa11.write(resultante);
b11 = resultante;
break;}
case 2:{
Serial.println("digite o valor do a12 (em graus)");
while (Serial.available() == 0) { }
delay (10000);
tecla2 = Serial.parseFloat();
float resultante2 = b12 - tecla2;
if (resultante2 > 180) {
resultante2 = 180;
}
if (resultante2 < 0) {
resultante2 = 0;
}
Serial.print("valor do a12: ");
Serial.println(resultante2);
servoa12.write(resultante2);
b12 = resultante2;
break;}
case 3:{
Serial.println("digite o valor de a21 (em graus)");
while (Serial.available() == 0) { }
delay (10000);
tecla3 = Serial.parseFloat();
float resultante3 = b21 - tecla3;
if (resultante3 > 180) {
resultante3 = 180;
}
}
}

```

```

}
if (resultante3 < 0) {
  resultante3 = 0;
}
Serial.print("valor da a21: ");
Serial.println(resultante3);
servoa21.write(resultante3);
b21 = resultante3;
break;}
case 4:{
  Serial.println("digite o valor de a22 (em graus)");
  while (Serial.available() == 0) { }
  delay (10000);
  tecla4 = Serial.parseFloat();
  float resultante4 = b22 - tecla4;
  if (resultante4 > 180) {
    resultante4 = 180;
  }
  if (resultante4 < 0) {
    resultante4 = 0;
  }
  Serial.print("valor de a22? ");
  Serial.println(resultante4);
  servoa22.write(resultante4);
  b22 = resultante4;
  break;}
}
}

```

No caso da subtração de matrizes, o estudante já teve a iniciação feita através da adição de matrizes e/ou deve ser instruído a seguir os seguintes passos:

1) Através do teclado do computador o estudante deve inserir a primeira matriz: $B = \begin{pmatrix} 10 & 5 \\ 12 & 20 \end{pmatrix}$,

2) Como já havia uma matriz inicial programada no Arduino, ele efetua a seguinte operação:

$\begin{pmatrix} 90 & 70 \\ 30 & 50 \end{pmatrix} - \begin{pmatrix} 10 & 5 \\ 12 & 20 \end{pmatrix} = \begin{pmatrix} 80 & 65 \\ 18 & 30 \end{pmatrix}$ e o braço robótico se movimenta para a seguinte posição:

A11 = A base gira 10° no sentido anti-horário e chega na posição 80°

A12 = O braço desce 5° e chega na altura de 65°

A21 = O braço recua 12° para frente e chega na posição 18°

A22 = A garra se fecha 20° e chega na posição 30°

Relembrando que toda a movimentação do braço é feita com base no modelo de matriz descrita no início da atividade de adição de matrizes.

O estudante deve concluir, através de discussões e experimentações, que o braço robótico prosseguiu o movimento de onde estava, resultando em um novo posicionamento do braço, uma vez que a matriz B foi subtraída da matriz A, previamente inserida no Arduino, resultando na matriz $C = \begin{pmatrix} 80 & 65 \\ 18 & 30 \end{pmatrix}$.

Durante todas as etapas do processo, os estudantes são orientados pelo seu professor/pesquisador a fazer anotações sobre suas dúvidas, hipóteses, experimentações e descobertas em seu diário de bordo.

Após esta experimentação inicial, o professor pode promover uma atividade lúdica, como por exemplo, pedir para que os estudantes coloquem objetos na mesa e que faça o braço robótico chegar até onde está o objeto e o agarre utilizando subtração de matrizes.

A.2.2.2. Multiplicação de escalar por matriz

Nesta segunda etapa, o professor deve ensinar o conceito de multiplicação de um escalar por uma matriz. Para esta atividade, continuamos com o experimento de movimentação de um braço robótico (Fig. A.9). Abaixo, apresentamos o código-fonte para inserção no Arduino que multiplica uma matriz por um escalar.

Código-Fonte Multiplicação de uma Matriz por um Escalar

```
#include <Servo.h> //BIBLIOTECA DE FUNÇÕES COM MOTORES SERVOS
// DECLARAÇÃO DOS MOTORES USADOS NO PROJETO...
Servo servoa11;
Servo servoa12;
Servo servoa21;
Servo servoa22;
escalar = 0;
float provisorio = 180;
float tecla = 0;
float provisorio2 =180;
float tecla2 = 0;
float provisorio3 = 180;
float tecla3 = 0;
float provisorio4 = 180;
float tecla4 = 0;
```

```

int posicao11 = 0;
int posicao12 = 0;
int posicao21 = 0;
int posicao22 = 0;
int opcao;
void direita(float valor) //FUNÇÃO RESPONSÁVEL POR GIRAR A BASE PARA A
DIREITA...
{
  servoa11.write(valor);
  delay(1300); //GIRA POR ATÉ ATINGIR UM ÂNGULO DE APROX.90 GRAUS E
PARA...
  //parado();
}
void setup() //FUNÇÃO ONDE SÃO ESPECIFICADAS AS CONFIGURAÇÕES DO
PROJETO...
{
  //ATRIBUIÇÃO DE PINOS PARA OS MOTORES
  servoa11.attach(10);
  servoa12.attach(11);
  servoa21.attach(12);
  servoa22.attach(13);
  Serial.begin(9600);
}
void loop()
{
  Serial.println("Digite a opcao no menu");
  Serial.println("1 - posição a11 base");
  Serial.println("2 - posição a12 altura");
  Serial.println("3 - posição a21 inclinação");
  Serial.println("4 - posição a22 garra");
  Serial.println("5 - escalar");
  while (Serial.available() == 0) { }
  opcao = Serial.parseInt();
  switch (opcao) {
    case 5:{
      Serial.println("digite o valor do escalar");
      while (Serial.available() == 0) { }
      delay (10000);
      escalar = Serial.parseFloat();
      Serial.print("valor do escalar: ");
      Serial.println(escalar);
      break;}

    case 1:{
      Serial.println("digite o valor da a11 (em graus)");
      while (Serial.available() == 0) { }
      delay (10000);
      tecla = Serial.parseFloat();
      float resultante = escalar * tecla;
      if (resultante > 180) {
        resultante = 180;
      }
    }
  }
}

```

```
}
if (resultante < 0) {
    resultante = 0;
}
Serial.print("valor da a11: ");
Serial.println(resultante);
servoa11.write(resultante);
provisorio = resultante;
break;}
case 2:{
    Serial.println("digite o valor do a12 (em graus)");
    while (Serial.available() == 0) { }
    delay (10000);
    tecla2 = Serial.parseFloat();
    float resultante2 = escalar * tecla2;
    if (resultante2 > 180) {
        resultante2 = 180;
    }
    if (resultante2 < 0) {
        resultante2 = 0;
    }
    Serial.print("valor do a12: ");
    Serial.println(resultante2);
    servoa12.write(resultante2);
    provisorio2 = resultante2;
    break;}
case 3:{
    Serial.println("digite o valor de a21 (em graus)");
    while (Serial.available() == 0) { }
    delay (10000);
    tecla3 = Serial.parseFloat();
    float resultante3 = escalar * tecla3;
    if (resultante3 > 180) {
        resultante3 = 180;
    }
    if (resultante3 < 0) {
        resultante3 = 0;
    }
    Serial.print("valor da a21: ");
    Serial.println(resultante3);
    servoa21.write(resultante3);
    provisorio3 = resultante3;
    break;}
case 4:{
    Serial.println("digite o valor de a22 (em graus)");
    while (Serial.available() == 0) { }
    delay (10000);
    tecla4 = Serial.parseFloat();
    float resultante4 = escalar * tecla4;
    if (resultante4 > 180) {
        resultante4 = 180;
```

```

}
if (resultante4 < 0) {
    resultante4 = 0;
}
Serial.print("valor de a22? ");
Serial.println(resultante4);
servoa22.write(resultante4);
provisorio4 = resultante4;
break;}
}
}

```

Devemos lembrar que o professor já iniciou este conceito com os estudantes ao estimular a soma da mesma matriz algumas vezes, na atividade aberta sobre adição de matrizes. Neste momento, o professor pode questionar os estudantes sobre as investigações realizadas com o experimento de adição de matrizes e quais as conclusões em relação ao resultado dessa soma.

A partir do retorno dos estudantes sobre as hipóteses das investigações realizadas, o professor pode indicar os seguintes comandos:

- 1) Através do teclado do computador o estudante deve inserir o valor do escalar: $x = 4$, e anotar o ocorrido no monitor e na movimentação do braço robótico.
- 2) A seguir, o estudante deve inserir a matriz através do teclado: $A = \begin{pmatrix} 8 & 21 \\ 13 & 10 \end{pmatrix}$.
- 3) Após a inserção do escalar e da matriz A, no monitor aparecerá o resultado, como também a movimentação do braço robótico.
- 4) O estudante já realizou a experimentação de adição de matrizes e verificou como ocorre a multiplicação de uma matriz por um escalar e fica fácil de conceituar esta operação e chegar ao resultado $x.A = \begin{pmatrix} 32 & 84 \\ 52 & 40 \end{pmatrix}$, movimentando o braço para a posição:
 - A11 = A base gira 32° no sentido horário, chegando na posição 32°
 - A12 = O braço sobe 84° e chega na altura de 84°
 - A21 = O braço avança 52° e chega na posição 52°
 - A22 = A garra se abre 40° e chega na posição 40°

Após esta experimentação inicial, o professor pode promover uma atividade lúdica, como por exemplo, pedir para que os estudantes coloquem objetos na mesa e que faça o braço robótico chegar até onde está o objeto e o agarre utilizando multiplicação por escalar.

A.2.2.3 Multiplicação de Matrizes

Nesta segunda etapa, o professor deve ensinar o conceito de multiplicação de matrizes. Para esta atividade, continuamos com o experimento de movimentação de um braço robótico (Fig. A.9). Abaixo, apresentamos o código-fonte para inserção no Arduino que efetua a multiplicação de matrizes.

Código-Fonte de Multiplicação de Matrizes

```
#include <Servo.h> //BIBLIOTECA DE FUNÇÕES COM MOTORES SERVOS
// DECLARAÇÃO DOS MOTORES USADOS NO PROJETO...
Servo servoa11;
Servo servoa12;
Servo servoa21;
Servo servoa22;
float provisorio = 0;
float tecla = 0;
float provisorio2 = 0;
float tecla2 = 0;
float provisorio3 = 0;
float tecla3 = 0;
float provisorio4 = 0;
float tecla4 = 0;
float provisorio5 = 0;
float tecla5 = 0;
float provisorio6 = 0;
float tecla6 = 0;
float provisorio7 = 0;
float tecla7 = 0;
float provisorio8 = 0;
float tecla8 = 0;
int posicao11 = 0;
int posicao12 = 0;
int posicao21 = 0;
int posicao22 = 0;
int posicaob11 = 0;
int posicaob12 = 0;
int posicaob21 = 0;
int posicaob22 = 0;
int opcao;
void direita(float valor) //FUNÇÃO RESPONSÁVEL POR GIRAR A BASE PARA A DIREITA...
{
  servoa11.write(valor);
  delay(1300); //GIRA POR ATÉ ATINGIR UM ÂNGULO DE APROX.90 GRAUS E PARA...
  //parado();
}
```

```

void setup() //FUNÇÃO ONDE SÃO ESPECIFICADAS AS CONFIGURAÇÕES DO PROJETO...
{
  //ATRIBUIÇÃO DE PINOS PARA OS MOTORES
  servoa11.attach(10);
  servoa12.attach(11);
  servoa21.attach(12);
  servoa22.attach(13);
  Serial.begin(9600);
}
void loop()
{
  Serial.println("Digite a opção no menu");
  Serial.println("1 - posição a11 base");
  Serial.println("2 - posição a12 altura");
  Serial.println("3 - posição a21 inclinação");
  Serial.println("4 - posição a22 garra");
  Serial.println("5 - posição b11 base");
  Serial.println("6 - posição b12 altura");
  Serial.println("7 - posição b21 inclinação");
  Serial.println("8 - posição b22 garra");
  while (Serial.available() == 0) { }
  opção = Serial.parseInt();
  switch (opção) {
    case 1:{
      Serial.println("digite o valor da a11 (em graus)");
      while (Serial.available() == 0) { }
      delay (10000);
      tecla = Serial.parseFloat();
      Serial.print("valor da a11: ");
      Serial.println(tecla);
      break;}
    case 2:{
      Serial.println("digite o valor do a12 (em graus)");
      while (Serial.available() == 0) { }
      delay (10000);
      tecla2 = Serial.parseFloat();
      Serial.print("valor do a12: ");
      Serial.println(tecla2);
      break;}
    case 3:{
      Serial.println("digite o valor de a21 (em graus)");
      while (Serial.available() == 0) { }
      delay (10000);
      tecla3 = Serial.parseFloat();
      Serial.print("valor da a21: ");
      Serial.println(tecla3);
      break;}
    case 4:{
      Serial.println("digite o valor de a22 (em graus)");
      while (Serial.available() == 0) { }

```

```

    delay (10000);
    tecla4 = Serial.parseFloat();
    Serial.print("valor de a22? ");
    Serial.println(tecla4);
    break;}
    case 5:{
Serial.println("digite o valor da b11 (em graus)");
    while (Serial.available() == 0) { }
    delay (10000);
    tecla5 = Serial.parseFloat();
    Serial.print("valor da b11: ");
    Serial.println(tecla5);
    break;}
    case 6:{
    Serial.println("digite o valor do b12 (em graus)");
    while (Serial.available() == 0) { }
    delay (10000);
    tecla6 = Serial.parseFloat();
    Serial.print("valor do b12: ");
    Serial.println(tecla6);
    break;}
    case 7:{
    Serial.println("digite o valor de b21 (em graus)");
    while (Serial.available() == 0) { }
    delay (10000);
    tecla7 = Serial.parseFloat();
    Serial.print("valor da b21: ");
    Serial.println(tecla7);
    break;}
    case 8:{
    Serial.println("digite o valor de b22 (em graus)");
    while (Serial.available() == 0) { }
    delay (10000);
    tecla8 = Serial.parseFloat();
    Serial.print("valor de b22? ");
    Serial.println(tecla8);
    break;}
}
float resultante1 = (tecla1*tecla5) + (tecla2*tecla7);
float resultante2 = (tecla1*tecla6) + (tecla2*tecla8);
float resultante3 = (tecla3*tecla5) + (tecla3*tecla6);
float resultante4 = (tecla1*tecla5) + (tecla4*tecla8);
    Serial.print("valor de c11");
    Serial.println(resultante1);
    Servoa11.write(resultante1);
    Serial.print("valor de c12 ");
    Serial.println(resultante2);
    Servoa12.write(resultante2);
    Serial.print("valor de c21 ");
    Serial.println(resultante3);
    servoa22.write(resultante3);

```

```

Serial.print("valor de c22");
Serial.println(resultante4);
servoa22.write(resultante4);
}

```

Nesta etapa, o professor pode solicitar aos estudantes que façam reflexões de como ocorreria à multiplicação de matrizes, elaborando as hipóteses de uma investigação matemática. A partir daí, o professor pode solicitar aos estudantes que experimentem as hipóteses a partir da movimentação do braço mecânico.

É possível que nesta etapa seja mais difícil que os estudantes deduzam a operação, no entanto, a experimentação é importante para que o estudante vivencie o teste de suas hipóteses.

Neste momento, o professor pode orientar os estudantes a digitar os seguintes comandos e anotar o ocorrido:

1) Através do teclado do computador o estudante deve inserir a primeira matriz: $A = \begin{pmatrix} 3 & 2 \\ 1 & 0 \end{pmatrix}$, e anotar o ocorrido no monitor e na movimentação do braço robótico.

2) A seguir, o estudante deve inserir a segunda matriz através do teclado: $B = \begin{pmatrix} 2 & 1 \\ 4 & 5 \end{pmatrix}$.

3) Após a inserção das duas matrizes, no monitor aparecerá o resultado da operação, a matriz $C = \begin{pmatrix} 14 & 13 \\ 2 & 1 \end{pmatrix}$, como também a movimentação do braço robótico para esta posição:

A11 = A base gira 14° no sentido horário, chegando na posição 14°

A12 = O braço sobe 13° e chega na altura de 13°

A21 = O braço avança 2° para frente e chega na posição 2°

A22 = A garra se abre e chega na posição 1°

Como esta operação não é tão facilmente dedutível quanto às outras, o professor pode realizar duas diferentes intervenções: pedir para que os estudantes tentem deduzir a operação, instigando para que os mesmos trabalhem com operações entre linhas e colunas; ou para que eles pesquisem outras operações possíveis entre duas matrizes e realizem experimentos para verificar e demonstrar a operação.

O professor pode até sugerir que os estudantes partam do mesmo princípio da adição de matrizes, multiplicar elemento a elemento e anotar o que ocorre, e a partir daí, verificar que não é o fato ocorrido e assim, partir para novas hipóteses.

Uma outra sugestão é que os estudantes realizem experimentações utilizando multiplicações do tipo $\begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 2 & 0 \\ 1 & 0 \end{pmatrix}$, depois $\begin{pmatrix} 2 & 1 \\ 1 & 3 \end{pmatrix} \cdot \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 2 \\ 0 & 1 \end{pmatrix}$ e assim sucessivamente até concluir que na multiplicação de matrizes ocorre uma multiplicação de linhas por colunas.

Essas intervenções relacionam-se com a investigação matemática uma vez que tentar deduzir ou pesquisar as operações envolvidas entre os elementos das linhas e colunas das matrizes induz os estudantes a elaborarem hipóteses e prova-las através da experimentação, que vem a ser a prova do ocorrido.

Ao realizar a pesquisa ou deduzir o conceito de multiplicação de matrizes, os estudantes podem realizar novas inserções de matrizes, analisar a movimentação do braço, coletar os resultados, analisar e verificar se o ocorrido na experimentação condiz com os resultados conceituais realizados por eles, assim criando um ciclo de experimentação, investigação matemática e generalização do conceito aprendido.

Ao realizar a pesquisa ou deduzir o conceito de multiplicação de matrizes, os estudantes podem realizar novas inserções de matrizes, analisar a movimentação do braço, coletar os resultados, analisar e verificar se o ocorrido na experimentação condiz com os resultados conceituais realizados por eles, assim criando um ciclo de experimentação, investigação matemática e generalização do conceito aprendido.

A.3. SUGESTÃO DE AVALIAÇÃO

Durante todas as etapas, o professor deve orientar os estudantes a anotar suas dúvidas, descobertas e conclusões em seu caderno.

A avaliação do aprendizado deve ser feita continuamente, durante todo o processo, através das observações e conclusões realizadas pelos estudantes durante as experimentações, como também através de atividades e exercícios aplicados após as experimentações (exercícios de fixação, resolução de situações-problemas, entre outros). Estes últimos, por sua vez, terão como objetivo verificar se os estudantes conseguem transpor os conhecimentos obtidos nas atividades para outros contextos trazidos pelos exercícios, além de possíveis generalizações.

É importante nesta atividade que o professor observe a participação do estudantes nas discussões e nas hipóteses elaboradas durante a investigação. A participação do estudante e a ação do professor como orientador da atividade são quesitos essenciais na aplicação deste roteiro.

A.4. REFERÊNCIAS

ARDUINO GENUINO. Disponível em: <www.arduino.cc> Acesso em: 04 de jan. 2016

ARDUINO 3X3 LED MATRIX DISPLAY. Disponível em: <<http://basbrun.com>> Acesso em: 04 de jan. 2017

BLOG FILIPE FLOP. Disponível em: <http://blog.filipeflop.com> Acesso em: 04 de Janeiro de 2017

SILVA, Fábio Anderson de Assumpção. **Utilizando o Arduino como atividade aberta de investigação matemática para o ensino de conceitos de matrizes**. São Paulo, SP, 2017. Dissertação (Mestrado Profissional em Ensino de Ciências e Matemática), IFSP – Campus São Paulo, São Paulo, 2017..

VINITRÔNICA. Disponível em: www.vinitronica.com.br. Acesso em: 04 de Janeiro de 2017;