



Universidade Federal de Mato Grosso  
Instituto de Ciências Exatas e da Terra  
Departamento de Matemática



---

# Projeto e Construção de um protótipo para plotagem de gráficos 2D utilizando a plataforma Arduino

**Ludimila Gonçalves Pereira**

Mestrado Profissional em Matemática: PROFMAT/SBM

Orientador: **Prof. Dr. Adilson Antônio Berlatto**

Barra do Garças - MT

julho de 2018

# Projeto e construção de um protótipo para plotagem de gráficos 2D utilizando a plataforma Arduino

Este exemplar corresponde à redação final da dissertação, devidamente corrigida e defendida por Ludimila Gonçalves Pereira e aprovada pela comissão julgadora.

Barra do Garças, 25 de julho de 2018.

Prof. Dr. Adilson Antônio Berlatto  
Orientador

## **Banca examinadora:**

Prof. Dr. Adilson Antônio Berlatto  
Prof. Dr. Juan Elmer Villanueva Zevallos  
Prof. Dr. Paulo César Cavalcante de Oliveira

Dissertação apresentada ao curso de Mestrado Profissional em Matemática – PROFMAT, da Universidade Federal de Mato Grosso, como requisito parcial para obtenção do título **de Mestre em Matemática**.

### **Dados Internacionais de Catalogação na Fonte.**

P436p Pereira, Ludimila Gonçalves.  
Projeto e construção de um protótipo para plotagem de gráficos 2D utilizando a plataforma Arduino / Ludimila Gonçalves Pereira. -- 2018  
x, 59 f. : il. color. ; 30 cm.

Orientador: Adilson Antônio Berlatto.  
Dissertação (mestrado profissional) - Universidade Federal de Mato Grosso, Instituto de Ciências Exatas e da Terra, Programa de Pós-Graduação em Matemática, Cuiabá, 2018.  
Inclui bibliografia.

1. Ensino de Matemática. 2. Protótipo. 3. Ensino de funções. I. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

**Permitida a reprodução parcial ou total, desde que citada a fonte.**



MINISTÉRIO DA EDUCAÇÃO  
UNIVERSIDADE FEDERAL DE MATO GROSSO  
PRÓ-REITORIA DE ENSINO DE PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM MATEMÁTICA EM REDE NACIONAL – PROFMAT  
Av. Fernando Corrêa da Costa, 2367 – Boa Esperança – 78.060-900 – Cuiabá/MT  
Fone: (65) 3615-8576 – E-mail: profmat@ufmt.br

## FOLHA DE APROVAÇÃO

Título: "Projeto e construção de um protótipo para plotagem de gráficos 2d utilizando a plataforma arduino"

Autor: **Ludimila Gonçalves Pereira**

defendida e aprovada em 25/07/2018.

Composição da Banca Examinadora:

---

Presidente Banca/Orientador      Doutor      Adilson Antônio Berlatto  
Instituição:      Universidade Federal de Mato Grosso

Examinador Interno      Doutor      Juan Elmer Villanueva Zevallos  
Instituição:      Universidade Federal de Mato Grosso

Examinador Externo      Doutor      Paulo César Cavalcante de Oliveira  
Instituição:      Universidade Regional do Cariri

Cuiabá, 25/07/2018.

# Resumo

Este trabalho trata da combinação multidisciplinar de uma ferramenta tecnológica moderna, acessível e de baixo custo com a matemática ensinada no ensino médio. Ele abordará o projeto e manufatura de um protótipo para plotagem, visualização e análise de funções em 2 dimensões, desenvolvido com a utilização de componentes eletrônicos e a plataforma Arduino. A proposta é capacitar e incentivar os educadores à utilizar tal ferramenta como suporte ao ensino da disciplina, incentivando os alunos às áreas correlatas do protótipo e também colaborando para visualização de conceitos matemáticos abstratos e conseqüentemente contribuindo para o seu aprendizado.

**Palavras chave:** Ensino de matemática, protótipo, ensino de funções.

# Abstract

This paper presents a multidisciplinary combination of a modern, accessible and low cost technological tool with the math taught in high school. He will approach the design and manufacture of a prototype to plotting, visualization and analysis of 2-D functions, developed using electronic components and Arduino. The proposal is to train and encourage educators to use the tool like a support in the teach process, encouraging students to learn about the correlate areas of the prototype and also collaborating their to visualization abstract mathematical concepts.

**Keywords:** Mathematics teaching, prototype, functions teaching.

# Sumário

<b>Resumo</b>	<b>iv</b>
<b>Abstract</b>	<b>v</b>
<b>Lista de figuras</b>	<b>ix</b>
<b>Lista de tabelas</b>	<b>x</b>
<b>Introdução</b>	<b>1</b>
<b>1 Funções, seus gráficos e sua relação com o cálculo numérico</b>	<b>6</b>
1.1 Funções . . . . .	7
1.2 Produto cartesiano . . . . .	7
1.3 O Plano numérico $\mathbb{R}^2$ . . . . .	8
1.4 Função afim . . . . .	10
1.5 Função polinomial . . . . .	11
1.5.1 Trinômio de segundo grau (polinômio de grau 2) . . . . .	11
1.6 Função exponencial e logarítmica . . . . .	13
1.7 Cálculo numérico . . . . .	15
1.8 Método de Newton-Raphson . . . . .	16
1.8.1 Algoritmo do Método de Newton-Raphson . . . . .	18
1.9 Método da Bissecção . . . . .	19
1.9.1 Algoritmo do Método da Bissecção . . . . .	22
1.10 Considerações finais do capítulo . . . . .	23
<b>2 Arduino, componentes eletroeletrônicos e o protótipo</b>	<b>25</b>
2.1 Arduino . . . . .	26

2.2	Componentes eletroeletrônicos . . . . .	28
2.2.1	LEDs . . . . .	28
2.2.2	Resistor . . . . .	29
2.2.3	LCD . . . . .	30
2.3	Protótipo . . . . .	32
2.4	Considerações finais do capítulo . . . . .	36
<b>3</b>	<b>Utilizando o protótipo</b>	<b>37</b>
3.1	Protótipo . . . . .	37
3.2	Proposta de atividades para sala de aula . . . . .	39
<b>4</b>	<b><i>Maxima</i></b>	<b>42</b>
4.1	Considerações finais do capítulo . . . . .	45
	<b>Considerações finais</b>	<b>47</b>
	<b>Apêndice: Código do programa do protótipo</b>	<b>52</b>

# Lista de Figuras

1.1	Ponto $P$ do sistema de eixos $OXY$ . . . . .	9
1.2	Gráfico da função $g$ , tal que $g(x) = 2x + 3$ . . . . .	10
1.3	Gráfico de uma função quadrática. Com $F$ indicando o foco, $B$ o vértice e o eixo “ $OX$ ” a reta diretriz $d$ . . . . .	12
1.4	Gráfico da função $f$ , tal que $f(x) = x^2 - 3x - 4$ . . . . .	13
1.5	Curvas características da função exponencial. . . . .	13
1.6	Curvas características da função logarítmica. . . . .	14
1.7	Gráficos das funções $f$ e $g$ , tal que $f(x) = 3^x$ e $g(x) = \log_3 x$ . . . . .	15
1.8	Interpretação geométrica do método de Newton-Raphson. . . . .	17
1.9	Gráfico das iterações realizadas para a função $f$ , com as respectivas retas tangentes em cada ponto da iteração e ponto de interseção entre elas (retas tangentes) e a função $f$ . . . . .	18
1.10	Método da Bissecção. . . . .	20
1.11	Gráfico das iterações realizadas para a função $f$ , com as respectivas aproximações. . . . .	22
2.1	Projetos construídos com Arduino. . . . .	26
2.2	Placa do Arduino UNO. . . . .	27
2.3	Tela do Arduino IDE. . . . .	28
2.4	Símbolo do LED, com 1 representando o anodo e 2 o catodo. . . . .	29
2.5	LED vermelho. . . . .	29
2.6	Símbolo do resistor. . . . .	29
2.7	Resistor de filme plástico. . . . .	30
2.8	Código de Cores. Fonte: AudioGroup (2017). . . . .	30
2.9	<i>Display</i> LED 16x2. . . . .	31
2.10	<i>Push-button</i> de 2 terminais. . . . .	31

2.11	Circuito de um das 12 teclas utilizadas no teclado. . . . .	32
2.12	Componentes do protótipo. . . . .	32
2.13	Conexões da matriz de LEDs. . . . .	33
2.14	Visão superior do painel manufaturado com todos os LEDs acesos. . . . .	34
2.15	Circuito do teclado e ligação com o Arduino. . . . .	35
2.16	Conexões elétricos necessárias entre o LCD e o Arduino Mega. . . . .	35
2.17	Protótipo construído. . . . .	36
3.1	Traçado da função $f$ , tal que $f(x) = -x$ . . . . .	38
3.2	Traçado da função $f$ , tal que $f(x) = x - 2$ . . . . .	39
3.3	Traçado da função $f$ , tal que $f(x) = 4$ . . . . .	39
4.1	Método de Newton-Raphson para a função $g$ , tal que $g(x) = x^3 - 5$ . . . . .	43
4.2	Método de Newton-Raphson para a função $g$ , tal que $g(x) = \frac{\log(x)}{x} + 2x$ . . . . .	43
4.3	Método da Bissecção para a função $f$ , tal que $f(x) = x^3 - 5$ . . . . .	44
4.4	Método da Bissecção para a função $g$ , tal que $g(x) = \frac{\log(x)}{x} + 2x$ . . . . .	44
4.5	Exibe comando de plotagem de gráfico 2D e janela com o gráfico da função $f$ , tal que $f(x) = x^3 - 5$ . . . . .	45
4.6	Exibe comando de plotagem de gráfico 2D e janela com o gráfico da função $g$ , tal que $g(x) = \frac{\log(x)}{x} + 2x$ . . . . .	45

# Lista de Tabelas

- 1.1 Resultado do cálculo das variáveis e funções relacionadas ao Método de Newton-Raphson para encontrar zero da função  $f$ , tal que  $f(x) = x^3 - 5$ . . 17
- 1.2 Intervalos e aproximações para as iterações do Método da Bissecção para encontrar zero da função  $f$ , tal que  $f(x) = x^3 - 5$ . . . . . 21

# Introdução

As funções matemáticas estão presentes em vários ramos das atividades humanas, desde atividades rotineiras até estudos mais complexos em diversas áreas do conhecimento, como na física, química, biologia, economia, etc. A disponibilidade de recursos tecnológicos e as transformações nos padrões de engajamento social e interativo dos adolescentes estão exigindo que as estratégias de ensino se modernizem, levando em conta o desenvolvimento tecnológico e o interesse individual atual, de forma que o estudo se torne interessante e possa desenvolver habilidades multidisciplinares concomitantemente, como eletrônica, programação e a própria matemática.

A Legislação Educacional Brasileira, como forma de desenvolvimento cognitivo aos alunos, inclui o uso das tecnologias na Lei de Diretrizes e Bases, nº 9394/96, e nos Parâmetros Curriculares Nacionais, 2002, como forma de capacitá-los para aquisição e o desenvolvimento de novas competências, para que ele seja capaz de responder a novos ritmos e processos, exigências da vida social e profissional moderna.

Nesse sentido, alguns estudiosos da educação, exibem em seus trabalhos a função do professor nesse processo. Segundo Bettega (2010), a tecnologia deve servir para enriquecer o ambiente educacional, propiciando a construção de conhecimentos por meio de uma atuação ativa, crítica e criativa por parte de alunos e professores. Já Golbert (2002), defende a ideia de que cabe ao professor ajudar os estudantes a adquirir as ferramentas culturais que lhes possibilitem refletir sobre suas próprias intuições e experiências e comunicá-las, articulando suas ideias, construindo compreensões mais ricas. Isto significa que é da competência do professor encontrar os meios de transpor a distância entre a linguagem usual dos alunos e as disciplinas do ensino clássico.

Existem vários estudos que relaciona a utilização da computação, eletrônica e robótica educacional como ferramenta e meio de aprendizagem para todos os níveis de

ensino. Rodarte (2014) traz em sua dissertação, as percepções que obteve com a utilização da robótica no ensino fundamental, mais especificamente no ensino de ângulos e geometria plana, no 7<sup>o</sup> ano do Ensino Fundamental. Rodrigues (2015), em seu trabalho, apresenta também os resultados obtidos com a utilização da plataforma LEGO® numa atividade desenvolvida para o ensino fundamental. Almeida (2017) exhibe em seu trabalho a plataforma Arduino e alguns exemplos práticos de projetos que podem ser utilizados no ensino da matemática.

Diante dessa realidade, idealizou-se um projeto de desenvolvimento de um protótipo para utilização no ensino da matemática na educação básica, mais especificamente para o estudo de funções. A pesquisa inclui o projeto e construção de uma estrutura para plotagem de gráficos, ou seja, visualização de um gráfico, por meio de um painel formado por LEDs, o qual se tem o controle pontual do acendimento de cada LED, que representa um valor definido em relação aos dois eixos cartesianos, ou seja, cada LED representará uma coordenada no gráfico.

Além do painel de LEDs, o protótipo é formado também por uma IHM (Interface Homem-Máquina), ferramenta que possibilitará que o usuário insira a função desejada num teclado, e em seguida poderá visualizar o traçado da função no painel e algumas informações sobre ela em um pequeno display.

O documento é instrutivo para construção e programação do protótipo, que pode ser desenvolvido por alunos e/ou professores para montagem, exibição e utilização em sala de aula, bem como para participação em feiras tecnológicas.

O desenvolvimento de ferramentas tecnológicas para o ensino além de auxiliar o aprendizado do aluno, possibilita à eles a participação em eventos nacionais e/ou internacionais como a *Campus Party*, que é o principal evento de internet e tecnologia do país. Ela ocorre no Brasil desde 2008 e é responsável por divulgar as tecnologias mais avançadas, assim como suas aplicações nos campos da Educação, da Economia, do Trabalho e da Cultura Digital. Ao longo do evento são realizadas palestras, debates, competições e oficinas que permitem que os frequentadores consigam interagir, compartilhar conhecimento, produzir novidades e acompanhar as principais tendências da atualidade (Campus Party, 2018, ver).

Ou ainda capacitam também os alunos a participarem da FEBRACE (Feira Brasileira de Ciências e Engenharia), um movimento nacional de estímulo ao jovem cientista,

que ocorre no Brasil desde 2003, com intuito de estimular uma cultura investigativa, de criatividade, inovação e empreendedorismo na Educação Básica brasileira, por meio da indução à realização de projetos e mostras científicas e tecnológicas nas escolas. Podem submeter projetos para a FEBRACE estudantes que estejam cursando o 8° ou 9° Ano do Ensino Fundamental, Ensino Médio ou o Ensino Técnico (FEBRACE, 2018, ver).

Além de medalhas e troféus para os melhores finalistas de cada categoria, os estudantes também podem receber equipamentos, certificados de diversas organizações em várias áreas científicas e algumas empresas, e demais prêmios concedidos com apoio de parcerias com a iniciativa privada.

A FEBRACE concede medalhas e troféu para os destaques nas seguintes categorias: Ciências Agrárias, Ciências Biológicas, Ciências da Saúde, Ciências Exatas e da Terra, Ciências Humanas, Ciências Sociais Aplicadas, Engenharia. Também recebem destaque os melhores finalistas em Votação Popular, Destaque em Criatividade, Destaque em Inovação, Destaque em Relevância Social, Destaque em Rigor Científico e Destaque em Empreendedorismo. A premiação da FEBRACE também concede o credenciamento para participação como finalista em outras feiras.

A robótica educacional permite também a participação na OBR (Olimpíada Brasileira de Robótica), que tem duas modalidades independentes: teórica e prática, e destina-se a todos os estudantes de qualquer escola, pública ou privada, do ensino fundamental, médio ou técnico em todo o território nacional (OBR, 2018, ver).

A OBR tem o objetivo de estimular os jovens às carreiras científico-tecnológicas, identificar jovens talentosos e promover debates e atualizações no processo de ensino-aprendizagem brasileiro. Ela ocorre desde 2006 e atualmente é considerado o maior evento de robótica da América Latina e classifica equipes para a RoboCup, maior evento de robótica do mundo.

O evento procura adequar-se tanto ao público que nunca viu robótica quanto ao público de escolas que já têm contato com a robótica educacional. As atividades acontecem através de competições práticas (com robôs) e provas teóricas em todo o Brasil. A prova teórica não exige do aluno o conhecimento específico de robótica. Na primeira fase, as provas são feitas na escola de cada estudante inscrito e as questões vão envolver problemas práticos do dia a dia a partir da robótica. Já a prática traz um desafio: equipes de até quatro alunos terão que construir um robô totalmente autônomo, ou seja, sem controle

remoto, e que seja capaz de navegar por um terreno acidentado, localizar vítimas e resgatá-las. Os inscritos nessa modalidade terão que participar de eventos regionais e, de acordo com sua pontuação, se classificarão às etapas estaduais e à final nacional.

Há também eventos internacionais que os alunos poderão competir como a LARC (Competição Latino Americana de Robôs), IEEE (Competição Latino Americana de Robótica), WRE (Workshop de Robótica na Educação), WRO (*World Robot Olympiad*), RoboCup, entre outros eventos e feiras locais, estaduais e internacionais de robótica e/ou ensino.

Alguns exemplos de participação e premiação nesses eventos são listados a seguir:

- o trabalho “A robótica educacional como intersecção entre as disciplinas técnicas do curso de informática” que ficou em 3<sup>o</sup> lugar geral na Febrace 2015 (IFRN, 2015, ver).
- O Certificado de Excelência recebido por alunos do PRONATEC, no Japão, em 2015, devido ao trabalho de robótica apresentado no país sobre o serviço de tratamento médico infantil (Brasil, 2015, ver).
- Medalha de ouro conquistada pelo estudante de 10 anos na “*K\*Bot World Championship*”, realizada nos EUA em 2017 (Porto, 2017, ver).

As oportunidades de exposição, competição e compartilhamento de informações são diversas, já existem diversos eventos consolidados em todo o mundo, o incentivo à participação é de fundamental importância para o desenvolvimento do espírito de pesquisador dos alunos ainda no ensino básico, para que eles se sintam motivados à participarem dos eventos e produzirem materiais e conteúdos que contribuam para o seu aprendizado e crescimento da ciência no país.

Posto isso, o objetivo desse documento é a capacitação e o incentivo de professores e/ou alunos para o desenvolvimento e utilização de ferramentas tecnológicas no ensino, seja ela replicação desse protótipo ou desenvolvimento de outros mais amplos ou diferenciados, contribuindo para o ensino da matemática e também contribuindo para a participação em eventos, feiras, mostras e competições tecnológicas.

Além do detalhamento do protótipo, este trabalho dá um ponta-pé inicial para o desenvolvimento de protótipos para estudo de funções mais complexas: são mostrados os principais métodos numéricos para cálculo de raíz(es) de funções não lineares e também

seus algoritmos. Foram desenvolvidos os códigos dos métodos estudados no software *Maxima*, no qual se pode observar e calcular raízes de funções e o traçado de seus respectivos gráficos.

Num trabalho mais amplo, tais métodos poderão ser aliados à plataforma Arduino para desenvolvimento de um protótipo que comporte mais “*pixels*”, ou seja, tenha mais LEDs, e possa exibir com uma resolução maior os variados tipos de curvas e que também resolva os mais variados tipos de funções.

Para exposição da proposta desse trabalho, ele foi dividido em introdução, três capítulos e considerações. No primeiro capítulo dessa dissertação será apresentada funções e características dos seus respectivos gráficos; trará também os métodos numéricos utilizados para encontrar zero(s) de funções, métodos que podem ser implementados para estudos de funções não lineares. Já no segundo capítulo será apresentada a plataforma Arduino e suas especificidades, os componentes eletrônicos utilizados e seu funcionamento, e também será detalhada a manufatura e programação do protótipo. No terceiro capítulo serão exibidos os resultados obtidos com a utilização do protótipo e no quarto capítulo os resultados obtidos com a utilização de métodos numéricos para estudo de funções no software *Maxima*; por fim as considerações finais do presente trabalho.

# Capítulo 1

## Funções, seus gráficos e sua relação com o cálculo numérico

O conceito de função está relacionado à ideia de associação de um elemento a outro, segundo uma regra específica. Assim, por exemplo, podemos considerar o tamanho de uma população variando em função do tempo, ou associado ao tempo e a sua localização, ou a qualquer outro fator que interfira na população em estudo; o preço de um produto pode estar associado apenas ao seu custo de produção, ou ao seu custo e demanda, ou ainda, ao seu custo, à demanda e outros fatores; e assim por diante.

Já o gráfico de uma função real descreve uma curva no plano, a qual representa o seu comportamento e facilita muito o seu entendimento. Ele nos dá uma imagem proveitosa da “história de vida” de uma função. Uma vez que a coordenada  $y$  de qualquer ponto  $(x, y)$  sobre o gráfico é  $y = f(x)$ , por exemplo, se os valores de  $f(x)$  são não negativos para todo  $x$  no domínio de  $f$ , podemos entender o valor de  $f(x)$  como a altura do ponto no gráfico acima de  $x$ . O gráfico de  $f$  nos permite visualizar o domínio sobre o eixo  $x$  e a imagem sobre o eixo  $y$ . É, portanto, uma ferramenta básica no estudo de cálculo e outras disciplinas da matemática.

Cálculo numérico é a disciplina, tradicionalmente vista no ensino superior, que estuda as técnicas para a solução aproximada de problemas matemáticos. Estas técnicas são de natureza analítica e computacional.

Devido, principalmente, ao aumento contínuo da capacidade e velocidade de processamento das máquinas, o desenvolvimento de métodos numéricos tornou a simulação computacional de problemas matemáticos uma prática usual nas mais diversas áreas ci-

entíficas e tecnológicas.

São utilizados para resolver problemas específicos como, por exemplo: resolver equações algébricas, resolver sistemas de equações lineares, interpolar e ajustar pontos, calcular derivadas e integrais, resolver equações diferenciais ordinárias, etc.

Neste trabalho, será apresentado o Método de Newton-Raphson e de Bissecção, para encontrar raíz(es) de funções não lineares, complementando assim o estudo de funções e de seus gráficos.

Foram utilizados como referenciais bibliográficos para escrita desse capítulo os seguintes autores: Lima (2013), Guidorizzi (2013) e Stewart (2013).

## 1.1 Funções

Segundo Guidorizzi (2013), *uma função*  $f$  é uma terna  $(A, B, a \mapsto b)$ , em que  $A$  e  $B$  são dois conjuntos e  $a \mapsto b$  uma regra que nos permite associar a cada elemento  $a$  de  $A$ , um único  $b$  de  $B$ . O conjunto  $A$  é o *domínio* de  $f$  e indica-se por  $Df$ , assim  $A = Df$ . O conjunto  $B$  é o *contradomínio* de  $f$ . O único  $b$  de  $B$  associado ao elemento  $a$  de  $A$  é indicado por  $f(a)$ ; dizemos que  $f(a)$  é o valor que  $f$  assume em  $a$  ou que  $f(a)$  é o valor que  $f$  associa a  $a$ . A imagem de  $f$  é o conjunto de todos os valores possíveis de  $f(x)$  quando  $x$  varia por todo o domínio.

Segundo Stewart (2013), podemos representar uma função de quatro maneiras diferentes: por uma equação, por uma tabela, por um gráfico ou mesmo por meio de palavras. Sendo que a equação matemática e o gráfico são as formas mais utilizadas no estudo matemático.

## 1.2 Produto cartesiano

Um *par ordenado*  $p = (x, y)$  é formado por um objeto  $x$ , chamado a primeira coordenada de  $p$  e um objeto  $y$ , chamado a segunda coordenada de  $p$ . Dois pares ordenados  $p = (x, y)$  e  $q = (u, v)$  serão chamados iguais quando  $x = u$  e  $y = v$ , isto é, quando tiverem a mesma primeira coordenada e a mesma segunda coordenada. O *produto cartesiano*  $X \times Y$  de dois conjuntos  $X$  e  $Y$  é o conjunto  $X \times Y$  formado por todos os pares ordenados  $(x, y)$

cuja primeira coordenada  $x$  pertence a  $X$  e cuja segunda coordenada  $y$  pertence a  $Y$ . Simbolicamente:

$$X \times Y = \{(x, y); x \in X, y \in Y\}.$$

Segundo Lima (2013), se  $X = \{x_1, \dots, x_m\}$  e  $Y = \{y_1, \dots, y_p\}$  são conjuntos finitos com  $m$  e  $p$  elementos respectivamente, então o produto cartesiano  $X \times Y$  é finito e possui  $mp$  elementos. Noutras palavras  $n(X \times Y) = n(X).n(Y)$ , com  $n(X)$  representando o número de elementos pertencentes a  $X$  e  $n(Y)$  o número de elementos pertencentes a  $Y$ . A melhor maneira de enxergar isto é pensar no produto cartesiano  $X \times Y$  como um quadrado retangular

$$\begin{array}{c} (x_1, y_1)(x_1, y_2) \dots (x_1, y_p) \\ (x_2, y_1)(x_2, y_2) \dots (x_2, y_p) \\ \vdots \\ (x_m, y_1)(x_m, y_2) \dots (x_m, y_p). \end{array}$$

com  $p$  colunas, cada uma das quais possui  $m$  elementos.

O produto cartesiano  $X \times Y$  acha-se intimamente ligado à ideia de relação binária. Uma relação (binária)  $R$  entre elementos do conjunto  $X$  e elementos do conjunto  $Y$  é uma condição ou um conjunto de condições que permitem determinar, dados  $x \in X$  e  $y \in Y$ , se  $x$  está ou não relacionado com  $y$  segundo  $R$ . No caso afirmativo, escreve-se  $xRy$ .

Um caso particularmente importante no estudo de funções é a relação funcional. Ela ocorre quando se tem uma função  $f : X \rightarrow Y$ . Diz-se então que o elemento  $x \in X$  está relacionado com o elemento  $y \in Y$  quando  $y = f(x)$ . Neste caso, não se costuma escrever  $xfy$  como se faria numa outra relação qualquer. Põe-se apenas  $y = f(x)$ .

O gráfico de uma relação  $R$  entre dois conjuntos  $X$  e  $Y$  é o subconjunto  $G(R)$  do produto cartesiano  $X \times Y$  formado pelos pares  $(x, y)$  tais que  $xRy$ . Assim,  $G(R) = \{(x, y) \in X \times Y; xRy\}$ .

### 1.3 O Plano numérico $\mathbb{R}^2$

Um dos exemplos mais importantes de produto cartesiano é o  $\mathbb{R}^2 = \mathbb{R} \times \mathbb{R}$ . Os elementos  $(x, y)$  de  $\mathbb{R}^2$  são os pares ordenados de números reais. Eles surgem como as coordenadas cartesianas de um ponto  $P$  do plano  $\Pi$  ( $x =$  abcissa,  $y =$  ordenada) quando

se fixa nesse plano um par de eixos ortogonais  $OX$  e  $OY$ , que se intersectam no ponto  $O$ , chamado a origem do sistema de coordenadas.

Dado o ponto  $P \in \Pi$ , a abscissa de  $P$  é o número  $x$ , coordenada do pé da perpendicular baixada de  $P$  sobre o eixo  $OX$ , enquanto a ordenada de  $P$  é a coordenada  $y$  do pé da perpendicular baixada de  $P$  sobre o eixo  $OY$ . Diz então que  $(x, y)$  é o par de coordenadas do ponto  $P$  relativamente ao sistema de eixos  $OXY$ . A Figura 1.1 ilustra graficamente o ponto  $P$ .

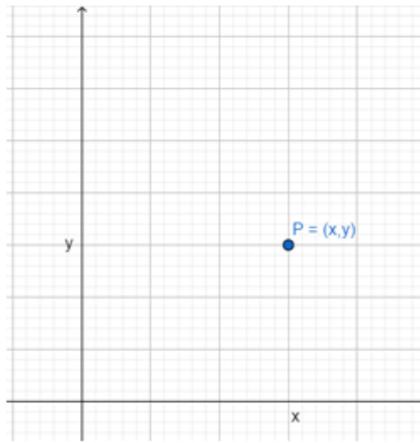


Figura 1.1: Ponto  $P$  do sistema de eixos  $OXY$ .

Os eixos  $OX$  e  $OY$  dividem o plano em quatro regiões, chamadas de *quadrantes*, que são numeradas no sentido anti-horário e começa pela região do espaço com  $x \geq 0$  e  $y \geq 0$ . Dessa forma, temos que o primeiro quadrante é formado pelos pontos cujas coordenadas apresentam  $x \geq 0$  e  $y \geq 0$ ; o segundo quadrante  $x \leq 0$  e  $y \geq 0$ ; o terceiro quadrante  $x \leq 0$  e  $y \leq 0$  e o quarto quadrante  $x \geq 0$  e  $y \leq 0$ .

A função  $f : \Pi \rightarrow \mathbb{R}^2$ , que associa a cada ponto  $P$  do plano  $\Pi$  seu par de coordenadas  $f(P) = (x, y)$  relativamente ao sistema de eixos  $OXY$ , é uma correspondência biunívoca. Ela permite traduzir conceitos e propriedades geométricas para uma linguagem algébrica e, reciprocamente, interpretar geometricamente relações entre números reais. Podemos então dizer que  $\mathbb{R}^2$  é o modelo aritmético do plano  $\Pi$  enquanto  $\Pi$  é o modelo geométrico de  $\mathbb{R}^2$ .

## 1.4 Função afim

Uma função  $f : \mathbb{R} \rightarrow \mathbb{R}$  chama-se *afim* quando existem constantes  $a, b \in \mathbb{R}$  tais que  $f(x) = ax + b$  para todo  $x \in \mathbb{R}$ . O gráfico da função afim com domínio e contradomínio em  $\mathbb{R}$  é uma reta com taxa de variação igual a  $a$  e que intercepta os eixos coordenados  $X$  e  $Y$  nos pontos  $\left(\frac{-b}{a}, 0\right)$  e  $(0, b)$ , respectivamente.

Quando  $b = 0$ , a função é chamada linear e é o modelo matemático para os problemas de proporcionalidade. Uma proporcionalidade é uma função  $f : \mathbb{R} \rightarrow \mathbb{R}$  tal que, para quaisquer números reais  $c, x$  tem-se  $f(cx) = cf(x)$  (proporcionalidade direta) ou  $f(cx) = \frac{f(x)}{c}$ , se  $c \neq 0$  (proporcionalidade inversa).

Já quando  $a = 0$ , a função é chamada de função constante, daí, a função  $f : \mathbb{R} \rightarrow \mathbb{R}$  é tal que, para qualquer real  $b, x$  tem-se  $f(x) = b$ .

É possível ainda determinar se uma função é afim, sem que os coeficientes  $a$  e  $b$  sejam fornecidos explicitamente. Nesse caso, obtém-se  $b$  como o valor que a função dada assume quando  $x = 0$ . O número  $b = f(0)$  às vezes se chama o valor inicial da função  $f$ . Quanto ao coeficiente  $a$ , ele pode ser determinado a partir do conhecimento dos valores  $f(x_1)$  e  $f(x_2)$  que a função  $f$  assume em dois pontos distintos e arbitrários  $x_1$  e  $x_2$ .

Vamos analisar a função  $g(x) = 2x + 3$ . Temos que a função intercepta o eixo  $X$  no ponto  $\left(\frac{-3}{2}, 0\right)$  e o eixo  $Y$  no ponto  $(0, 3)$ . O que nos fornece o gráfico mostrado na Figura 1.2.

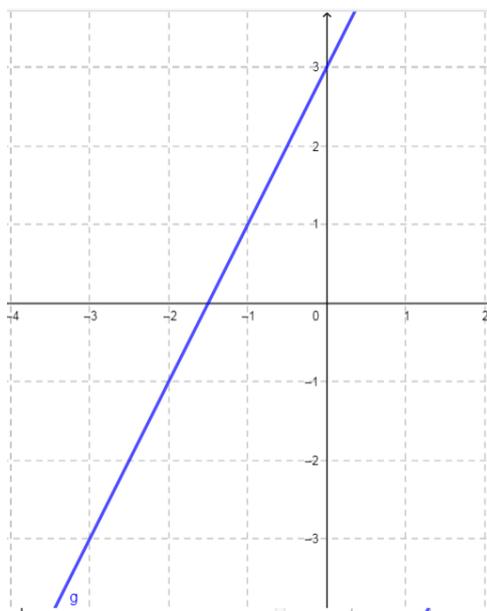


Figura 1.2: Gráfico da função  $g$ , tal que  $g(x) = 2x + 3$ .

## 1.5 Função polinomial

Uma função  $p : \mathbb{R} \rightarrow \mathbb{R}$  é chamada de *polinomial* quando são dados números reais  $a_0, a_1, \dots, a_n$  tais que, para todo  $x \in \mathbb{R}$ , temos

$$p(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0.$$

Se  $a_n \neq 0$ , dizemos que  $p$  tem grau  $n$ .

Segundo Stewart (2013), os polinômios são usados comumente para modelar várias quantidades que ocorrem em ciências sociais e naturais. Um exemplo é a utilização, por parte dos economistas, de polinômio para representar o custo na produção de  $x$  unidades de um produto.

Os métodos que se usam atualmente para determinar uma raiz do polinômio  $p$  localizada no intervalo  $[a, b]$  quando se sabe que  $p(a)$  e  $p(b)$  têm sinais opostos baseiam-se em algoritmos aproximativos, os quais instruem, passo a passo, como proceder para obter uma sequência de números  $x_1, x_2, \dots, x_n$  tais que os valores  $p(x_1), p(x_2), \dots, p(x_n)$  estão cada vez mais próximos de zero.

### 1.5.1 Trinômio de segundo grau (polinômio de grau 2)

Uma função  $f : \mathbb{R} \rightarrow \mathbb{R}$ , de números reais  $a, b, c$ , com  $a \neq 0$ , tais que  $f(x) = ax^2 + bx + c$  para todo  $x \in \mathbb{R}$  é o trinômio de segundo grau, também chamado de *função quadrática*, ou somente polinômio de grau dois. O gráfico de uma função quadrática é uma parábola que abre-se para cima se  $a > 0$  e para baixo quando  $a < 0$ .

Dados um ponto  $F$  e uma reta  $d$  que não o contém, a parábola de foco  $F$  e diretriz  $d$  é conjunto de pontos do plano que distam igualmente de  $F$  e de  $d$ . A reta perpendicular à diretriz, baixada a partir do foco, chama-se o eixo da parábola. O ponto da parábola mais próximo da diretriz chama-se o vértice dessa parábola. Ele é o ponto médio do segmento cujas extremidades são o foco e a interseção do eixo com a diretriz. A Figura 1.3 ilustra graficamente a função quadrática e os parâmetros supramencionados.

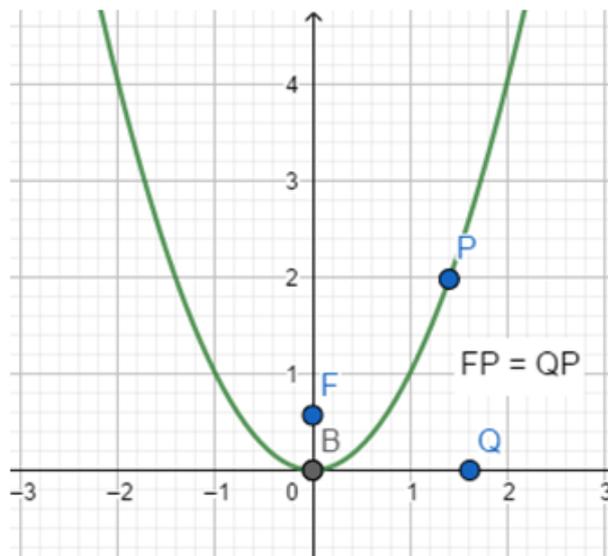


Figura 1.3: Gráfico de uma função quadrática. Com  $F$  indicando o foco,  $B$  o vértice e o eixo “OX” a reta diretriz  $d$ .

Temos que o ponto do gráfico de  $f(x) = ax^2 + bx + c$  mais próximo da diretriz é aquele de abscissa  $\frac{-b}{2a}$ . Nesse ponto,  $f$  atinge seu valor mínimo quando  $a > 0$  e seu valor máximo quando  $a < 0$ . Ainda quando  $x = \frac{-b}{2a}$ , o ponto  $(x, f(x))$  é o vértice da parábola que constitui o gráfico de  $f$ .

O gráfico da função quadrática é um elemento de grande importância para entender o comportamento dessa função. As abscissas  $x_1, x_2$  dos pontos onde esse gráfico intersecta o eixo  $OX$  são as raízes do trinômio.

O ponto médio do segmento  $[x_1, x_2]$  é a abscissa do vértice da parábola. Se o gráfico está inteiramente acima, ou inteiramente abaixo do eixo  $OX$ , a equação não possui raízes. Se o gráfico apenas tangencia o eixo  $OX$ , a equação tem uma raiz única. Se  $x_1 < x < x_2$  então  $f(x)$  tem sinal contrário ao sinal de  $a$ ; se  $x < x_1$  ou  $x > x_2$ ,  $f(x)$  tem o mesmo sinal de  $a$ .

Vamos analisar a função  $f$ , tal que  $f(x) = x^2 - 3x - 4$ . Como  $a > 0$ , a parábola apresentará sua concavidade voltada para cima, e a função apresenta raízes iguais a  $-1$  e  $4$ . Temos ainda que seu valor mínimo é  $x = \frac{3}{2}$ , o vértice da parábola é o ponto  $(1, 5; -6, 25)$ , e temos também que sua função cortará o eixo  $Y$  no ponto  $(0, -4)$ . A Figura 1.4 ilustra o gráfico dessa função.

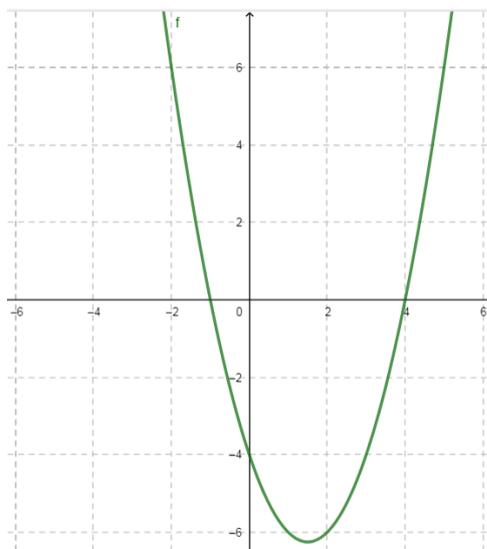
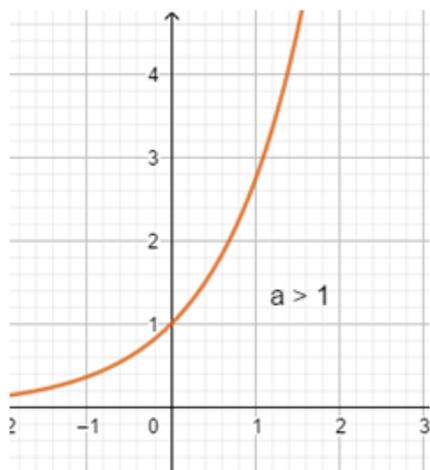


Figura 1.4: Gráfico da função  $f$ , tal que  $f(x) = x^2 - 3x - 4$ .

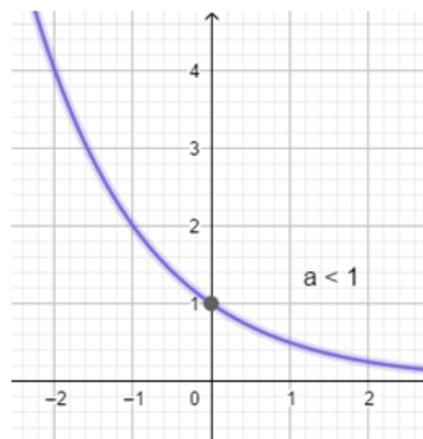
## 1.6 Função exponencial e logarítmica

Seja  $a$  um número real positivo e  $a \neq 1$ . A função  $f : \mathbb{R} \rightarrow \mathbb{R}^+$ , dada por  $f(x) = a^x$ , é chamada de *função exponencial de base  $a$* .

Os gráficos dessa função apresentam os traçados apresentados na Figura 1.5.



(a) para  $a > 1$



(b) para  $1 < a < 0$

Figura 1.5: Curvas características da função exponencial.

A função exponencial mais comum em aplicações é a função exponencial de base  $a = e$ , onde  $e$  é a constante de Euler. A função, nesse caso, é chamada de *função exponencial natural* ou, simplesmente *função exponencial*. Ela também pode ser representada na notação

$$f(x) = e^x.$$

A função exponencial de base  $a$ , definida anteriormente, admite uma função inversa  $g$  chamada de *função logarítmica na base  $a$* . Portanto,  $g : \mathbb{R}^+ \rightarrow \mathbb{R}$ . A regra de associação desta função é representada como  $g(x) = \log_a x$ .

Como as funções exponencial e logarítmica (ambas de base  $a$ ) são inversas uma da outra, segue pela definição de função inversa que

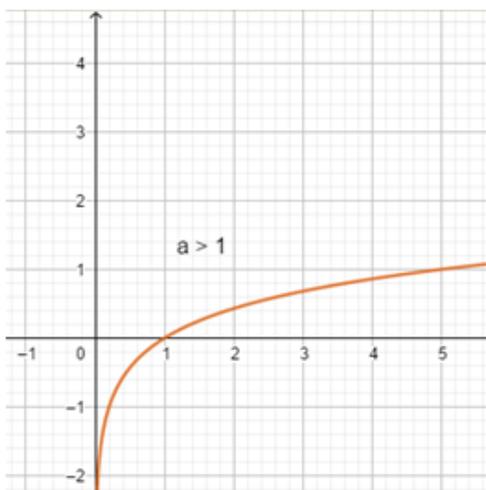
$$a^{\log_a x} = x, \text{ para todo } x \in \mathbb{R}^+$$

e

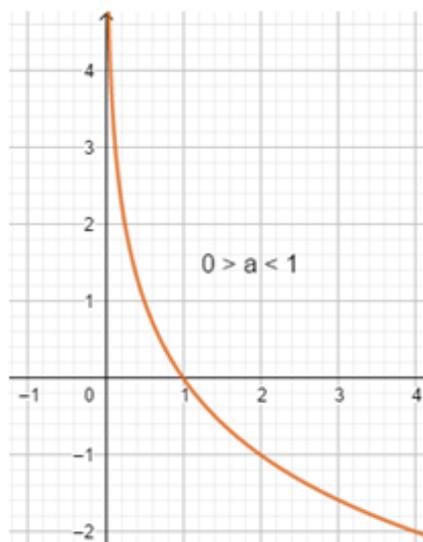
$$\log_a(a^x) = x, \text{ para todo } x \in \mathbb{R}.$$

Sendo assim, tomando  $x = 0$  e  $x = 1$ , obteremos  $\log_a 1 = 0$  e  $\log_a a = 1$ .

Seu gráfico é a reflexão do gráfico de  $y = a^x$  em torno da reta  $y = x$ . As suas curvas características são mostradas na Figura 1.6.



(a) para  $a > 1$



(b) para  $0 < a < 1$

Figura 1.6: Curvas características da função logarítmica.

A função inversa da exponencial natural é o *logaritmo natural*, ele pode ser representado da seguinte forma

$$\log_e x = \ln x.$$

Em particular, se fizermos  $x = e$ , obteremos  $\ln e = 1$ .

Vamos analisar a função exponencial  $f$ , tal que  $f(x) = 3^x$  e a função logarítmica  $g$ , tal que  $g(x) = \log_3 x$ . Temos que  $g$  é a inversa da função  $f$ , portanto, temos que seu

traçado é a reflexão do gráfico de  $f$  em torno da reta  $y = x$ . A Figura 1.7 ilustra o gráfico de ambas as funções.

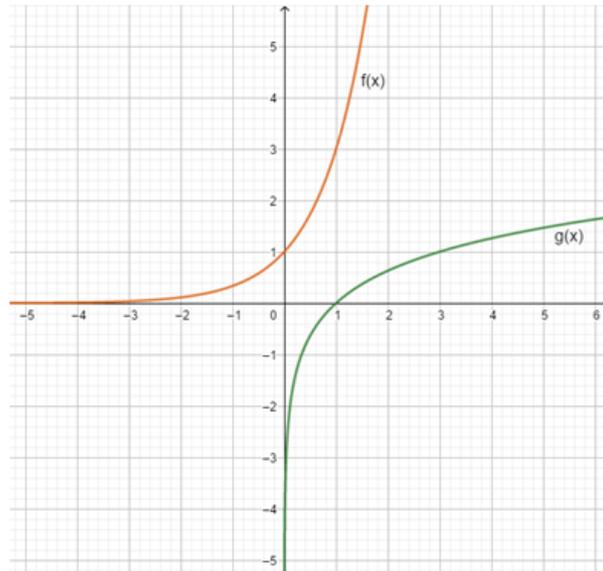


Figura 1.7: Gráficos das funções  $f$  e  $g$ , tal que  $f(x) = 3^x$  e  $g(x) = \log_3 x$ .

## 1.7 Cálculo numérico

Os métodos numéricos quando implementados numa lógica computacional se mostram eficientes no cálculo das raízes devido à velocidade de operação das máquinas, conseguindo realizar várias iterações num curto intervalo de tempo. Com isso, conseguimos obter raíz(es) aproximadas de funções não lineares com erros muito pequenos.

Quando a função é polinomial, de primeiro grau, o problema de encontrar zeros dessa função se reduz a encontrar as raízes da equação  $f(x) = ax + b = 0$ . Tradicionalmente, nesses casos, os estudantes aprendem a manipular algebricamente a equação, utilizando o algoritmo de isolar a incógnita. Daí obtemos que o zero da função será:

$$x = \frac{-b}{a},$$

desde que  $a \neq 0$ .

Já para as funções polinomiais de segundo grau, um dos métodos é a utilização da chamada “Fórmula de Bhaskara”. Se  $f : \mathbb{R} \rightarrow \mathbb{R}$  é tal que  $f(x) = ax^2 + bx + c$ , com  $a \neq 0$ , então os zeros da função serão:

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

desde que  $b^2 - 4ac \geq 0$ .

Para as demais funções não lineares, encontrar as suas raízes se torna mais complexo. Nesse sentido, os métodos numéricos oferecem recursos com os quais podemos nos aproximar da solução desejada.

## 1.8 Método de Newton-Raphson

A fim de tornar o entendimento do método simples, até mesmo para o Ensino Básico, esse trabalho se ateve somente ao entendimento gráfico do método. Para isso, o Método de Newton-Raphson, trabalha de forma iterativa, na busca do ponto de interseção entre o gráfico de uma função  $f : \mathbb{R} \rightarrow \mathbb{R}$  e a reta das abscissas ( $y = 0$ ).

Antes de apresentar o método propriamente dito, precisamos conhecer o conceito de *derivada* de uma função. A derivada está intimamente relacionado à taxa de variação instantânea de uma função. Se uma função  $f$  é definida em um intervalo aberto contendo  $x_0$ , então a derivada de  $f$  em  $x_0$  denotada por  $f'(x_0)$ , é dada por

$$f'(x_0) = \lim_{\Delta x \rightarrow 0} \frac{f(x_0 + \Delta x) - f(x_0)}{\Delta x}.$$

se esse limite existir, com  $\Delta x$  representando uma pequena variação em  $x$ , próximo de  $x_0$ .

A ideia geral do método é produzir, a partir de uma aproximação inicial conhecida ( $x_0$ ) e uma precisão  $\epsilon$ , uma seqüência de valores que convirja para a solução do problema.

Temos que a equação da reta tangente ao gráfico da função  $f$  no ponto  $(x_0, f(x_0))$ , tem o coeficiente angular igual a  $f'(x_0)$ , e é dada por:

$$y - f(x_0) = f'(x_0) \cdot (x - x_0)$$

Sabendo que essa reta passa por  $(x_1, 0)$ , temos que

$$0 - f(x_0) = f'(x_0) \cdot (x_1 - x_0).$$

Portanto,

$$x_1 = x_0 - \frac{f(x_0)}{f'(x_0)}.$$

De modo geral, temos

$$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}.$$

Ou seja, dada aproximação  $x_i$ , a próxima aproximação  $x_{i+1}$  é o ponto de interseção entre o eixo das abscissas ( $y = 0$ ) e a reta tangente ( $y$ ) ao gráfico da função  $f$  no ponto  $x = x_i$ .

As iterações ocorrerão até que  $|x_{i+1} - x_i| < \epsilon$ , ou seja, até  $f(x_i)$  se aproxime de  $y = 0$  o quanto se desejar. A Figura 1.8 mostra a interpretação geométrica do método.

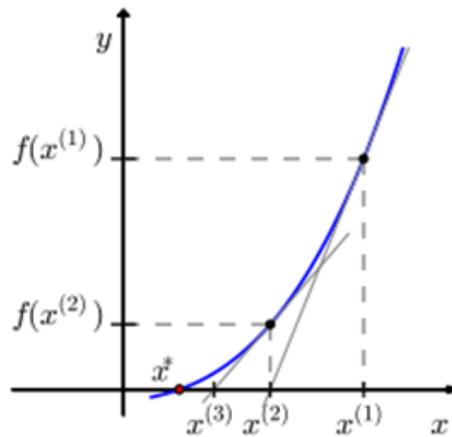


Figura 1.8: Interpretação geométrica do método de Newton-Raphson.

Vamos utilizar o método para encontrar as raízes da função  $f : \mathbb{R} \rightarrow \mathbb{R}$ , tal que  $f(x) = x^3 - 5$ , com precisão de  $\epsilon = 10^{-3}$  e utilizando o valor inicial de  $x_0 = 1$ .

Calculou-se cada ponto  $(x_i, f(x_i))$ , o coeficiente angular da reta tangente ( $f'(x)$ ) em cada ponto  $x_i$  e em seguida a reta tangente, determinando assim, sucessivamente, a aproximação  $x_{i+1}$ , até obtermos o mais próximo que se desejar  $f(x_i)$  de  $y = 0$ , ou seja,  $|x_{i+1} - x_i| < \epsilon$ .

A Tabela 1.1 mostra os resultados obtidos para a função  $f$  e a Figura 1.9 exibe o Gráfico das Iterações, com as funções e pontos de interesses destacados.

Tabela 1.1: Resultado do cálculo das variáveis e funções relacionadas ao Método de Newton-Raphson para encontrar zero da função  $f$ , tal que  $f(x) = x^3 - 5$ .

$i$	$(x_i, f(x_i))$	$f'(x_i) = 3x_i^2$	$y - f(x_i) = f'(x_i) \cdot (x - x_i)$	$x_{i+1} = x_i - \frac{f(x_i)}{f'(x_i)}$
0	(1;-4)	3	$y_0 = 3x - 7$	2,3333
1	(2,3333;7,7032)	16,3329	$y_1 = 16,3329x - 30,4064$	1,8617
2	(1,8617;1,4525)	10,3978	$y_2 = 10,3978x - 17,9051$	1,7220
3	(1,7220;0,1062)	8,8958	$y_3 = 8,8958x - 15,2124$	1,7101
4	(1,7101;0,0011)	8,7733	$y_4 = 8,7733x - 14,9922$	1,7100
5	(1,7100;0,0002)	8,7723	$y_5 = 8,7723x - 15,0008$	1,7100

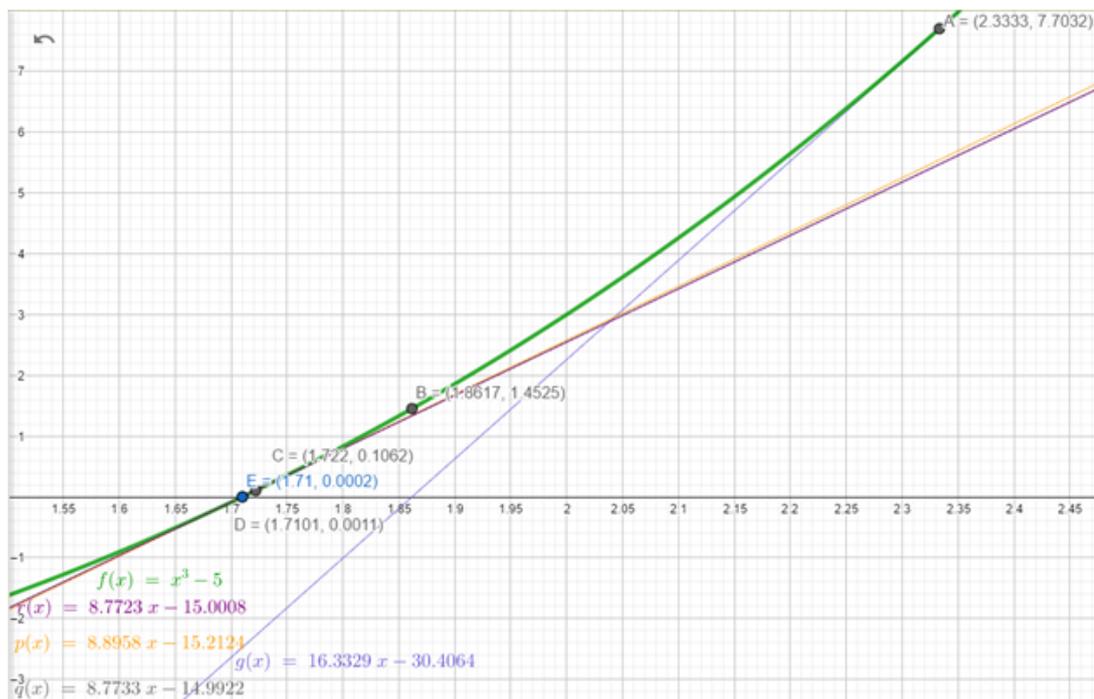


Figura 1.9: Gráfico das iterações realizadas para a função  $f$ , com as respectivas retas tangentes em cada ponto da iteração e ponto de interseção entre elas (retas tangentes) e a função  $f$ .

Como  $|x_5 - x_4| = 0,0001 < \epsilon$ , a iteração para e temos que  $x \cong x_5 = 1,7100$ .

### 1.8.1 Algoritmo do Método de Newton-Raphson

O algoritmo do método de Newton-Raphson, para encontrar zero(s) de funções, será da seguinte forma: primeiro será fornecido a função  $f$ , uma aproximação inicial " $x_0$ "; a precisão " $\epsilon$ "; e número máximo de iterações " $nmax$ " e o programa fornecerá como saída uma solução aproximada ou uma mensagem de erro. A lógica de programação para implementação do método segue o passo-a-passo abaixo:

Passo 1: declare

$x$ , (aproximação inicial para a raiz)

$x_0$ , (aproximação para a raiz, a cada iteração)

$\epsilon$ , (precisão requerida)

$i$  (contador de iterações)

$nmax$  (número máximo de iterações)

Passo 2: declare

$f$ , (função que se quer achar a raiz)

$d$ , (derivada da função  $f$ )

Passo 3: Faça  $i = 1$ ;

Passo 4: Repita  $x = x_0 - \frac{f(x_i)}{f'(x_i)}$ ;

Passo 5: Se  $|x - x_0| < \epsilon$  ou  $i = nmax \rightarrow$  então

interrompa

fim se

Passo 6:  $i \leftarrow i + 1$ ;

Passo 7:  $x_0 \leftarrow x$ ;

fim repita

Passo 8: se  $i > nmax$ , então

escreva "Convergência não obtida"

Passo 9: senão

escreva  $x, f(x)$

fim se

pare

## 1.9 Método da Bissecção

O método da bissecção explora o *Teorema do Valor Intermediário*, segundo o qual: "Seja uma função  $f$  contínua no intervalo fechado  $[a, b]$ , se  $y_0$  é um valor entre  $f(a)$  e  $f(b)$ , então existe pelo menos um  $x_0 \in [a, b]$ , tal que  $f(x_0) = y_0$ ."

Sendo assim, a ideia para aproximar o zero de  $f(x)$  é tomar como primeira aproximação o ponto médio do intervalo  $[a, b]$ , isto é:

$$x_0 = \frac{a + b}{2}.$$

Pode ocorrer de  $f(x_0) = 0$  e, nesse caso, o zero de  $f(x)$  é  $x^* = x_0$ . Caso contrário, se  $f(a).f(x_0) < 0$ , então  $x^* \in (a, x_0)$ . Nesse caso, tomamos como segunda aproximação do zero de  $f(x)$  o ponto médio do intervalo  $[a, x_0]$ , isto é,  $x_1 = \frac{a + x_0}{2}$ . No outro caso, se  $f(x_0).f(b) < 0$ , então, aproximação desejada é  $x_1 = \frac{b + x_0}{2}$ , que é o ponto médio do intervalo  $[x_0, b]$ .

O método ocorrerá de forma iterativa, a partir de um intervalo inicial dado  $[a, b]$  e uma precisão  $\epsilon$ . A cada iteração se determinará a aproximação  $x_i$  que é ponto médio do intervalo, a partir da qual permitirá que seja avaliado e determinado o intervalo da

iteração subsequente, no qual se tenha uma raiz da função, até que obtenha-se  $f(x_i)$  próximo de  $y = 0$  quanto se desejar, ou seja,  $|x_{i+1} - x_i| < \epsilon$ . A Figura 1.10 ilustra o funcionamento do método.

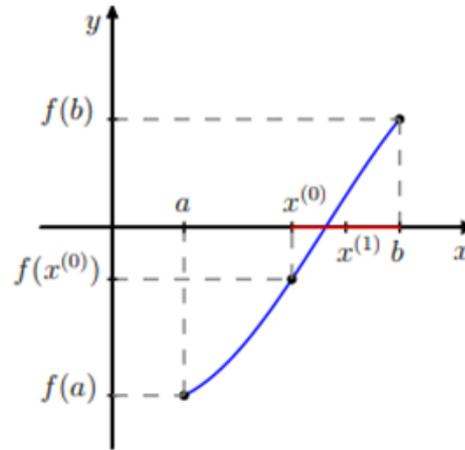


Figura 1.10: Método da Bissecção.

Vamos utilizar o método também para encontrar as raízes da função  $f$ , tal que  $f(x) = x^3 - 5$ , com precisão de  $\epsilon = 10^{-2}$ . Consideraremos o intervalo inicial  $I_0 = [1, 2]$ . Daí obtemos:

$$x_0 = \frac{a + b}{2} = \frac{1 + 2}{2} = 1,5.$$

Como  $f(1,5) = -1,625 < 0$ ,  $f(1) = -4 < 0$  e  $f(2) = 3 > 0$ . Então o novo intervalo  $I_1 = [1,5; 2]$ , daí:

$$x_1 = \frac{1,5 + 2}{2} = 1,75.$$

Como  $f(1,75) = 0,3594 > 0$ ,  $f(1,5) = -1,625 < 0$  e  $f(2) = 3 > 0$ . Então o novo intervalo  $I_2 = [1,5; 1,75]$ , daí:

$$x_2 = \frac{1,5 + 1,75}{2} = 1,625.$$

Como  $f(1,625) = -0,7090 < 0$ ,  $f(1,5) = -1,625 < 0$  e  $f(1,75) = 0,3594 > 0$ . Então o novo intervalo  $I_3 = [1,625; 1,75]$ , daí:

$$x_3 = \frac{1,625 + 1,75}{2} = 1,6875.$$

Como  $f(1,6875) = -0,1946 < 0$ ,  $f(1,625) = -0,7090 < 0$  e  $f(1,75) = 0,3594 > 0$ . Então o novo intervalo  $I_4 = [1,6875; 1,75]$ , daí:

$$x_4 = \frac{1,75 + 1,6875}{2} = 1,7188.$$

Como  $f(1,7188) = 0,0778 > 0$ ,  $f(1,6875) = -0,1946 < 0$  e  $f(1,75) = 0,3594 > 0$ . Então o novo intervalo  $I_5 = [1,6875; 1,7188]$ , daí:

$$x_5 = \frac{1,6875 + 1,7188}{2} = 1,7032.$$

Como  $f(1,7032) = -0,0592 < 0$ , e  $f(1,7188) = 0,0778 > 0$ . Então o novo intervalo  $I_6 = [1,7032; 1,7188]$ , daí:

$$x_6 = \frac{1,7032 + 1,7188}{2} = 1,7110.$$

Como  $|x_6 - x_5| = 0,0078 < \epsilon$ , a iteração para e temos que  $x \cong x_6 = 1,7110$ . A Tabela 1.2 e o gráfico da Figura 1.11 concatenam os cálculos das aproximações e sua visualização gráfica.

Tabela 1.2: Intervalos e aproximações para as iterações do Método da Bissecção para encontrar zero da função  $f$ , tal que  $f(x) = x^3 - 5$ .

$i$	$I_i$	$x_i$
0	$I_0 = [1; 2]$	A=1,5
1	$I_1 = [1,5; 2]$	B=1,75
2	$I_2 = [1,5; 1,75]$	C=1,625
3	$I_3 = [1,625; 1,75]$	D=1,6875
4	$I_4 = [1,6875; 1,75]$	E=1,7188
5	$I_5 = [1,6875; 1,7188]$	F=1,7032
6	$I_6 = [1,7032; 1,7188]$	G=1,7110

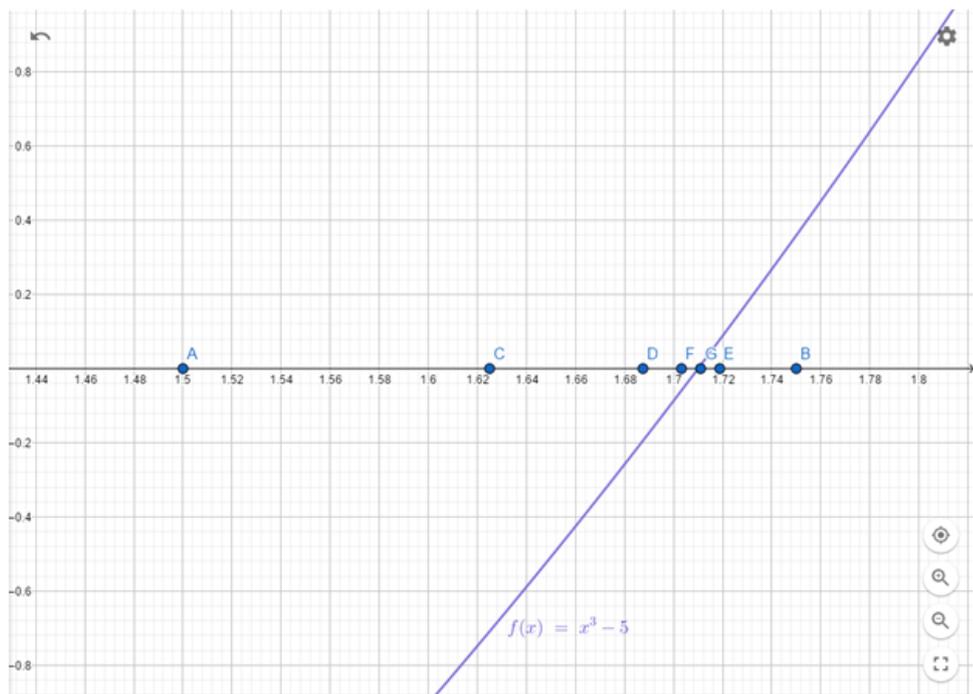


Figura 1.11: Gráfico das iterações realizadas para a função  $f$ , com as respectivas aproximações.

### 1.9.1 Algoritmo do Método da Bissecção

Seja  $f$  contínua em  $[a, b]$  e tal que  $f(a)$  e  $f(b)$  tenham sinais opostos, o algoritmo do método da bissecção para encontrar zeros de função terá como entrada a função “ $f$ ”, extremidades “ $a$ ” e “ $b$ ”, precisão “ $\epsilon$ ” e número máximo de iterações “ $nmax$ ”. Já a sua saída será a solução aproximada ou mensagem de erro. O passo-a-passo da lógica de programação é descrito a seguir:

Passo 1: declare

$a, b$ , (limites inferior e superior do intervalo)

$xm$ , (valor médio entre  $a$  e  $b$ )

$\epsilon$ , (precisão requerida)

$i$  (contador de iterações)

$nmax$  (número máximo de iterações)

Passo 2: declare

$f(x)$  (função que se quer achar raiz)

leia  $a, b, \epsilon$

se  $f(a).f(b) = 0$  então

Passo 3: escreva Valores de  $a$  e  $b$  inválidos

interrompa

fim se

$i \leftarrow 1$

Passo 4: repita

$$xm \leftarrow \frac{b+a}{2}$$

Passo 5: se  $|b-a| < \epsilon$  ou  $i > nmax$  então

interrompa

fim se

Passo 6: se  $f(xm).f(b) < 0$  então

$a \leftarrow xm$

Passo 7: senão

$b \leftarrow xm$

fim se

Passo 8:  $i \leftarrow i + 1$

fim repita

se  $i > nmax$  então

escreva Convergência não obtida

senão

escreva  $xm, f(xm)$

fim se

para

## 1.10 Considerações finais do capítulo

As funções são utilizadas para modelar matematicamente vários problemas do dia-a-dia e questões científicas, portanto, o conhecimento de seu gráfico é de extrema importância para estudo de seu comportamento.

Os métodos numéricos quando implementados numa lógica computacional, se mostram eficientes no cálculo das raízes devido à velocidade de operação das máquinas, conseguindo realizar várias iterações num curto intervalo de tempo. Com isso, conseguimos obter raiz(es) aproximada(s) de funções não lineares com erros muito pequenos.

Observamos que dos métodos numéricos supramencionados, o da Bissecção con-

vergiu mais lentamente que o de Newton-Raphson, ou seja, para encontrar uma raiz com o mesmo erro nos dois métodos, teríamos que realizar mais iterações utilizando da Bissecção. No entanto, o método da Bissecção é mais simples, já que não é necessário o cálculo da derivada função, somente é feito uma média aritmética simples.

Mas isso depende de condições iniciais ideais:

- Para o Método de Newton-Raphson: depende da escolha de uma boa aproximação inicial  $x_0$  e que a primeira derivada da função seja diferente de zero, ou seja,  $f'(x_i) \neq 0$ ;
- Para o Método da Bissecção: depende se a função é contínua no intervalo inicial escolhido  $[a, b]$ , e que exista uma raiz da função nele, ou seja,  $f(a) \cdot f(b) < 0$ .

Apesar das peculiaridades dos dois métodos, eles apresentam fácil visualização gráfica, o que pode facilitar o seu entendimento por alunos, inclusive do Ensino Médio.

## Capítulo 2

# Arduino, componentes eletroeletrônicos e o protótipo

Neste capítulo, será abordado a plataforma utilizada (Arduino), a qual apresenta um microcontrolador embarcado com núcleo de processador, memória e periféricos de entrada e saída num único circuito integrado. Esse pequeno computador pode ser utilizado para controle e leitura de sensores e também controle e ativação de atuadores, como motores, relés, sinalizadores, etc.

Devido seu baixo custo e facilidade de programação tem sido utilizado por projetistas, estudantes e curiosos da área para desenvolvimento de produtos e dispositivos automatizados nos mais diversos ramos, como por exemplo, I) automação residencial: controle de cortinas, portas, eletrodomésticos, portões, comedouros de animais de estimação, etc; II) controle de lavouras: umidade do solo, irrigação, supervisão aérea com drones, etc; III) desenvolvimento de robôs: seguidores de linha, competições de luta, competições de futebol, etc; IV) protótipos autônomos diversos: cubo de LEDs com efeitos em seu acendimento, jaqueta de LEDs para ciclistas, fileira de LEDs com técnica de POV (*Persistent of Vision*) que fornece como resultado um *display* esférico, como um globo girante, ou circular, como um relógio e seus ponteiros.

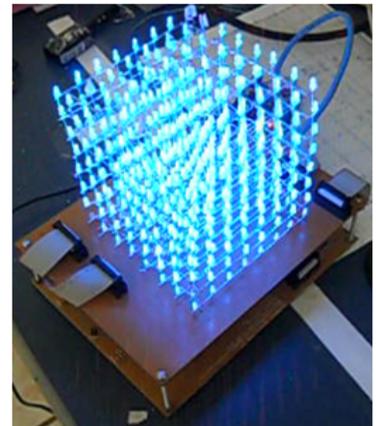
Temos vários exemplos de projetos utilizando Arduino. Um deles trata-se de um Seguidor Solar, que é um dispositivo para ser utilizado como suporte para placas solares, que a partir da comparação da leitura de sensores de luminosidade, há ativação de motores que permitem à estrutura que se movimente de acordo com a movimentação do Sol, com isso há um aumento na geração de energia se comparado ao modelo utilizado, que são as

placas fixas nos telhados.

Outro exemplo é o cubo de LEDs, de 8x8x8, desenvolvido a partir da interligação de 512 LEDs formando um cubo, o qual se tem o controle pontual de cada um deles, permitindo a programação de vários efeitos: como efeito chuva, acendimento de cada camada por vez, acendimento de cada linha por vez, acendimento de todos, escrita tridimensional de letras, etc. A Figura 2.1 exhibe ambos projetos.



(a) Seguidor Solar.



(b) Cubo de LEDs.

Figura 2.1: Projetos construídos com Arduino.

Neste capítulo ainda será abordado o funcionamento e especificações dos componentes eletroeletrônicos utilizados no projeto e as conexões necessárias entre eles para manufatura do protótipo, escopo desse trabalho.

## 2.1 Arduino

Para o controle, supervisão e ativação dos periféricos será utilizado o *Arduino*, que é uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR com suporte de *E/S(Entrada/Saída)* embutido, uma linguagem de programação padrão, a qual é essencialmente *C/C++*. (Arduino, 2017,ver).

A plataforma é uma ferramenta acessível, de baixo custo, flexível e fácil de se usar por profissionais e amadores. Pode ser usada para o desenvolvimento de objetos interativos independentes, ou ainda para ser conectado a um computador hospedeiro.

Uma típica placa Arduino é composta por um controlador, algumas *E/S* digitais e analógicas, além de uma interface serial ou USB, para interligar-se ao hospedeiro, que é usado para programá-la e interagi-la em tempo real. Usando-se extensões chamadas de Shields, é possível ainda conectá-las à rede, à outros Arduinos e dispositivos móveis, por vários protocolos de comunicação (*Bluetooth*, Internet, RF (rádio frequência), Xbee, etc).

A Figura 2.2 mostra uma das placas oficiais do Arduino: o Arduino UNO. No entanto, existe vários outros modelos, como por exemplo, o Arduino Mega que é uma placa que conta com mais *E/S* que o UNO, o Arduino Nano que apresenta dimensões mais compactas, etc.

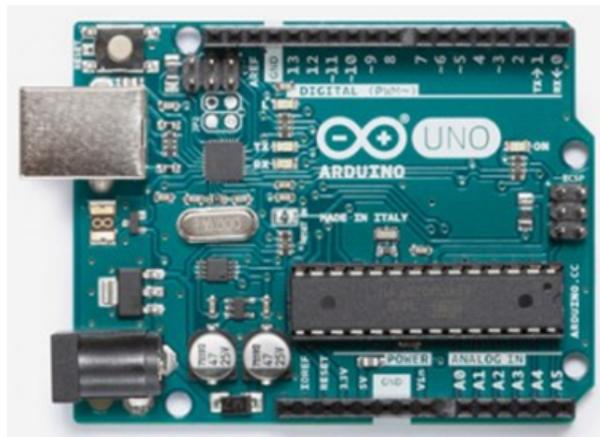


Figura 2.2: Placa do Arduino UNO.

A maior vantagem dessa plataforma de desenvolvimento sobre as demais é a facilidade de sua utilização, pois, pessoas que não são da área técnica podem aprender o básico e criar seus próprios projetos em um intervalo de tempo relativamente curto.

Além do *hardware*, há o Arduino IDE (Ambiente de Desenvolvimento Integrado) que é uma aplicação multiplataforma escrita em Java e esquematizado para introduzir a programação aos seus utilizadores, sejam eles profissionais ou também pessoas não familiarizadas com o desenvolvimento de software. Ela inclui um editor de código com recursos de realce de sintaxe, parênteses correspondentes e identificação automática, sendo capaz de compilar e carregar programas para a placa com um único clique.

Tendo uma biblioteca chamada *Wiring*, ele possui a capacidade de programar em *C/C++*. Isto permite criar com facilidade muitas operações de entrada e saída, tendo que definir apenas duas funções no pedido para fazer um programa funcional:

- `setup()` Inserida no início, na qual pode ser usada para inicializar configuração, e

- `loop()` Chamada para repetir um bloco de comandos ou esperar até que seja desligada.

O Arduino IDE e as duas funções básicas supramencionadas são mostradas na Figura 2.3.

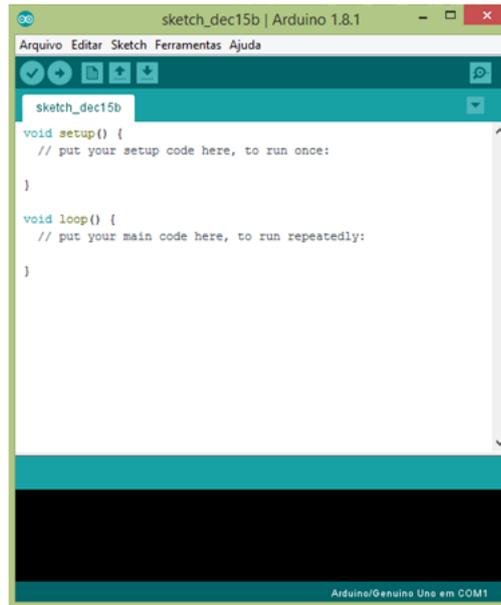


Figura 2.3: Tela do Arduino IDE.

## 2.2 Componentes eletroeletrônicos

Na construção do protótipo foram utilizados alguns componentes eletroeletrônicos, como LEDs, resistores, LCD (tela de cristal líquido) e construído um teclado numérico com botões de contato. O funcionamento e as ligações elétricas necessárias de cada componente, serão descritos nos subcapítulos a seguir.

### 2.2.1 LEDs

Os *LEDs* são diodos emissores de luz, componentes do tipo bipolar, ou seja, que apresentam dois terminais, sendo que um é chamado de anodo e o outro de catodo. Dependendo da forma como ele é polarizado, ele permite ou não a passagem de corrente, e conseqüente emissão de luz. Na Figura 2.4 é possível observar o símbolo elétrico do LED e a identificação de seus terminais.

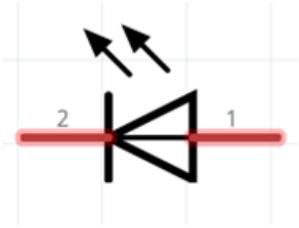


Figura 2.4: Símbolo do LED, com 1 representando o anodo e 2 o catodo.

Atualmente os LEDs disponíveis no mercado são os redondos de 3mm e de 5mm, quadrado de 8x8 mm e o tipo “piranha”. Foram utilizados LEDs do tipo redondo de 5 mm, a Figura 2.5 mostra um deles, o de cor vermelha.



Figura 2.5: LED vermelho.

Em termos práticos, para o LED ser ativado deve-se ter uma corrente circulando em seus terminais no sentido anodo  $\rightarrow$  catodo, o que implica numa diferença de potencial positiva entre os terminais anodo e catodo.

### 2.2.2 Resistor

A propriedade física de um componente ou dispositivo de se opor à passagem de corrente elétrica é chamada de resistência e é representada pelo símbolo  $R$ .

A unidade de resistência é chamada de ohm e é representada pelo símbolo  $\Omega$ ;  $1\Omega = 1V/A$ .

Um componente que possui uma resistência  $R$  é chamado de *resistor*. Um resistor é representado pelo símbolo de dois terminais que aparece na Figura 2.6.



Figura 2.6: Símbolo do resistor.

Existem quatro tipos básicos de resistores: resistores de carbono, resistores de filme de carbono, resistores de filme de metal e resistores de fio. A Figura 2.7 mostra um resistor de filme plástico (tipo que foi utilizado no protótipo), com suas faixas coloridas, que representam um código de cores.



Figura 2.7: Resistor de filme plástico.

O Código de Cores é um sistema criado para identificar a resistência e a tolerância dos resistores, ele é mostrado na Figura 2.8. Os resistores comerciais podem ter valores entre 1 ohm e 10 megaohm.

Color	Cor	1ª Faixa	2ª Faixa	3ª Faixa	Multiplicador	Tolerância
Black	Preto	0	0	0	x 1 $\Omega$	
Brown	Marrom	1	1	1	x 10 $\Omega$	+/- 1%
Red	Vermelho	2	2	2	x 100 $\Omega$	+/- 2%
Orange	Laranja	3	3	3	x 1K $\Omega$	
Yellow	Amarelo	4	4	4	x 10K $\Omega$	
Green	Verde	5	5	5	x 100K $\Omega$	+/- 5%
Blue	Azul	6	6	6	x 1M $\Omega$	+/- 25%
Violet	Violeta	7	7	7	x 10M $\Omega$	+/- .1%
Gray	Cinza	8	8	8		+/- .05%
White	Branco	9	9	9		
Gold	Dourado				x .1 $\Omega$	+/- 5%
Silver	Prateado				x .01 $\Omega$	+/- 10%

Figura 2.8: Código de Cores. Fonte: AudioGroup (2017).

### 2.2.3 LCD

Tela de cristal líquido, acrônimo de *LCD* (em inglês *liquid crystal display*), é um painel fino usado para exibir informações por via eletrônica, como texto, imagens e vídeos. Um LCD consiste de um líquido polarizador da luz, eletricamente controlado, que se encontra comprimido dentro de celas entre duas lâminas transparentes polarizadoras. Os eixos polarizadores das duas lâminas estão alinhados perpendicularmente entre si. Cada cela é provida de contatos elétricos que permitem que um campo elétrico possa ser aplicado ao líquido no interior.

Os módulos LCD são especificados principalmente por sua capacidade gráfica de comunicação, ou seja, o número de caracteres por linha e o número de linhas. Alguns

valores típicos para essas especificações são:

- Número de caracteres: 8, 12, 16, 20, 24 e 40.
- Número de linhas: 1, 2 e 4.

O *display* utilizado apresenta um módulo I2C integrado para simplificar suas ligações com o Arduino. A Figura 2.9 mostra um display LCD de 16 colunas x 2 linhas.



Figura 2.9: *Display* LED 16x2.

O Módulo LCD  $I^2C$  ou  $I2C$  é ideal para ser utilizado em projetos envolvendo LCDs, podendo estar presente em projetos com Arduino ou outros microcontroladores que tenham suporte ao protocolo  $I^2C$  onde há diversos dispositivos (não necessariamente LCDs) que comunicarão entre si com apenas duas linhas de dados.

O protocolo  $I^2C$  foi desenvolvido para conectar diversos dispositivos (periféricos) utilizando apenas as duas linhas de dados SDA (*Serial Data*) e SCL (*Serial Clock*). A ideia principal é definir um endereço hexadecimal para cada dispositivo e no momento de comunicação somente o dispositivo solicitado responderá.

### 2.2.3.1 Teclado de *Push-Button*

Foi manufaturado um teclado matricial de 11 linhas x 1 coluna, identificados pelos numerais de “0” a “9” e “=”, utilizando teclas tácteis de 2 pinos de contatos do tipo normalmente aberto (NA), chamados de *push-buttons*. Esse interruptor fica normalmente aberto (desligado), mas se você pressionar e segurar o botão, o interruptor fecha (liga). Ao soltar o botão, o interruptor abre novamente. A Figura 2.10 mostra o componente.



Figura 2.10: *Push-button* de 2 terminais.

A Figura 2.11 mostra a ligação elétrica de cada tecla, com “Arduino” tratando-se da porta do Arduino que será responsável pela leitura do sinal elétrico da tecla conectada a ela, “GND” é o Terra, “5V” é a tensão de alimentação do circuito e “R” corresponde ao resistor de resistência  $10k\Omega$ .

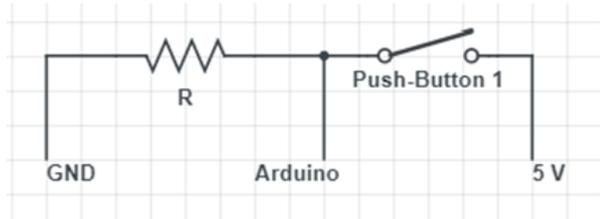


Figura 2.11: Circuito de um das 12 teclas utilizadas no teclado.

## 2.3 Protótipo

O protótipo trata-se de um equipamento composto por um painel de LEDs e uma IHM composta de um LCD 16x2 e um teclado numérico de “0” a “9” e uma tecla de “=”, microcontrolados pela plataforma Arduino. O Arduino será responsável por realizar a leitura do teclado e controlar o acendimento do painel de LEDs e do display. A Figura 2.12 exhibe os componentes do protótipo.

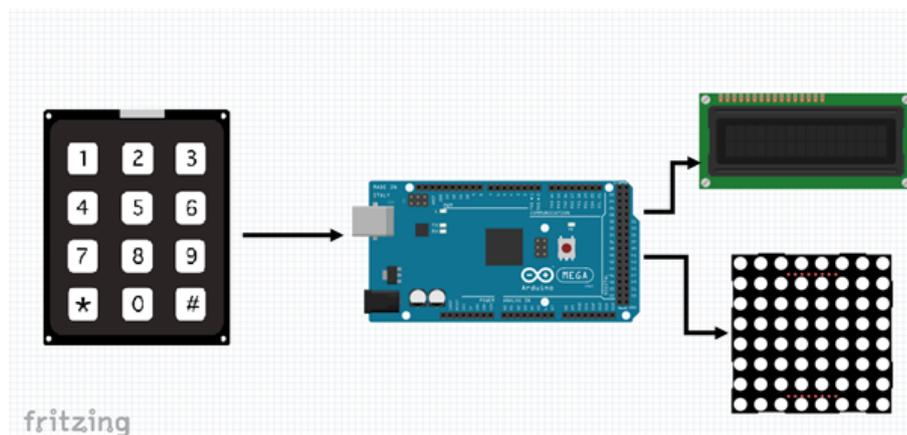


Figura 2.12: Componentes do protótipo.

O microcontrolador escolhido foi o Arduino Mega, já que, para controle individual de cada LED, sem utilizar nenhuma técnica de multiplexagem, seriam necessários pelo menos 121 saídas. Com o Mega e a técnica utilizada, para controle de todos os LEDs serão necessários 22 saídas, fora as portas que seriam demandas para o controle da IHM,

justificando assim a sua escolha, já que é um modelo que apresenta uma quantidade superior em relação aos demais modelos disponíveis.

O painel manufacturado é uma matriz formada por 11 linhas e 11 colunas, totalizando 121 LEDs. Sendo que a linha e a coluna central, de número 6, representam os eixos cartesianos e seus LEDs são de cor vermelha, diferente dos demais que são de cor verde.

Nele, os LEDs de uma mesma coluna apresentam seus anodos interligados entre si, que dali se interliga à um resistor limitador de corrente de  $220\Omega$  e por fim à uma das portas digitais do microcontrolador. Já os LEDs de uma mesma linha, apresentam seus catodos interligados entre si e à uma das portas digitais do microcontrolador. Nas Figuras 2.13 e 2.14 é possível visualizar as conexões elétricas entre os LEDs e o Painel de LEDs já montado.



Figura 2.13: Conexões da matriz de LEDs.

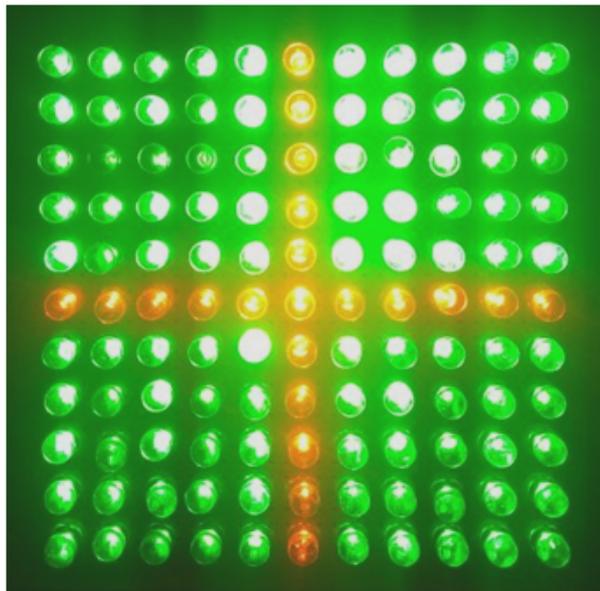


Figura 2.14: Visão superior do painel manufaturado com todos os LEDs acesos.

A ativação de cada LED, que representa um par ordenado, ocorrerá da seguinte forma: de acordo com a função inserida, o programa desenvolvido encontrará os LEDs (coordenada) que deverão ser acesos para formar o traçado da função. Para ativar um LED específico o microcontrolador deverá mandar dois sinais simultâneos de forma a se estabelecer uma diferença de potencial entre os terminais do componente e assim permitir a circulação de corrente e seu acendimento.

Por exemplo, para acender o LED da quinta coluna e da terceira linha, o programa deverá mandar um sinal alto (5V) para a porta do microcontrolador a qual a quinta coluna está conectada e um sinal baixo (0V) para a porta do microcontrolador a qual está conectada a terceira linha.

A IHM desenvolvida é parte da estrutura que permite que o usuário interaja com o painel, ou seja, é nela que ele inserirá a função que deseja visualizar, com auxílio de um teclado numérico que está conectado ao microcontrolador, poderá ser inserido os coeficientes da equação; e é por onde também, poderá se visualizar, em um visor LCD de 16 linhas x 2 colunas, a própria função e o valor do seu zero.

A Figura 2.15 exhibe as ligações elétricas do circuito completo do teclado. Nele podemos ver os 12 botões, os conectores que serão utilizados para conectá-la ao Arduino e à alimentação.

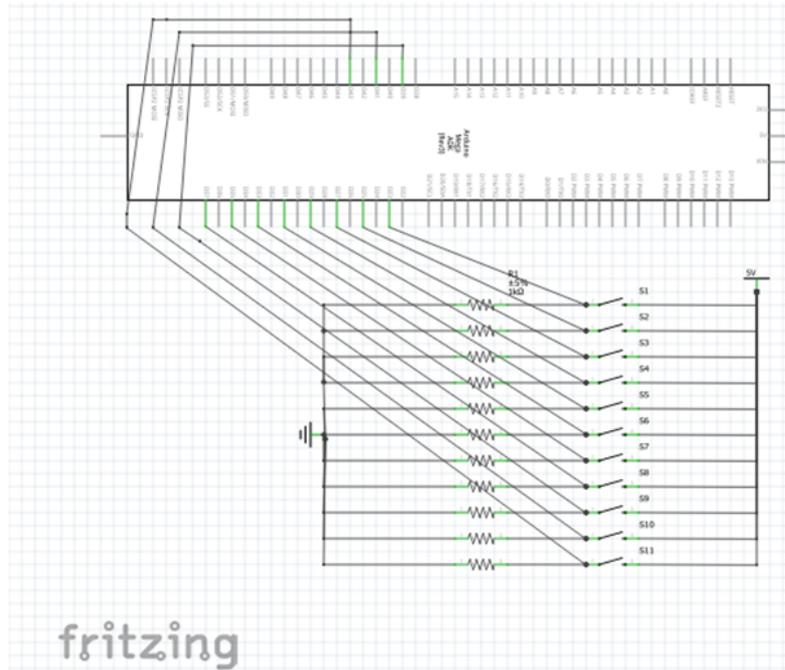


Figura 2.15: Circuito do teclado e ligação com o Arduino.

O circuito funcionará da seguinte forma: como trata-se de uma tecla NA, a leitura que o Arduino fará do circuito será 0V, caso pressionado, o circuito será fechado, e ele passará a ler 5V, comparando essa diferença leitura, o controlador saberá quando cada botão foi pressionado, e assim poderar atuar de forma correta.

A Figura 2.16 indica as conexões necessárias do LCD 16x2 (com módulo *I2C* integrado) que será utilizado no protótipo.

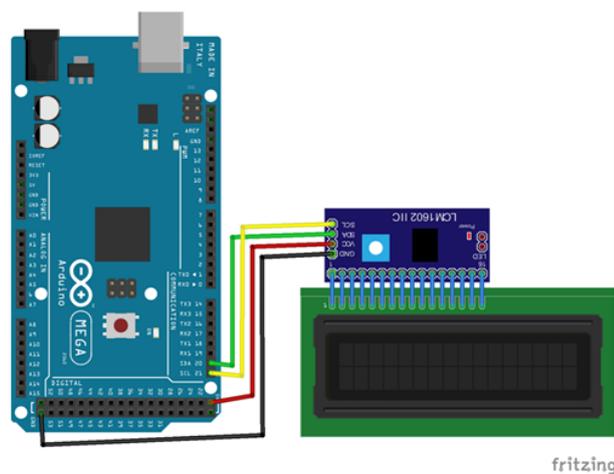


Figura 2.16: Conexões elétricos necessárias entre o LCD e o Arduino Mega.

Por fim, foi construída uma caixa de placas de compensando de madeira com as dimensões: 15 cm de comprimento x 15 cm de largura x 11 cm de altura. Em sua região

superior foi fixado o painel de LEDs, o LCD e o teclado numérico; e manufaturado, em material plástico, uma placa para recobri-la, deixando assim, exposto somente a parte de interação com o usuário.

Na região frontal foi colocado um botão, de dois estados (ON /OFF), para abrir/fechar o circuito da alimentação, e conseqüentemente ligar/desligar o protótipo. As conexões elétricas, o microcontrolador e sua bateria foram acondicionados em seu interior para uma visualização mais atrativa do protótipo. A Figura 2.17 mostra o protótipo construído.



(a) Visão superior.



(b) Visão traseira.

Figura 2.17: Protótipo construído.

## 2.4 Considerações finais do capítulo

Para construção do protótipo foram utilizados componentes eletrônicos de fácil manipulação e fáceis de serem encontrados em lojas de eletrônica. O dimensionamento dos componentes foram realizados de acordo com a alimentação que será utilizada para eles (5V), podendo variar caso seja utilizada uma fonte de alimentação diferente para placas periféricas (placa de LEDs e teclado). Foi utilizado para alimentar o Arduino uma bateria de 9V, é recomendado utilizar uma fonte de tensão de 7V a 12V.

No capítulo 3 será abordado alguns exemplos de estudo de funções, os quais poderão ser utilizado o protótipo como apoio ao ensino em sala de aula.

# Capítulo 3

## Utilizando o protótipo

Este capítulo trará os resultados obtidos utilizando o protótipo construído para aplicações com a função afim e, também, será apresentada uma proposta de atividade que poderá ser utilizada em sala de aula pelos alunos utilizando o equipamento desenvolvido.

### 3.1 Protótipo

O protótipo desenvolvido é para plotagem exclusiva de funções afins, do tipo  $f(x) = ax + b$ , com  $a, b$  e  $x \in \mathbb{Z}$ . Devido a quantidade limitada de LEDs por quadrante, as demais funções ficariam com suas curvas desconfiguradas, além disso, o protótipo é limitado a  $-5 \leq x \leq 5$  e a  $-5 \leq f(x) \leq 5$ , com isso cada LED representa uma coordenada  $(x, y)$  com  $x, y \in \mathbb{Z}$ .

Ao ser iniciado, o protótipo pedirá, via LCD, que seja inserida os coeficientes  $a$  e  $b$  da função  $f(x) = ax + b$  e em seguida pedirá ainda que seja informado se são positivos ou negativos.

A escala do programa, que será utilizada para o cálculo da imagem da função, será de 1 : 1, partindo disso é calculado a imagem da função  $f$  para  $-5 \leq x \leq 5$ .

Além de calcular os pares ordenados das funções inseridas, o programa irá determinar, também, a raiz da função, por meio da lógica desenvolvida, que basicamente trata-se de encontrar o valor de  $x$  para  $y = 0$ .

Após ser informado os coeficientes da função e o programa realizar o cálculo dos pares ordenados e de seu zero, o microcontrolador irá ativar os LEDs correspondentes aos pares ordenados que compõe o traçado da função pedida. Além disso, irá fazer com que

seja exibido no LCD a função fornecida pelo usuário e também o valor do zero calculado. Tais informações irão ser mostradas até que qualquer uma das teclas seja acionada e o programa reinicie pedindo novamente os coeficientes da nova função requerida pelo usuário.

Vamos analisar como o programa irá trabalhar caso fosse inserida a função  $f$ , tal que  $f(x) = -x$ . Temos que para  $x = -5 \rightarrow f(x) = 5$ , então, para acender o LED correspondente ao par ordenado  $(-5, 5)$ , o programa deverá mandar um sinal alto (5V) para a porta do microcontrolador a qual a primeira coluna está conectada e um sinal baixo (0V) para a porta do microcontrolador a qual está conectada a primeira linha. O programa procederá da mesma forma para todos os pares ordenados, para que assim, seja obtido o traçado da função. A Figura 3.1 exhibe como ficará o traçado da função  $f$  supramencionada.

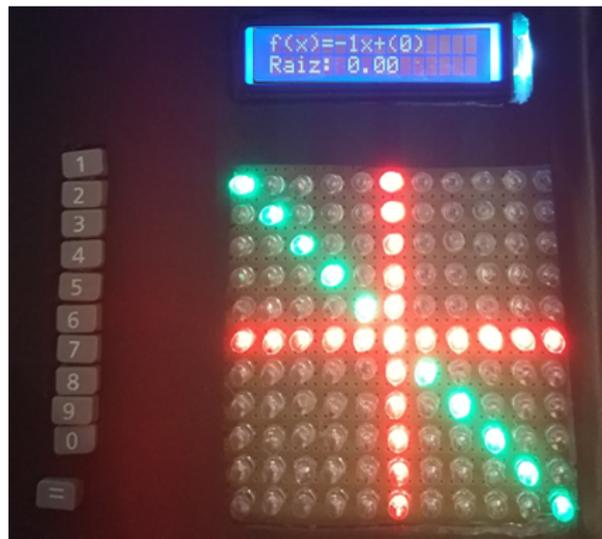


Figura 3.1: Traçado da função  $f$ , tal que  $f(x) = -x$ .

Vamos analisar também a função  $f$ , tal que  $f(x) = x - 2$ . Temos que para  $x = -5 \rightarrow f(x) = -7$ , como o painel de LEDs não comporta o valor da imagem encontrado, o programa não acenderá nenhum LED verde na primeira coluna, isso ocorrerá sempre que  $x \rightarrow -5 > f(x) > 5$ . Ou seja, só haverá acendimento de coordenadas, LEDs, de acordo com as limitações físicas do painel. A Figura 3.2 exhibe como ficará o traçado dessa função.



Figura 3.2: Traçado da função  $f$ , tal que  $f(x) = x - 2$ .

Vamos analisar também a função constante  $f$ , tal que  $f(x) = 4$ . Temos que para todo  $x \rightarrow f(x) = 4$ , então o painel acenderá todos os LEDs da segunda linha e os LEDs correspondentes ao eixo  $XOY$ . A Figura 3.3 exibe como ficará o traçado dessa função.

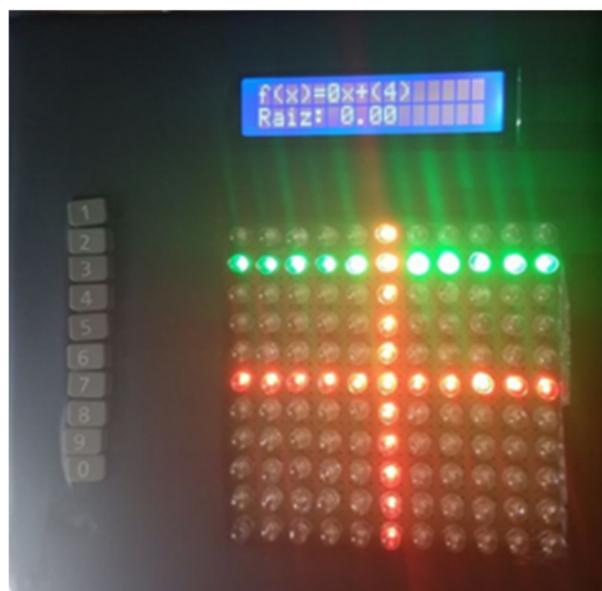


Figura 3.3: Traçado da função  $f$ , tal que  $f(x) = 4$ .

## 3.2 Proposta de atividades para sala de aula

Para utilização do Protótipo, como apoio às aulas, com intuito de estudo de funções e visualização do seu respectivo gráfico, é proposto as atividades a seguir para serem resolvidas e plotadas para sua verificação:

1. Foi realizado o lançamento de um míssil a partir de um submarino, cuja trajetória é vertical (sentido fundo do mar  $\rightarrow$  céu) e definida pela reta  $y = 2x - 4$ , com  $x$  tratando-se do tempo em segundos e  $y$  da altura (em km) em relação ao fundo mar. Determine: a) coordenada do lançamento.

*O lançamento ocorrerá com  $x = 0$ , o que nos fornecerá  $y = 2 \cdot 0 - 4 = -4$ . Portanto, a coordenada do lançamento é  $(0, -4)$ .*

b) instante em que o míssil cruza a linha do oceano.

*O míssil irá cruzar a linha do oceano quando  $y = 0$ , como isso substituindo na equação, obtemos  $0 = 2x - 4 \rightarrow x = 2$ . Portanto, o míssil cruzará a linha do oceano 2 segundos após seu lançamento.*

c) determine a altura que ele estará, em relação a superfície, 3 segundos após o lançamento.

*Temos que  $x = 3$ , daí obtemos que  $y = 2 \cdot 3 - 4 = 2$ . Portanto, o míssil estará a 2 km da superfície.*

d) determine a altura que ele estará, em relação ao ponto de lançamento, após 4 segundos.

*Temos que  $x = 4$ ; daí, obtemos que  $y = 2 \cdot 4 - 4 = 4$ . Como o míssil foi lançado a uma profundidade de 4 km, e como, após 4 segundos, ele se encontra a 4 km da superfície, temos então que ele estará a 8 km do ponto em que foi lançado.*

2. Um entregador de uma pizzaria cobra por entrega realizada. O valor de cada entrega é proporcional a distância da entrega até a pizzaria. Para realizar uma entrega, numa residência que se localizava a 3 km da pizzaria ele cobrou 6 reais. Determine:

a) a função matemática que representa o valor cobrado por ele (em reais) em função da distância (em km) do local à pizzaria.

*Como o valor cobrado é proporcional a distância, então teremos uma função do tipo  $f(x) = ax$ , com  $x$  representando a distância percorrida.*

*Como ele recebeu 6 reais por um deslocamento de 3 km, teremos:  $6 = a \cdot 3 \rightarrow a = 2$ . Daí, obtemos a função  $f(x) = 2x$ .*

b) Numa entrega ele faturou 14 reais, determine a distância do local até a pizzaria.

*Sabendo que ele faturou 14 reais, e que a função matemática é a encontrada em a), temos que  $14 = 2 \cdot x \rightarrow x = 7$ . Portanto, a distância percorrida foi de 7 km.*

3. Num parque de diversão, o valor da entrada é 2 reais e o custo para utilização de cada brinquedo é de 1 real. Determine:

a) a função matemática que representa o valor gasto (em reais) em função do número de brinquedos utilizados.

*Como o valor a ser pagado é composto por uma parcela fixa de 2 reais mais a quantidade de brinquedos vezes o seu custo (1 real), a função que representa o gasto é dado por:  $f(x) = x + 2$ , com  $x$  representando o número de brinquedos utilizados.*

b) o valor gasto por uma criança que utilizou 7 brinquedos.

*Como foram utilizados 7 brinquedos, temos que  $x = 7$ ; daí, obtemos que o gasto será de  $f(7) = 7 + 2 = 9$  reais.*

c) João levou 10 reais, determine o número máximo de brinquedos que ele poderá utilizar.

*Como  $f(x) = 10$ , temos que  $x + 2 = 10 \rightarrow x = 8$ . Portanto, o número máximo de brinquedos que ele poderá utilizar será 8.*

# Capítulo 4

## *Maxima*

O *Maxima* é um Sistema de Computação Algébrica (CAS), de *software* livre, podendo ser baixado, copiado e redistribuído livremente. Ele permite efetuar cálculos numéricos e simbólicos, plotar gráficos em duas e três dimensões e elaborar implementações computacionais eficientes e precisas, mostrando-se uma ferramenta importante para estudo das funções, para encontrar zero(s) e também para plotagem de gráficos.

A utilização do *software* matemático é uma alternativa à construção de um dispositivo para uma visão tecnológica do estudo da teoria matemática e também uma iniciação para o desenvolvimento de equipamentos mais robustos, utilizando os métodos numéricos aqui estudados ou outros de acordo com o foco do trabalho.

No *software Maxima* foi desenvolvido um código do método de Newton-Raphson e outro código para o método da Bissecção para encontrar zero(s) de funções.

Primeiro vamos analisar o código do método de Newton-Raphson para a função  $g$ , tal que  $g(x) = x^3 - 5$ . Inicialmente foi criado o método “newton”, ao qual é fornecida a função “ $f$ ”, aproximação inicial “ $x$ ” e a precisão desejada “*erro*”. O “newton” foi definido como um bloco, no qual obteremos a derivada da função “ $df$ ”, o valor assumido pela função e por sua derivada a cada iteração “ $f(x)$ ” e “ $y$ ”, respectivamente; e será calculado o valor de cada aproximação: “ $x - \frac{f(x)}{y}$ ”. Isso ocorrerá até que o módulo do valor da função para aproximação calculada seja menor que o “erro” informado.

A Figura 4.1 mostra (em azul) o código desenvolvido e o valor obtido (em preto) de cada aproximação.

```

(%i1) newton(f,x,erro):=block([y,df,dfx],df:diff(f('x),'x),
do (
y:ev(df),
x:x-f(x)/y,
print(float(x)),
if abs(f(x))<erro then return (float(x)))
);
func(x):=x^3-5$
newton(func,1,10^-3);

(%o1) newton(f,x,erro):=block([y,df,dfx],df:diff(f('x),'x),do
(y:ev(df),x:x- $\frac{f(x)}{y}$ ,print(float(x)),if |f(x)|<erro then return(float(x))))
2.3333333333333334
1.861678004535147
1.722001880058607
1.710059736600294
(%o3) 1.710059736600294

```

Figura 4.1: Método de Newton-Raphson para a função  $g$ , tal que  $g(x) = x^3 - 5$ .

Agora vamos utilizar o código desenvolvido do método de Newton-Raphson para a função  $g$ , tal que  $g(x) = \frac{\log(x)}{x} + 2x$ . A Figura 4.2 mostra (em azul) o código desenvolvido e o valor obtido (em preto) de cada aproximação.

```

(%i4) newton(f,x,erro):=block([y,df,dfx],df:diff(f('x),'x),
do (
y:ev(df),
x:x-f(x)/y,
print(float(x)),
if abs(f(x))<erro then return (float(x)))
);
func(x):=log(x)/x +2*x$
newton(func,1,10^-3);

(%o4) newton(f,x,erro):=block([y,df,dfx],df:diff(f('x),'x),do
(y:ev(df),x:x- $\frac{f(x)}{y}$ ,print(float(x)),if |f(x)|<erro then return(float(x))))
0.3333333333333333
0.45920616949543
0.53362176823222
0.54784237637325
0.54821683694175
(%o6) 0.54821683694175

```

Figura 4.2: Método de Newton-Raphson para a função  $g$ , tal que  $g(x) = \frac{\log(x)}{x} + 2x$ .

Agora vamos analisar o código do método da Bissecção para a função  $f$ , tal que  $f(x) = x^3 - 5$ . Inicialmente foi criada um método “bissec”, ao qual é fornecida a função  $f$ , o intervalo inicial “a” e “b”, número máximo de iterações “n” e a precisão desejada “erro”. O “bissec” foi definido como um bloco, no qual obteremos  $x_m = \frac{a+b}{2}$ , fazendo  $x_0 = a$  e  $x_1 = b$ . Foi criado, também, no código, um laço do tipo *while*, que avaliará se o número de iterações realizadas é menor do que o número máximo fornecido “n” e se  $|x_1 - x_0| > \text{erro}$ .

Enquanto as duas condições forem verdadeiras o loop ficará contando o número de iterações realizadas a partir do contador “int”, que a cada iteração somará 1 ao valor anterior e também irá avaliar se  $f(x_m).f(x_0) < 0$ , caso positivo fará com que  $x_1 = x_m$  e calculará “xm”, caso negativo fará  $x_0 = x_m$  e calculará “xm”. A Figura 4.3 mostra (em azul) o código desenvolvido e o valor obtido (em preto) para “xm” a cada iteração.

```
(%i4) bissec(f,a,b,n,erro):=block([int:0,xm:(a+b)/2,x0:a,x1:b],
while (int<n and abs(x1-x0)>erro) do
(int:int+1,
if(ev(f,x=xm)*ev(f,x=x0)<0) then (x1:xm, xm:(x0+x1)/2)
else
(x0:xm, xm:(x0+x1)/2),
print(float(xm)) ),

if int=n then
print("Não obteve-se resultado com o erro desejado, a aproximação é", xm)
else xm)$

bissec(x^3-5,1,2,30,10^-3)$
1.75
1.625
1.6875
1.71875
1.703125
1.7109375
1.70703125
1.708984375
1.7099609375
1.71044921875
```

Figura 4.3: Método da Bissecção para a função  $f$ , tal que  $f(x) = x^3 - 5$ .

Utilizaremos o código desenvolvido do método da Bissecção para a função  $g$ , tal que  $g(x) = \frac{\log(x)}{x} + 2x$ . A Figura 4.4 mostra (em azul) o código desenvolvido e o valor obtido (em preto) para  $f(p_n)$  a cada iteração.

```
(%i42) bissec(f,a,b,n,erro):=block([int:0,xm:(a+b)/2,x0:a, x1:b],
while(int<n and abs(x1-x0)>erro) do
(int:int+1,
if(ev(f,x=xm)*ev(f,x=x0)<0) then (x1:xm, xm: (x0+x1)/2)
else
(x0:xm, xm:(x0+x1)/2),
print(float(xm)) ),

if int=n then
print("Não obteve-se o resultado com o erro desejado, a aproximação é", xm)
else xm)$

bissec((log(x)/(x)+(2*x)),0.1,1,30,10^-3)$
0.325
0.4375
0.49375
0.521875
0.5359375
0.54296875
0.546484375
0.5482421875
0.54736328125
0.547802734375
```

Figura 4.4: Método da Bissecção para a função  $g$ , tal que  $g(x) = \frac{\log(x)}{x} + 2x$ .

As Figura 4.5 e 4.6 exibem os gráficos das funções estudadas anteriormente por meio dos métodos numéricos mostrados, obtido no programa Máxima.

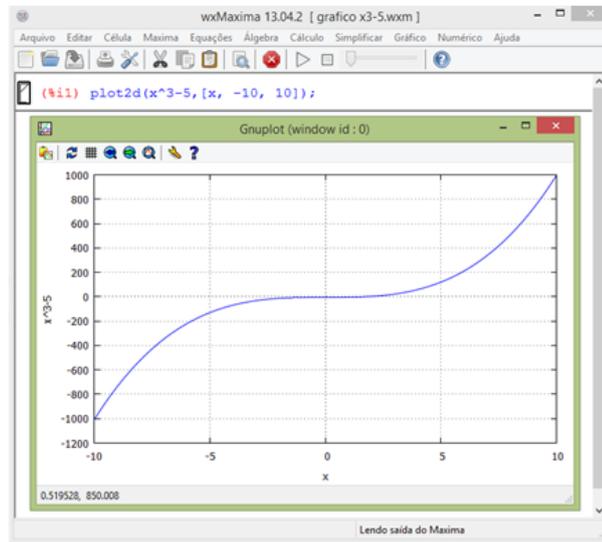


Figura 4.5: Exibe comando de plotagem de gráfico 2D e janela com o gráfico da função  $f$ , tal que  $f(x) = x^3 - 5$ .

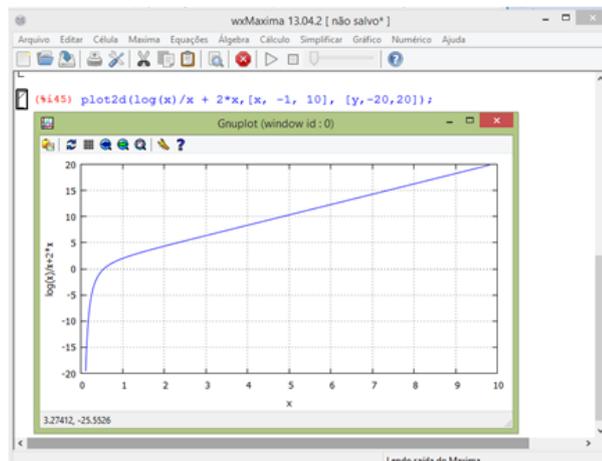


Figura 4.6: Exibe comando de plotagem de gráfico 2D e janela com o gráfico da função  $g$ , tal que  $g(x) = \frac{\log(x)}{x} + 2x$ .

## 4.1 Considerações finais do capítulo

O *software Maxima* apesar de pouco conhecido é uma boa alternativa à outros softwares de domínio, os quais necessitam de licença para o seu uso. Nele pudemos inserir linhas de comando para programação dos métodos estudados e obter as aproximações desejadas, bem como o gráfico das funções estudadas, podendo verificar assim que os

zeros encontrados nos métodos numéricos são de fato os pontos que cortam o eixo das abcissas, como se era esperado.

O *software* serviu de ferramenta para verificação de alguns dos métodos que podem ser utilizados para estudos de funções de maneira computacional, e que, aliado à robótica poderá servir de base para construção de outras ferramentas para o ensino de funções mais complexas, bem como outros assuntos da matemática e de outras disciplinas.

O uso de tais métodos no *software* é um “pontapé inicial” para sua utilização em outras plataformas e um incentivo à utilização de tecnologia no ensino.

# Considerações finais

A plataforma Arduino por ter uma grande comunidade mundial de utilizadores e ser amparada por fóruns, blogs, diversos sites com tutoriais e bibliotecas compartilhadas com códigos prontos para uso, facilita bastante o primeiro contato com o microcontrolador para toda a comunidade das diversas áreas de ensino.

A construção e programação do protótipo pode ser desenvolvida somente pelos educadores e utilizada como ferramenta de ensino, ou até mesmo envolver os alunos em todas etapas, propiciando a eles contato com programação, eletrônica, cálculo numérico e as funções matemáticas em si. Incentivando-os também à participação em eventos tecnológicos, e ao envolvimento com projetos e pesquisas.

O desenvolvimento de um trabalho prático, como a manufatura de um protótipo, apresenta dificuldades intrínsecas, como o aperfeiçoamento da estrutura até obtenção de uma versão funcional e apresentável, sendo necessário substituição de componentes, de soldas, de fiação, teste da programação inúmeras vezes, até obtenção de um código que funcione de maneira satisfatória para a arquitetura utilizada, etc.

Como LEDs são componentes sensíveis à temperatura, no processo de soldagem, foi necessária a substituição de alguns deles e, também, que fosse refeita algumas soldagens devido à mau contato ou rompimento. No desenvolvimento do teclado, foram desenvolvidos 3 modelos diferentes, reaproveitando teclas de calculadoras, mudando a disposição (horizontal e vertical) delas, até se obter teclas que se adaptassem melhor ao tamanho da estrutura e também aos *push-buttons*.

O protótipo desenvolvido é um incentivo para pesquisas multidisciplinares entre a matemática, eletrônica e programação, o protótipo pode ser ampliado não só em dimensões físicas (quantidades de LEDs), mas também quanto à sua aplicação, utilizando os métodos numéricos apresentados para plotagem de funções não lineares, como as polinomiais, logarítmicas e exponenciais e também em outras assuntos da matemática. O

*software Maxima* possibilitou a exibição do funcionamento de tais métodos, servindo como base para sua aplicação em outras plataformas microcontroladas ou não.

Aliando o ensino da matemática à uma ferramenta tecnológica, além de facilitar a visualização de conceitos matemáticos abstratos, desperta o interesse do aluno, já que passará a lidar com um elemento novo no usual padrão de ensino da matemática, que tem uma abordagem muita mais teórica do que prática.

Para trabalhos futuros é possível replicar o protótipo com outros microcontroladores, ou em dimensões maiores, ou até mesmo utilizar microncontroladores, eletrônica e a robótica para desenvolvimento de outras ferramentas de ensino.

# Referências Bibliográficas

Comissão Organizadora. *Campus Party*. Disponível em: <<http://brasil.campus-party.org/>>.

Comissão Organizadora. *FEBRACE*. Disponível em: <<http://febrace.org.br/>>.

Comissão Organizadora. *Olimpíada Brasileira de Robótica*. Disponível em: <<http://www.obr.org.br/>>.

IFRN. *Projeto de robótica conquista 3<sup>o</sup> lugar geral na Febrace 2015*. Disponível em: <<http://portal.ifrn.edu.br/campus/reitoria/noticias/com-projeto-de-robotica-alunos-do-campus-pau-dos-ferros-garantem-3o-lugar-geral-na-febrace-2015>>. Acesso em: abril de 2018.

Brasil. Ministério da Educação. *Estudantes brasileiros ganham certificado de excelência no Japão*. Disponível em: <<http://portal.mec.gov.br/ultimas-noticias/209-564834057/32851-estudantes-brasileiros-ganham-certificado-de-excelencia-no-japao>>. Acesso em: abril de 2018.

Porto, A. *Brasileiro campeão mundial de robótica*. Disponível em: <<https://www.metrojornal.com.br/foco/2017/07/27/brasileiro-e-campeao-mundial-de-robotica.html>>. Acesso em: maio de 2018.

Arduino Group. *Getting Started with Arduino and Genuino products*. Disponível em: <<https://www.arduino.cc/en/Guide/HomePage>>. Acesso em: setembro de 2017.

AudioGroup. *Código de Cores de Resistores*. Disponível em: <<http://encurtador.com.br/cuEGR>>. Acesso em: outubro de 2017.

Golbert, C. S. *Novos rumos na aprendizagem de matemática*. Porto Alegre: Mediação, 2002.

Bettega, M. H. S. *Educação continuada na era digital*. 2 ed. São Paulo: Cortez, 2010.

Rodrigues, W. S. *Atividades com robótica educacional para as aulas de matemática do 6<sup>o</sup> ao 9<sup>o</sup> Ano do Ensino Fundamental: utilização da metodologia LEGO® Zoom Education*. Disponível em: <<https://repositorio.unesp.br/bitstream/handle/11449/127594/000844410.pdf>>. Acesso em: em maio de 2018.

Rodarte, A. P. M. *A Robótica como auxílio à aprendizagem da matemática: percepções de uma professora do ensino fundamental público*. Disponível em: <<https://encurtador.com.br/aCPUV>>. Acesso em: abril de 2018.

Almeida, D. S. *Uso da plataforma arduino para o ensino e aprendizado de matemática*. Disponível em: <<http://webartigos.com/artigos/uso-da-plataforma-arduino-para-o-ensino-e-aprendizado-de-matematica/150524>>. Acesso em: de outubro de 2017.

Guidorizzi, H. L. *Um Curso de Cálculo*. Rio de Janeiro: LTC, 2013.

Stewart, J. *Cálculo*. Rio de Janeiro: Thomson, 2013.

Lima, E. L. *Números e Funções Reais*. Coleção Profmat. Porto Alegre: SBM, 2013.

Brasil. Ministério da Educação. Secretaria de Educação Média e Tecnológica. *Parâmetros Curriculares Nacionais: Ensino Médio*. Brasília, 2000.

Brasil. Ministério da Educação. Secretaria de Educação Média e Tecnológica. *Parâmetros Curriculares Nacionais: Matemática*. Brasília, 1997.

Hohenwarter, M. *GeoGebra Graphing Calculator*, version 6.0. Calculadora Gráfica. Disponível em: <<https://www.geogebra.org/download>>. Acesso em: janeiro de 2018.

MIT. *Maxima*, version 5.41.0. CAS Software. Disponível em: <<http://maxima.sourceforge.net/download.html>>. Acesso em: abril de 2018.

Interaction Design Lab Potsdam. *Fritzing*, version 0.9.3b. CAD Software. Disponível em: <<http://fritzing.org/download/>>. Acesso em: dezembro de 2017.

# Apêndice: Código do programa do protótipo

```
include < Wire.h > include < LiquidCrystal_I2C.h > //TECLADO
intbot1 = 23; intbot2 = 25; intbot3 = 27; intbot4 = 29; intbot5 = 31; intbot6 = 33; intbot7 = 35;
intbot8 = 37; intbot9 = 39; intbot0 = 41; intbotX = 43;

int b1 = 0; int b2 = 0; int b3 = 0; int b4 = 0; int b5 = 0; int b6 = 0; int b7 = 0;
int b8 = 0; int b9 = 0; int b0 = 0; int bX = 0;

// PORTAS DAS COLUNAS DO PAINEL
int coluna1 = 2; int coluna2 = 3; int coluna3 = 4; int coluna4 = 5; int coluna5 = 6;
int coluna6 = 7; int coluna7 = 8; int coluna8 = 9; int coluna9 = 10; int coluna10 = 11;
int coluna11 = 12;

//PORTAS DAS LINHAS DO PAINEL
int linha1 = 22; int linha2 = 24; int linha3 = 26; int linha4 = 28; int linha5 = 30;
int linha6 = 32; int linha7 = 34; int linha8 = 36; int linha9 = 38; int linha10 = 40;
int linha11 = 42;

LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE); //PORTAS DO LCD
int a = 0; int b = 0; int c = 0; int d = 0;

void setup() pinMode(bot1, INPUT); pinMode(bot2, INPUT); pinMode(bot3, INPUT);
pinMode(bot4, INPUT); pinMode(bot5, INPUT); pinMode(bot6, INPUT); pinMode(bot7, INPUT);
pinMode(bot8, INPUT); pinMode(bot9, INPUT); pinMode(bot0, INPUT); pinMode(botX, INPUT);

pinMode(coluna1, OUTPUT); pinMode(coluna2, OUTPUT); pinMode(coluna3, OUTPUT);
pinMode(coluna4, OUTPUT); pinMode(coluna5, OUTPUT); pinMode(coluna6, OUTPUT);
pinMode(coluna7, OUTPUT); pinMode(coluna8, OUTPUT); pinMode(coluna9, OUTPUT);
pinMode(coluna10, OUTPUT); pinMode(coluna11, OUTPUT);

pinMode(linha1, OUTPUT); pinMode(linha2, OUTPUT); pinMode(linha3, OUTPUT);
pinMode(linha4, OUTPUT); pinMode(linha5, OUTPUT); pinMode(linha6, OUT-
```

```

PUT); pinMode(linha7, OUTPUT); pinMode(linha8, OUTPUT); pinMode(linha9, OUT-
PUT); pinMode(linha10, OUTPUT); pinMode(linha11, OUTPUT); Serial.begin(9600);
lcd.begin(16, 2); lcd.setBacklight(HIGH);

void loop()
  leituraTeclado();

  lcd.clear(); lcd.setCursor(0, 0); lcd.print("Painel de Leds:"); lcd.setCursor(0, 1);
lcd.print("Coeficiente a?");

  while (bX == 0) coluna(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0); linha(0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0); //desliga todos coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1); //acende eixo x coluna(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); linha(1, 1, 1, 1, 1, 0, 1, 1, 1, 1,
1);

  leituraTeclado();

  if (b1 == HIGH) c = 1; else if (b2 == HIGH) c = 2; else if (b3 == HIGH) c
= 3; else if (b4 == HIGH) c = 4; else if (b5 == HIGH) c = 5; else if (b6 == HIGH)
c = 6; else if (b7 == HIGH) c = 7; else if (b8 == HIGH) c = 8; else if (b9 == HIGH)
c = 9; else if (b0 == HIGH) c = 0; leituraTeclado();

  b1 = 0; b2 = 0; b3 = 0; b4 = 0; b5 = 0; b6 = 0; b7 = 0; b8 = 0; b9 = 0; b0 =
0; bX = 0; delay(200);

  lcd.clear(); lcd.setCursor(0, 0); lcd.print("Painel de Leds:"); lcd.setCursor(0, 1);
lcd.print("1.a+ e 2.a-"); while (bX == 0)

  leituraTeclado(); if (b1 == HIGH) a = c; else if (b2 == HIGH) a = ((-1) *
c); leituraTeclado(); b1 = 0; b2 = 0; b3 = 0; b4 = 0; b5 = 0; b6 = 0; b7 = 0; b8 = 0;
b9 = 0; b0 = 0; bX = 0; delay(200); lcd.clear(); lcd.setCursor(0, 0); lcd.print("Painel de
Leds:"); lcd.setCursor(0, 1); lcd.print("Coeficiente b?"); while (bX == 0)

  leituraTeclado();

  if (b1 == HIGH) d = 1; else if (b2 == HIGH) d = 2; else if (b3 == HIGH)
d = 3; else if (b4 == HIGH) d = 4; else if (b5 == HIGH) d = 5; else if (b6 ==
HIGH) d = 6; else if (b7 == HIGH) d = 7; else if (b8 == HIGH) d = 8; else if (b9
== HIGH) d = 9; else if (b0 == HIGH) d = 0; leituraTeclado();

  b1 = 0; b2 = 0; b3 = 0; b4 = 0; b5 = 0; b6 = 0; b7 = 0; b8 = 0; b9 = 0; b0 =
0; bX = 0; delay(200);

  lcd.clear(); lcd.setCursor(0, 0); lcd.print("Painel de Leds:"); lcd.setCursor(0, 1);

```

```

lcd.print("1.b+ e 2.b-"); while (bX == 0)
    leituraTeclado(); if (b1 == HIGH) b = d; else if (b2 == HIGH) b = ((-1) *
d); leituraTeclado();
    b1 = 0; b2 = 0; b3 = 0; b4 = 0; b5 = 0; b6 = 0; b7 = 0; b8 = 0; b9 = 0; b0 =
0; bX = 0; delay(200);
    lcd.clear(); lcd.setCursor(0, 0); lcd.print("f(x)="); lcd.print(a); lcd.print("x+");
lcd.print(b); lcd.print(")");
    //CALCULO DA RAIZ float raiz = -(b / a);
    lcd.setCursor(0, 1); lcd.print("Raiz: "); lcd.print(raiz); //CALCULO PARES
ORDENADOS
    int y0 = b;
    int yn1 = ((a * (-1)) + b); int yn2 = ((a * (-2)) + b); int yn3 = ((a * (-3)) + b);
int yn4 = ((a * (-4)) + b); int yn5 = ((a * (-5)) + b);
    int y1 = ((a * 1) + b); int y2 = ((a * 2) + b); int y3 = ((a * 3) + b); int y4 =
((a * 4) + b); int y5 = ((a * 5) + b);
    Serial.println(yn1); Serial.println(yn5);
    while (b1 == 0 b2 == 0 b3 == 0 b4 == 0 b5 == 0 b6 == 0 b7 == 0 b8
== 0 b9 == 0 b0 == 0 bX == 0) if (y5 == (-5)) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0); else if (y5 == (-4)) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1); else if (y5 == (-3)) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1); else if (y5 == (-2)) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1); else if (y5 == (-1)) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1); else if (y5 == 0) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1); else if (y5 == 1) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1); else if (y5 == 2) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1); else if (y5 == 3) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1); else if (y5 == 4) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1); else if (y5 == 5) coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); else coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0);
linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
    coluna(1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); delay(2);
    if (y4 == (-5)) coluna(0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1);

```







1, 1, 1, 1, 1, 1, 0, 1, 1, 1); else if (yn3 == (-1)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0);  
linha(1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1); else if (yn3 == 0) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,  
0); linha(1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1); else if (yn3 == 1) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1,  
0, 0); linha(1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1); else if (yn3 == 2) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 0, 0); linha(1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1); else if (yn3 == 3) coluna(0, 0, 0, 0, 0, 0, 0, 0,  
0, 1, 0, 0); linha(1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1); else if (yn3 == 4) coluna(0, 0, 0, 0, 0, 0,  
0, 0, 1, 0, 0); linha(1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1); else if (yn3 == 5) coluna(0, 0, 0, 0, 0,  
0, 0, 0, 1, 0, 0); linha(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); else coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,  
0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1);  
coluna(0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); delay(2);  
if (yn4 == (-5)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 0); else if (yn4 == (-4)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0); linha(1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 0, 1); else if (yn4 == (-3)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0); linha(1, 1, 1, 1,  
1, 1, 1, 1, 0, 1, 1); else if (yn4 == (-2)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0); linha(1,  
1, 1, 1, 1, 1, 1, 0, 1, 1, 1); else if (yn4 == (-1)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0);  
linha(1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1); else if (yn4 == 0) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,  
0); linha(1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1); else if (yn4 == 1) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1, 0); linha(1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1); else if (yn4 == 2) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 1, 0); linha(1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1); else if (yn4 == 3) coluna(0, 0, 0, 0, 0, 0, 0, 0,  
0, 0, 1, 0); linha(1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1); else if (yn4 == 4) coluna(0, 0, 0, 0, 0, 0,  
0, 0, 0, 1, 0); linha(1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1); else if (yn4 == 5) coluna(0, 0, 0, 0, 0,  
0, 0, 0, 0, 1, 0); linha(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); else coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1,  
0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0); linha(1, 1, 1,  
1, 1, 1, 1, 1, 1, 1, 1); delay(2);  
if (yn5 == (-5)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,  
1, 1, 0); else if (yn5 == (-4)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1); linha(1, 1, 1, 1, 1, 1, 1,  
1, 1, 1, 0, 1); else if (yn5 == (-3)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1); linha(1, 1, 1, 1,  
1, 1, 1, 1, 0, 1, 1); else if (yn5 == (-2)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1); linha(1,  
1, 1, 1, 1, 1, 1, 0, 1, 1, 1); else if (yn5 == (-1)) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1);  
linha(1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1); else if (yn5 == 0) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
1); linha(1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1); else if (yn5 == 1) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,  
0, 1); linha(1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1); else if (yn5 == 2) coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,

```

0, 0, 1); linha(1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1); else if (yn5 == 3) coluna(0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1); linha(1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1); else if (yn5 == 4) coluna(0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1); linha(1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); else if (yn5 == 5) coluna(0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1); linha(0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); else coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1); linha(1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1); delay(2);

    coluna(0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0); linha(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0); delay(2);
//desliga todos coluna(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0); linha(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1);
//acende eixo x coluna(1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1); linha(1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1);
delay(2); leituraTeclado();

    void coluna (int a, int b, int c, int d, int e, int f, int g, int h, int i, int j, int
k) digitalWrite(coluna1, a); digitalWrite(coluna2, b); digitalWrite(coluna3, c); digital-
Write(coluna4, d); digitalWrite(coluna5, e); digitalWrite(coluna6, f); digitalWrite(coluna7,
g); digitalWrite(coluna8, h); digitalWrite(coluna9, i); digitalWrite(coluna10, j); digital-
Write(coluna11, k);

    void linha (int l, int m, int n, int o, int p, int q, int r, int s, int t, int u,
int v) digitalWrite(linha1, l); digitalWrite(linha2, m); digitalWrite(linha3, n); digital-
Write(linha4, o); digitalWrite(linha5, p); digitalWrite(linha6, q); digitalWrite(linha7, r);
digitalWrite(linha8, s); digitalWrite(linha9, t); digitalWrite(linha10, u); digitalWrite(linha11,
v);

    //LEITURA DO TECLADO void leituraTeclado() b1 = digitalRead(bot1); b2
= digitalRead(bot2); b3 = digitalRead(bot3); b4 = digitalRead(bot4); b5 = digital-
Read(bot5); b6 = digitalRead(bot6); b7 = digitalRead(bot7); b8 = digitalRead(bot8);
b9 = digitalRead(bot9); b0 = digitalRead(bot0); bX = digitalRead(botX);

```