



AQUISIÇÃO DE DADOS COM ARDUINO E SMARTPHONE:
EXPERIMENTO DO PÊNDULO SIMPLES.

João Bosco Araújo Fernandes

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação da Universidade Federal Rural de Pernambuco no Curso de Mestrado Profissional de Ensino de Física (MNPEF), como parte dos requisitos necessários à obtenção do título de Mestre em Ensino de Física.

Orientador:
Prof. Dr. Jairo Ricardo Rocha de Oliveira

Recife – PE
2020

AQUISIÇÃO DE DADOS COM ARDUINO E SMARTPHONE:
EXPERIMENTO DO PÊNDULO SIMPLES.

João Bosco Araújo Fernandes

Orientador:
Prof. Dr. Jairo Ricardo Rocha de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação da Universidade Federal Rural de Pernambuco no Curso de Mestrado Profissional de Ensino de Física (MNPEF), como parte dos requisitos necessários à obtenção do título de Mestre em Ensino de Física.

Aprovada por:

Dr. Jairo Ricardo Rocha de Oliveira (DF-UFRPE)
Presidente

Dr. Michael Lee Sundheimer (DF-UFRPE)
Membro Titular Interno

Dr. Eivaldo Montarroyos Rodrigues Lima (UAEADTec/UFRPE)
Membro Titular Externo

Recife – PE
2020

Dados Internacionais de Catalogação na Publicação
Universidade Federal Rural de Pernambuco
Sistema Integrado de Bibliotecas
Gerada automaticamente, mediante os dados fornecidos pelo(a) autor(a)

F363a Fernandes, João Bosco Araújo
Aquisição de dados com Arduino e smartphone: Experimento do pêndulo simples / João Bosco Araújo
Fernandes. - 2020.
263 f. : il.

Orientador: Jairo Ricardo Rocha de Oliveira.
Inclui referências e apêndice(s).

Dissertação (Mestrado) - Universidade Federal Rural de Pernambuco, Programa de Mestrado
Profissional em Ensino de Física (PROFIS), Recife, 2020.

1. Ensino de Física. 2. Arduino. 3. Smartphone Android. I. Oliveira, Jairo Ricardo Rocha de, orient. II.
Título

CDD 530

DEDICATÓRIA

A minha amada esposa Andréia que sempre me encorajou a continuar.

A minha querida mãe que desde criança me orientou a nunca parar de estudar.

Ao meu pai que mesmo depois de sua morte está muito presente em minhas lembranças.

A meus irmãos que sempre tiveram comigo uma atitude de muito carinho e apoio.

AGRADECIMENTOS

Ao meu orientador Professor Doutor Jairo Rocha pela atenção, disponibilidade e dedicação.

Ao Professor Doutor Adauto José Ferreira de Souza pelas conversas e orientações que muito me ajudaram.

A todos os professores do Programa de Pós Graduação em Ensino de Física da UFRPE que contribuíram para meu aprimoramento profissional.

Aos meus colegas de mestrado que nas trocas de experiências pessoais e profissionais me ajudaram de forma muito expressiva.

À CAPES pelo apoio financeiro por meio da bolsa concedida.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001.

RESUMO

AQUISIÇÃO DE DADOS COM ARDUINO E SMARTPHONE: EXPERIMENTO DO PÊNDULO SIMPLES.

João Bosco Araújo Fernandes

Orientador:

Dr. Jairo Ricardo Rocha de Oliveira

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação da Universidade Federal Rural de Pernambuco no Curso de Mestrado Profissional de Ensino de Física (MNPEF), como parte dos requisitos necessários à obtenção do título de Mestre em Ensino de Física.

Esta dissertação tem como objetivo apresentar um guia básico que permita o desenvolvimento de projetos em que um smartphone Android é usado para comunicar-se via bluetooth com uma placa robótica Arduino e assim controlar tanto o acionamento de dispositivos elétricos como também a aquisição de dados em experimentos físicos. Assim, primeiro é desenvolvida uma sequência de estudo do hardware e software mínimos necessários. E por último é apresentado um roteiro de atividade experimental para obtenção da expressão da relação entre o período e o comprimento de um pêndulo simples. A realização dessa atividade com vários grupos de educandos teve como característica marcante o alto grau de interesse e aprendizagem dos educandos participantes.

Palavras-chave: Ensino de Física, Arduino, Smartphone Android.

ABSTRACT

DATA ACQUISITION WITH ARDUINO AND SMARTPHONE: SIMPLE PENDULUM EXPERIMENT

João Bosco Araújo Fernandes

Supervisor:

Dr. Jairo Ricardo Rocha de Oliveira

Master's Dissertation submitted to the Programa de Pós-Graduação da Universidade Federal Rural de Pernambuco in the Curso de Mestrado Profissional de Ensino de Física (MNPEF), as part of the necessary requirements to obtain the title of Master in Physics Teaching.

This dissertation aims to present a basic guide that allows the development of projects in which an Android smartphone is used to communicate via bluetooth with an Arduino robotic board and thus control both the activation of electrical devices as well as the acquisition of data in experiments physicists. Thus, a sequence of study of the minimum necessary hardware and software is first developed. Finally, an experimental activity guide is presented to obtain the expression of the relationship between the period and the length of a simple pendulum. The accomplishment of this activity with several groups of students was characterized by the high degree of interest and learning of the participating students.

Keywords: Teaching Physics, Arduino, Android Smartphone.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 3.1 – Sistema massa-mola | 40 |
| Figura 3.2 – O Pêndulo Simples..... | 43 |
| Figura 4.1 – Sistema interativo..... | 47 |
| Figura 4.2 – Circuito elétrico simples | 48 |
| Figura 4.3 – Circuito elétrico automatizado..... | 49 |
| Figura 4.4 – Dispositivos de entrada e saída em um projeto Arduino..... | 50 |
| Figura 4.5 – Arduino Uno..... | 54 |
| Figura 4.6 – Ícone da IDE do Arduino | 55 |
| Figura 4.7 – IDE do Arduino quando aberto pela primeira vez | 56 |
| Figura 4.8 – Menu Ferramentas do Arduino..... | 56 |
| Figura 4.9 – Item de menu Placas do Arduino | 57 |
| Figura 4.10 – Item de menu Porta do Arduino | 57 |
| Figura 4.11 – Detalhe de um LED verde com 5mm de tamanho | 66 |
| Figura 4.12 – Protoboard..... | 69 |
| Figura 4.13 – Esquema de ligação de um LED externo ao Arduino | 70 |
| Figura 4.14 – Esquema de um relé EMR simples | 72 |
| Figura 4.15 – Módulo relé de 1 canal | 73 |
| Figura 4.16 – Acionamento de uma lâmpada por um módulo relé. | 74 |
| Figura 4.17 – Estados de um comutador elétrico..... | 75 |
| Figura 4.18 – Aspecto e símbolo do botão de pressão de quatro pinos. | 76 |
| Figura 4.19 – Controle do nível lógico com botão de pressão. | 77 |
| Figura 4.20 – Resistor de pull-up e resistor de pull-down. | 77 |
| Figura 4.21 – Botão pull-down para controlar acionamento do <i>LED built-in</i> | 78 |
| Figura 4.22 – Aspecto, elementos e símbolos de um LDR. | 82 |
| Figura 4.23 – Circuito do divisor de tensão. | 83 |
| Figura 4.24 – Divisor de tensão com LDR..... | 85 |
| Figura 4.25 – Controle do acionamento de um LED com um LDR..... | 86 |
| Figura 4.26 – Monitor serial integrado a IDE do Arduino. | 88 |
| Figura 4.27 – Circuito para monitorar o estado de um botão momentâneo..... | 90 |
| Figura 4.28 – Módulo HC-06. | 92 |

| | |
|---|-----|
| Figura 4.29 – Circuito para acionamento da placa HC-06. | 93 |
| Figura 5.1 – Ícone e tela inicial do app Serial Bluetooth Terminal. | 97 |
| Figura 5.2 – Lista de telas e Devices do Serial Bluetooth Terminal. | 98 |
| Figura 5.3 – Configuração dos botões de macros M1, M2 e M3. | 100 |
| Figura 5.4 – Resultado da configuração dos botões de macros M1, M2 e M3. | 100 |
| Figura 6.1 – Repouso e movimento do pêndulo simples. | 101 |
| Figura 6.2 – Esquema de construção do pêndulo simples. | 102 |
| Figura 6.3 – Haste metálica conectada ao fixador de tubos de PVC. | 103 |
| Figura 6.4 – Esquema de incidência da luz do LED sobre um LDR. | 104 |
| Figura 6.5 – Componentes do pêndulo e do dispositivo de medida. | 105 |
| Figura 6.6 – Invólucro opaco do LDR contra efeitos da luz ambiente. | 107 |
| Figura 6.7 – Circuito elétrico de ligação do LED ao Arduino. | 108 |
| Figura 6.8 – Circuito elétrico de ligação do LDR ao Arduino. | 109 |
| Figura 6.9 – Circuito elétrico de ligação do eletroímã ao Arduino. | 109 |
| Figura 6.10 – Circuito elétrico de ligação do Bluetooth ao Arduino. | 110 |
| Figura 6.11 – Circuito de medição do período na protoboard. | 111 |
| Figura 6.12 – Circuito de medição e fontes de tensão. | 111 |
| Figura 6.13 – Dispositivo de medida do período do pêndulo simples. | 112 |
| Figura 6.14 – Fluxograma do sketch de controle. | 114 |
| Figura 7.1 – Interface de medição. | 137 |
| Figura 7.2 – Alunas realizando a medição. | 137 |
| Figura 7.3 – Alunos realizando a medição. | 138 |

LISTA DE GRÁFICOS

Gráfico 7.1 – Gráfico log T x log L traçado pelo 7º grupo 132

LISTA DE QUADROS

| | |
|--|-----|
| Quadro 1.1 – Contribuições dos experimentos ao ensino de ciências. | 16 |
| Quadro 2.1 – Critérios de classificação dos experimentos de física. | 31 |
| Quadro 2.2 – Graus de liberdade do aluno em aulas de laboratório | 34 |
| Quadro 4.1 – Código de cores dos resistores | 68 |
| Quadro 5.1 – Botões de comando do Serial Bluetooth Terminal. | 98 |
| Quadro 6.1 – Setores de um ciclo de movimento da esfera. | 106 |
| Quadro 6.2 – Códigos de ação enviados pelo smartphone. | 117 |
| Quadro 7.1 – Sugestões de alteração da Ficha de Experimentos. | 135 |

LISTA DE TABELAS

| | |
|--|-----|
| Tabela 4.1 – Valores de V_{OUT} para um LDR com $V_{in} = 5 V$ | 85 |
| Tabela 6.1 – Relação entre o raio e o ângulo inicial do pêndulo. | 108 |
| Tabela 7.1 – Aspecto final da Tabela 1 da Ficha de Experimento..... | 130 |
| Tabela 7.2 – Aspecto final da Tabela 2 da Ficha de Experimento..... | 131 |
| Tabela 7.3 – Relação experimental período x comprimento..... | 134 |

SUMÁRIO

| | | |
|-------|--|----|
| 1 | Introdução | 15 |
| 2 | Fundamentação pedagógica | 19 |
| 2.1 | O construtivismo | 19 |
| 2.1.1 | A teoria de Jean Piaget..... | 20 |
| 2.1.2 | A teoria de Vygotsky | 23 |
| 2.1.3 | Implicações do construtivismo para a prática pedagógica | 25 |
| 2.2 | Classificações das atividades experimentais | 31 |
| 2.3 | Proposta de atividade experimental na concepção construtivista | 35 |
| 3 | Fundamentação física | 38 |
| 3.1 | Movimento periódico | 38 |
| 3.2 | Movimento oscilatório | 39 |
| 3.3 | Período do movimento harmônico simples (MHS)..... | 39 |
| 3.4 | Período do pêndulo simples..... | 43 |
| 4 | A plataforma Arduino | 47 |
| 4.1 | Aquisição de dados, monitoramento e controle | 47 |
| 4.2 | O microcontrolador Arduino..... | 50 |
| 4.3 | A IDE Arduino | 55 |
| 4.4 | Configurando a IDE off-line do Arduino..... | 56 |
| 4.5 | A linguagem de programação do Arduino..... | 58 |
| 4.5.1 | Controle de acionamento do LED built-in | 63 |
| 4.5.2 | Controle de acionamento de um LED externo | 66 |
| 4.5.3 | Eletroímã | 71 |
| 4.5.4 | Controle de acionamento de um módulo relé | 73 |
| 4.5.5 | Controle da recepção de sinal do sensor de toque | 74 |
| 4.5.6 | Controle da recepção de sinal do sensor de luz..... | 81 |
| 4.5.7 | Transmissão via porta serial | 87 |
| 4.5.8 | Transmissão via Bluetooth | 91 |
| 5 | Serial Bluetooth Terminal | 96 |

| | | |
|-------|---|-----|
| 5.1 | Usando o app Serial Bluetooth Terminal..... | 96 |
| 5.2 | Botões de Macros..... | 99 |
| 6 | Dispositivo de medida do período | 101 |
| 6.1 | Montagem do pêndulo simples | 101 |
| 6.2 | Medição do período do pêndulo com Arduino e LDR | 103 |
| 6.3 | Controle do início do movimento do pêndulo | 107 |
| 6.4 | Esquemas elétricos do dispositivo de medição..... | 108 |
| 6.5 | Determinação da frequência operação do loop | 112 |
| 6.6 | Sketch para controle do dispositivo de medição..... | 113 |
| 6.6.1 | Variáveis globais do sketch..... | 114 |
| 6.6.2 | Procedimento setup | 116 |
| 6.6.3 | Procedimento loop | 116 |
| 6.6.4 | Procedimento lerSerial | 117 |
| 6.6.5 | Procedimento medirLDR..... | 119 |
| 7 | Aplicação da proposta didática..... | 122 |
| 7.1 | Perfil da escola e dos educandos participantes | 122 |
| 7.2 | Perfil da proposta e avaliação diagnóstica..... | 122 |
| 7.3 | Etapa Pré-Experimental | 124 |
| 7.4 | Etapa Experimental | 128 |
| 7.5 | Etapa Pós-Experimental..... | 131 |
| 7.6 | Registro fotográfico da aplicação da proposta | 137 |
| 8 | Considerações finais | 139 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 144 |
| | APÊNDICE A – Sketch PENDULO_SIMPLES | 153 |
| | APÊNDICE B – Ficha de Experimento do Pêndulo Simples..... | 155 |
| | APÊNDICE C – Produto Educacional..... | 162 |

1 INTRODUÇÃO

A física é uma ciência fundamentalmente experimental, contendo também aspectos filosóficos, teóricos e matemáticos. Portanto, a realização de experiências deveria ser essencial. Porém, o ensino dessa disciplina no ensino médio é em geral centrado em conceitos teóricos e dificilmente atividades experimentais são realizadas. Além disso, um grande número de educandos traz do ensino fundamental limitações de ler, escrever e calcular. Como consequência, boa parcela dos estudantes tem a percepção de que durante as aulas de física são estudadas coisas totalmente abstratas, coisas estas que não lhes atraem e sem nenhuma ligação com os fenômenos físicos naturais vivenciados por eles no dia a dia. Na visão deles, a disciplina é difícil de entender, desmotivante e chata. Isso gera aversão à disciplina e culmina em baixo rendimento (PERUZZO, 2013, p. 9-10).

Na tentativa de resolver este problema, Peruzzo (2013) e diversos outros autores, entre os quais Araújo e Abib (2003), Azevedo et al (2009), Oliveira (2010) e Wesendonk e Prado (2015) e o próprio PCN+ (BRASIL, 2000) defendem a experimentação como uma metodologia indispensável para despertar o interesse dos alunos, facilitar a assimilação e o entendimento dos conteúdos e trazer um aprendizado enriquecedor. E Carvalho et al (1998, p. 20) chegam a afirmar que: “A importância do trabalho prático é inquestionável na Ciência e deveria ocupar lugar central em seu ensino”.

Oliveira (2010, p. 141-146) e Wesendonk e Prado (2015, p. 60) concordam que as aulas experimentais trazem diversas contribuições ao ensino e aprendizagem de ciências estando aquelas que eles consideram mais importantes listadas no Quadro 1.1.

Desta forma, o trabalho experimental aguça a curiosidade, o hábito de sempre indagar, interpretar cálculos e situações do cotidiano. Leva a uma visão de mundo melhor do que aquela adquirida através de informações teóricas, herméticas e fórmulas sem sentido, o que entusiasma os discentes e torna a aprendizagem de Física muito mais motivadora.

Quadro 1.1 – Contribuições dos experimentos ao ensino de ciências.

| Ordem | Contribuição |
|-------|--|
| 1 | Aprender a analisar dados e propor hipóteses para os fenômenos |
| 2 | Aprender conceitos científicos |
| 3 | Aprimorar a capacidade de observação e registro de informações |
| 4 | Aprimorar habilidades manipulativas |
| 5 | Compreender a natureza da ciência |
| 6 | Compreender as relações entre ciência, tecnologia e sociedade |
| 7 | Compreender o papel do cientista em uma investigação |
| 8 | Desenvolver a capacidade de trabalhar em grupo |
| 9 | Desenvolver a iniciativa pessoal e a tomada de decisão |
| 10 | Detectar e corrigir erros conceituais dos alunos |
| 11 | Estimular a criatividade |
| 12 | Motivar e despertar a atenção dos alunos |

Adaptado de: Oliveira (2010, p. 141-146)

Martinazzo et all (2014), Cavalcante, Tavolaro e Molisani (2011) e ainda Souza et all (2011), entre muitos outros, apontam que uma placa de protopinagem tipo microcontrolador Arduino quando usada como metodologia experimental de aquisição de dados permite acesso rápido a uma grande quantidade de dados que podem ser processados, tabulados e convertidos em gráficos e facilmente usados em uma grande diversidade de análises estatísticas, representando a possibilidade real de desenvolver nos educandos uma série de importantes habilidades e competências, o que seria muito difícil sem o uso dessa metodologia.

Além disso, segundo UNESCO (2014), desenvolver uma atividade educativa que utilize algum dispositivo de comunicação móvel, como smartphones e tablets, torna essa atividade muito mais atrativa, pois as pessoas, principalmente os jovens, usam cada vez mais telefones celulares básicos, smartphones, tablets, leitores eletrônicos e leitores de smartcard para acessar redes sociais, jogos, vídeos, desenhos animados, imagens, figuras, gráficos, simulações, áudios, histórias em quadrinhos, apresentações multimídias entre outros. Assim, a aprendizagem móvel (*Mobile Learning*) que é o fruto do fortalecimento dos vínculos entre inovações técnicas e pedagógicas, permitirá uma integração sistemática na educação dentro e fora das escolas e

assumirá um papel cada vez mais importante na educação geral na próxima década. Mas isso apresenta uma grande inconsistência com a atitude de muita desconfiança e resistência que atualmente boa parcela de educadores tem ao uso pelo educando em sala de aula de dispositivos móveis, em particular do smartphone, pois como Lopes e Pimenta (2017) lembram, esses dispositivos são vistos como meros instrumentos de distração para os estudantes. Isso ocorre principalmente, como afirma Silva (2016), com uma parcela de professores que não têm o conhecimento que os permita sentir conforto e segurança para aplicar essa metodologia e assim se tornam resistentes ao seu uso na sala de aula e deixam de utilizá-las por falta de formação. Porém, ainda segundo UNESCO (2014), como qualquer tecnologia, com o passar do tempo, esses aparelhos estão ficando mais baratos e mais acessíveis, levando os educadores a se sentirem cada vez mais confortáveis em usá-los em uma grande variedade de propósitos. Assim, a tendência é que os professores passem a utilizar as tecnologias móveis primeiro para facilitar a aprendizagem em suas próprias vidas e depois, em algum momento, junto aos educandos.

Em resumo, aproveitar o fascínio e as vantagens obtidas na realização de experimentos físicos e também no uso das tecnologias presentes nos aparelhos celulares, computadores e robótica, deveria ser uma estratégia comum usada no processo de ensino-aprendizagem de Física.

Nesse sentido, a presente dissertação tem como primeiro objetivo desenvolver um guia de estudo para permitir a professores criar projetos com seus educandos em que um smartphone possa comunicar-se via Bluetooth com uma placa robótica Arduino e assim controlar tanto o acionamento de dispositivos elétricos como a aquisição de dados em experimentos físicos. E tem como segundo objetivo desenvolver uma aplicação prática para determinar experimentalmente a expressão harmônica do período para um pêndulo simples.

Com essa intenção, no capítulo dois é desenvolvida a fundamentação pedagógica que embasa a metodologia usada.

Já no capítulo três é apresentada a fundamentação física através da determinação teórica da expressão harmônica para o período de um pêndulo simples.

No quarto capítulo, inicialmente é apresentado o microcontrolador Arduino que, a partir de um programa computacional, gerencia o acionamento de dispositivos elétricos e a leitura de dados externos. A seguir, ainda no mesmo capítulo, é apresentado um guia de atividades com o Arduino e alguns outros componentes em que a complexidade aumenta gradativamente, tanto em nível de hardware quanto de programação, sempre buscando tornar mais suave o esforço de aprendizagem: Acender o LED *on-board* (na placa) do Arduino; Acionar um LED externo; Controlar um módulo rele; Receber dados de um sensor de luz; E finalmente, fazer a comunicação entre um smartphone Android e o Arduino com um módulo Bluetooth.

No quinto capítulo é apresentado o aplicativo Serial Bluetooth Terminal, um programa que instalado em um smartphone Android permite enviar e receber dados de forma muito simples através de uma conexão Bluetooth.

Em seguida, no sexto capítulo é apresentado em detalhes um dispositivo de medição do período do pêndulo simples por aquisição de dados com Arduino e smartphone.

No capítulo sete é discutida a metodologia usada na aplicação da proposta didática em uma turma de educandos da segunda série do ensino médio.

Por último, no capítulo oito, são apresentadas as considerações finais.

2 FUNDAMENTAÇÃO PEDAGÓGICA

2.1 O CONSTRUTIVISMO

O Construtivismo tem como tese central que o conhecimento é construído e reconstruído continuamente e não pode ser considerado como algo terminado. Ele é o resultado da interação entre o indivíduo com seu meio físico e social, logo não é apenas produto do ambiente nem se origina unicamente da bagagem hereditária individual (BECKER, 1993).

Assim, o construtivismo claramente difere do empirismo, orientado pelo chamado determinismo ambiental em que o homem é produto do ambiente e o conhecimento só pode ser derivado a partir dos dados acumulados pelas experiências ao longo de sua vida. É também diferente do inatismo, em que a herança genética com a qual o indivíduo nasce já traz todas as estruturas do conhecimento, que apenas se atualizam à medida que ele se desenvolve (LEÃO, 1999).

Custódio et al (2013) relatam que o Construtivismo atualmente designa um conjunto teórico com desdobramentos práticos que é resultado das obras de diversos autores, existindo referências a ele como sendo uma teoria psicológica, ou como uma teoria em ciências sociais, ou teoria em ensino de ciências ou ainda como teoria educativa. E na sua vasta literatura é possível encontrar mais de quinze significados diferentes para o termo construtivismo, todos compartilhando a ideia central de construção do conhecimento, mas possuindo algumas singularidades e interpretações diferentes as quais são fontes de controvérsias e críticas.

El-Hani e Biso (2002) o dividem em três correntes: educacional; filosófico; e sociológico. Mas lembram que apesar dessa divisão, não se pode esquecer que o construtivismo educacional ou pedagógico tem aspectos filosóficos e sociológicos, em conjunto com proposições sobre o ensino e a aprendizagem. E

grande parte da produção que tem como fundamento o construtivismo refere-se à educação, mais especificamente, à educação formal.

Vários estudiosos podem ser classificados como teóricos do construtivismo e historicamente as obras de Piaget e de Vygotsky são consideradas as que tiveram maior influência. Além deles, podem ser citados também os trabalhos de Henri Wallon, A. N. Leontiev, A. R. Luria e Emília Ferreiro (LEÃO, 1999).

Ver em detalhes as obras desses pesquisadores está fora do escopo do presente trabalho, por isso a seguir será visto apenas um pequeno resumo das principais ideias contidas nas pesquisas de Piaget e Vygotsky e algumas consequências dessas teorias para a educação escolar.

2.1.1 A teoria de Jean Piaget

Os trabalhos do biólogo, filósofo e epistemólogo suíço Jean Piaget (1896-1980) surgem a partir da segunda década do século XX. Ao longo de sua vida Piaget escreveu mais de 60 livros e uns 1500 artigos com objetivo de explicar como o sujeito passa de um conhecimento anterior menor para um posterior maior, ou seja, como o conhecimento é construído. A psicologia e a epistemologia que estudam esse processo foram classificadas pelo próprio Piaget como psicologia genética e epistemologia genética, uma vez que tinham como objetivo estudar a gênese do conhecimento. Assim, seus estudos não tinham uma preocupação eminentemente pedagógica e sim epistemológica (GOMES e BELLINI, 2009).

Embora tenha escrito alguns trabalhos sobre educação, Piaget não foi e não pretendeu ser pedagogo, e não teve, portanto, como interesse, desenvolver projetos ou métodos pedagógicos, não propondo um método de ensino ou elaborando materiais pedagógicos. Mas, os métodos pedagógicos hoje ditos construtivistas são fortemente influenciados pela teoria desenvolvida por ele,

principalmente os métodos sobre o desenvolvimento infantil (SANCHIS e MAHFOUD, 2010).

Moreira (1999) deixa claro que Piaget produziu uma das mais influentes visões da teoria construtivista do desenvolvimento cognitivo humano. Tal influência provoca confusão, chegando-se a pensar com naturalidade que por definição, a teoria de Piaget é a própria teoria construtivista. Por isso, é importante observar que o enfoque piagetiano não é a teoria construtivista, e sim, a mais conhecida e importante dentre as visões construtivistas.

Moreira (1999) ainda enfatiza que os quatro estágios ou períodos gerais de desenvolvimento cognitivo, que Leão (1999) chama construtivismo sequencial, é parte muito conhecida e importante da teoria piagetiana, mas ele afirma que o "núcleo duro" da teoria está na Assimilação, na Acomodação e na Equilibração, que Leão (1999) chama interacionismo piagetiano, pois nelas se encontra a idéia chave a respeito do mecanismo de desenvolvimento intelectual humano. O extenso trabalho de Piaget aborda uma enorme quantidade de outros tópicos, mas somente estes dois serão vistos mais detalhadamente aqui.

Segundo Gomes e Bellini (2009), a teoria piagetiana diz que em cada um dos quatro estágios de desenvolvimento cognitivo do indivíduo são construídas uma série de estruturas que serão indispensáveis ao próximo estágio e a idade que um indivíduo entra e sai de cada estágio depende do desenvolvimento e da qualidade de sua educação. Esses estágios têm a seguinte ordem cronológica:

- a) Período Sensório-Motor: também chamado pré-verbal. Dura, aproximadamente do nascimento aos dezoito meses de vida. Nele a conduta social do bebê é de isolamento e para ele "o mundo é ele". Mas a partir de reflexos neurológicos básicos, tem início a construção de estruturas para assimilar mentalmente o meio e desenvolver o conhecimento prático. De início, quando um objeto sai do campo de visão da criança, o objeto não mais existe para ela. A construção do

objeto permanente só vai ocorrer quando o bebê tentar achá-lo em sua nova localização espacial. Assim, a construção do espaço prático acompanha a construção do objeto permanente, que leva a construção da sucessão temporal e da causalidade sensório-motora básica.

- b) Período Pré-Operatório: também chamado Simbólico ou Intuitivo. Dura aproximadamente dos 2 anos aos 7 anos. Nele ocorre a etapa intuitiva e de aprendizagem instrumental básica. É quando se desenvolve a capacidade de substituir o objeto pela sua representação simbólica e assim surge a linguagem e as capacidades de desenhar, de imitar, de dramatizar, entre outras.
- c) Período Operatório Concreto: Dura aproximadamente dos 7 anos aos 11/12 anos. Neste estágio, ela ainda depende do mundo concreto para abstrair, mas desenvolve noções de tempo, espaço, velocidade, ordem e casualidade e é capaz de relacionar diferentes aspectos e abstrair dados da realidade. Ela constrói a ideia de número e assim consegue realizar as operações de classificação, ordenamento, operações espaciais e temporais e todas as operações fundamentais da lógica elementar de classes e relações, da matemática elementar, da geometria elementar e até da física elementar.
- d) Período Operatório Lógico Formal: também chamado Abstrato ou Hipotético-dedutivo. Dura dos 12 aos 16 anos em diante. É o ápice do desenvolvimento das estruturas intelectuais características dos adultos. Agora as estruturas cognitivas alcançam seu nível mais elevado de desenvolvimento e tornam-se aptas a aplicar o raciocínio lógico a todas as classes de problemas.

Para Piaget a construção do conhecimento só é possível através de uma interação mediada pela ação do sujeito. Quando o sujeito se depara com algo novo, ocorre uma perturbação, um processo de desequilíbrio em suas estruturas

cognitivas (ou Esquemas) preexistentes e assim ele procura reformular esses Esquemas para atingir um novo estado de equilíbrio e para isso, podem ocorrer dois processos (SANCHIS e MAHFOUD, 2010).

No primeiro deles, o sujeito pode sofrer Assimilação, ou seja, uma integração de novos dados provenientes da experiência com seus Esquemas prévios, que ou permanecem invariáveis ou sofrem pequena modificação, mas sem descontinuidade com o estado anterior, sem serem destruídos e apenas incorporando-se à nova situação.

Porém, quando o indivíduo não consegue Assimilar um novo estímulo ocorre o segundo processo, chamado de Acomodação. Agora para haver uma integração de novos dados é necessário criar novos Esquemas ou modificar profundamente Esquemas existentes para ser possível fazer a integração.

A Equilibração ocorre quando após um desequilíbrio, acontece uma Assimilação ou Acomodação bem sucedida, isto é, a passagem de um estado desequilibrado para um novo estado de equilíbrio. E assim, o crescimento cognitivo pode ser visto como uma sucessão de interações entre os processos de assimilação e acomodação com o objetivo de atingir estados de equilíbrio cada vez mais estáveis e duradouros.

A aprendizagem só é possível se estruturas mais simples ao serem atingidas por um estímulo externo sofrerem um processo de absorção desse estímulo mudando para estruturas mais complexas, ou seja, quando há um processo natural de desenvolvimento de estruturas e não simplesmente um reforço externo.

2.1.2 A teoria de Vygotsky

A teoria de Lev Semenovich Vygotsky (1896-1934) é atualmente reconhecida como uma das mais importantes para o construtivismo, apesar de

sua relativa incompletude, devido à morte precoce de Vygotsky, aos 38 anos de idade. Além disso, um decreto de 1936 proibiu a publicação e venda de suas obras por serem consideradas não marxistas. Esse fato dificultou a divulgação de suas produções e somente depois de vinte anos seus estudos puderam ser reunidos e reeditados, sendo a difícil tarefa de sistematização realizada principalmente por seus colaboradores e seguidores (MIRANDA, 2005).

Vygotsky afirmava que aquilo que parece individual na pessoa é na verdade resultado da construção da sua relação com o outro, no coletivo, que está ligado à cultura. Assim, as características e atitudes individuais estão profundamente impregnadas das trocas com o coletivo e é justamente na cultura dos seus valores, na negociação dos sentidos que se constrói e se internaliza o conhecimento.

Miranda (2005) aponta que as três Zonas de Desenvolvimento são um conceito central da teoria de Vygotsky:

- I. Zona de desenvolvimento real: refere-se ao nível de maturação já alcançado pela criança e que permite a ela solucionar problemas de forma independente.
- II. Zona de desenvolvimento potencial: é a capacidade que a criança tem de solucionar problemas com a ajuda de adultos ou companheiros mais capazes. Para tanto, a participação do outro mais experiente é fundamental.
- III. Zona de desenvolvimento próximo (ZDP): é a distância entre as zonas de desenvolvimento real e potencial. Ou seja, é tudo que a criança ainda não sabe fazer, mas que se espera que ela aprenda a fazer e assim a ZDP é um indicador do desenvolvimento em processo e que está por se consolidar.

A consequência direta das zonas de desenvolvimento para a prática pedagógica é que o professor deve planejar atividades a partir do nível de

desenvolvimento real de seus alunos, tendo como referência o nível de desenvolvimento potencial deles, para possibilitar a eles aprender o que se espera nas atividades propostas.

Miranda (2005) também destaca que Vygotsky considerava a mediação um conceito muito importante, pois para ele a relação do homem com o mundo físico e social é sempre mediada e se dá através do uso de instrumentos e de signos. Os instrumentos são vinculados às possibilidades de transformação da natureza pelo homem. Já os Signos são mediadores de natureza psicológica, ligados a realização de tarefas que precisam de atenção ou memória, que podem representar dados da realidade. As mediações tornam-se mais complexas e articuladas, e as representações do real são combinadas em sistemas simbólicos compartilhados pelo grupo social. É o caso da linguagem, considerada por Vygotsky como sistema simbólico básico.

Finalmente, Miranda (2005) indica a linguagem como o terceiro conceito importante na teoria de Vygotsky. Para ele, a linguagem como sistema de signos levou o ser humano do reino do biológico para o do sócio-histórico. Para satisfazer a necessidade que os homens têm de se comunicar entre si, eles criam e fazem uso dos sistemas de linguagem. Assim, a principal função da linguagem é a de intercâmbio social, cuja evolução levou à utilização de signos, os quais transmitem idéias, pensamentos, sentimentos e podem ser entendidos por outras pessoas.

2.1.3 Implicações do construtivismo para a prática pedagógica

O construtivismo vê a aprendizagem como um processo de construção dos conhecimentos pelo aprendiz a partir da sua interação com seu meio ambiente e social, em um processo dialético de geração e superação de conflitos cognitivos. Nesse processo, os conhecimentos prévios do aluno, suas estruturas cognitivas e a interação dele com o meio são fundamentais. O educando não pode ser visto como um recipiente passivo de conhecimentos,

mas, como um ser ativamente engajado na busca de desenvolver novas habilidades, atitudes e conhecimentos e que traz consigo seu conhecimento anterior como pré-requisito para atingir esses objetivos. E também que a aprendizagem somente tem a possibilidade de ocorrer se os conteúdos programáticos apresentados se situarem na zona de desenvolvimento próximo, por isso, o professor deve constantemente desafiar os alunos com atividades que se referem a habilidades e conhecimentos um pouco além de seu nível atual de domínio, tendo cada nova aprendizagem o efeito de aumentar mais confiança e motivação do aluno devido a seus sucessos anteriores.

Argento (sd) apresenta uma série de características do construtivismo aplicadas ao ensino, entre as quais: o encorajamento da autonomia e iniciativa dos estudantes; o contínuo questionamento que o professor deve ter sobre o atual estágio de compreensão do estudante antes de apresentar novos conceitos; o estímulo aos estudantes a dialogar com o professor e entre si; o encorajamento aos estudantes a resolverem problemas abertos e ajudarem-se uns aos outros; e a permissão aos estudantes de conduzir as aulas, alterar estratégias instrucionais e de conteúdo.

Também no construtivismo, o erro passa a ser visto como uma importante fonte de aprendizagem, um passo na direção do acerto. O erro revela ao aprendiz a inadequação de seus esquemas e mostra a necessidade da construção de outros e/ou a reformulação daqueles previamente existentes. Ele aponta novos caminhos metodológicos na ação didática do professor ao qual cabe incentivar o educando a sempre questionar-se sobre o que errou e tentar novamente para conseguir a devida aprendizagem. Nesse contexto, a forma e a importância da avaliação mudam completamente em relação às práticas convencionais (SALSA, 2017).

O construtivismo considera que aprender é um longo processo no qual o aluno vai reconstruindo seu conhecimento a partir das atividades que realiza, por isso, Camargo e Faria (2011) entende a avaliação como um processo contínuo

que tem como única e grande finalidade diagnosticar o que o aluno já apreendeu e o que ainda está necessitando aprender. Dessa maneira, a avaliação tem a função básica de ao detectar os acertos e erros à medida que a aprendizagem do educando evolui, fornecer um feedback informativo tanto para ele próprio quanto para o professor. A tomada de consciência do seu estágio de aprendizagem estimula o envolvimento e responsabilização pessoal do aluno nas tarefas para melhorar seu desempenho. Os erros também se tornam os meios pelos quais o professor identifica as principais dificuldades e facilidades dos alunos, permitindo realizar as intervenções que julgar necessário para aprimorar o processo de ensino e aprendizagem e por isso ela é chamada de avaliação formativa. A própria Lei de Diretrizes e Bases (LDB) 9.394/1996 (BRASIL, 2005) acentua a importância da avaliação formativa, ao definir no artigo 24 que a verificação do rendimento escolar deverá ser uma “avaliação contínua e cumulativa do desempenho do aluno, com prevalência dos aspectos qualitativos sobre os quantitativos e dos resultados ao longo do período sobre os de eventuais provas finais”.

Ao docente também cabe definir padrões de avaliação que reforcem o exercício do pensar, a habilidade de solucionar problemas e minimizar o memorizar simplesmente. Assim, ela pode ser realizada diariamente ou periodicamente, através de observações do professor, entrevistas individuais e coletivas, análise de anotações e tarefas dos educandos, através de perguntas ou participação em aula, pela realização de testes, projetos, autoavaliação dos alunos e outras formas.

Evidentemente antes de iniciar um novo conteúdo, o professor deve fazer uma avaliação, agora chamada diagnóstica, nos mesmos moldes da avaliação formativa para determinar os conhecimentos prévios dos educandos e realizar as ações necessárias para que eles venham a ter condições de aprender tais conteúdos.

No construtivismo, passa a ser importante também entender como os educandos conseguem aprender, como eles constroem e reconstróem seus conhecimentos. Valadares (2001) lembra que a capacidade intelectual das pessoas apresenta múltiplas características. Algumas têm uma grande capacidade verbal e de comunicação. Outras têm um raciocínio lógico acentuado. Existe às que resolvem problemas práticos com grande facilidade. E há pessoas que têm dificuldades com o raciocínio abstrato ou têm alguma deficiência intelectual, mas têm alguns tipos de inteligência mais desenvolvidos que outros. Portanto, todas as pessoas aprendem e se desenvolvem segundo suas características, limitações e potencialidades. A essas diferentes formas de aprender das pessoas chamam-se estilos de aprendizagem. Assim, existem os visualizadores, que recorrem aos aspectos icônicos para aprenderem. Existem também os verbalizadores, que aprendem explicando para si mesmos e para os outros. Há os perguntadores, que recorrem a perguntas aos outros. Também existem os socializadores, que aprendem discutindo com os outros. Finalmente, existem os ouvintes, que gostam de aprender ouvindo os outros. Já Laburú e Carvalho (2011) apresentam vários outros critérios de diferenciação. Assim, existem os educandos holistas, que têm uma postura individualista de aprendizagem e preferem formar uma visão mais global quando da resolução de problemas, e os serialistas, que preferem estudar, progressivamente, uma hipótese por vez. Há também os estudantes com personalidade competitiva e que gostam de exibir sua capacidade intelectual, em oposição aos que são pessimistas sobre suas habilidades, ou que são metodicamente estudiosos, estudando por longos períodos. Outra possível classificação é quanto a preferência pelos modos de instrução da ciência: Os executores, que são indiferentes ao estilo; Os curiosos, pois preferem a aprendizagem a partir de livros, por descoberta e realização de atividades práticas; Os cumpridores de tarefas, pois preferem o ensino didático convencional com experimentos sustentados por instruções; E por último, os sociais, pois mostram maior afinidade na realização de atividades em grupo.

Não há o que discutir da óbvia noção de que o conhecimento é construído individualmente, mas também não há a menor dúvida de que o processo pedagógico para essa construção é altamente complexo, pois os educandos possuem diferentes estilos de aprendizagem e a adoção de uma única estratégia de ensino, qualquer que seja ela, terá como resultado a diminuição da aprendizagem dos alunos uma vez que não respeita seus estilos. Assim, a tarefa pedagógica do educador para ser mais efetiva deve usar várias técnicas de ensino para colocar os educandos em situações que atendam a seus estilos de aprendizagem, o que torna a atividade de ensinar muito heterogênea e leva o professor a parecer, segundo Argento (sd), “tanto como um apresentador do conhecimento como um facilitador de experiências”. Porém, o professor deve planejar com cuidado o uso de cada atividade, a sequência a seguir e o tempo de duração de cada uma delas, sempre buscando o equilíbrio metodológico entre elas, pois a utilização excessiva de qualquer uma pode causar desmotivação nos educandos e prejudicar ou até impedir a aprendizagem.

Como Camargo e Faria (2011) deixam claro, no construtivismo não existe um modelo pedagógico a ser seguido, o que existe é uma teoria do conhecimento e, de desenvolvimento humano que traz implicações para o ensino, entre as quais está o fato do aprendizado ser fruto da troca entre o indivíduo e meio a partir de suas ações. Assim o professor deve promover metodologias que estimulem essa troca.

No presente trabalho serão destacadas quatro metodologias de ensino: as didáticas colaborativas, as atividades com projetos, o autodidatismo e o didatismo.

As didáticas colaborativas são uma das práticas mais utilizadas no construtivismo. Nela, os estudantes trabalham fundamentalmente em grupos ou equipes, debatendo e trocando ideias com seus colegas sobre o que está sendo ensinado em sala de aula e com objetivo de construir algo em conjunto. Essa

prática estimula uma série de habilidades como o diálogo, a empatia, a liderança e a tomada de decisões.

As atividades com projetos são voltadas para a resolução criativa de situações-problema na sala de aula, na escola, na comunidade e na vida social mais ampla. Elas são muito valorizadas no construtivismo por estimularem a autonomia dos educandos e a necessidade de uso de múltiplos conteúdos para elaboração da atividade, o que favorece a interdisciplinaridade.

No autodidatismo, o professor estimula o educando a adquirir conhecimento de forma autônoma, através de tarefas individuais de pesquisa, resolução de exercícios, etc. O seu objetivo é tornar o aprendiz um caçador de conhecimentos na escola, no tempo livre e ao longo de toda a vida.

No didatismo, basicamente o professor apresenta o conteúdo aos alunos. Porém, em vez da aula expositiva em que o aluno se limita apenas a assistir e refletir sobre o que é apresentado, aqui se defende o que Anastasiou e Alves (2007) chamam de aula expositiva dialogada. Nela, a participação ativa dos estudantes é sempre estimulada por meio de um diálogo. Assim, continuamente e durante toda a atividade, o professor faz questionamentos sobre o que está sendo apresentado e estimula que os educandos façam perguntas e ponderações. O conhecimento prévio dos alunos pode ser tomado como ponto de partida e o tema deve ser contextualizado e trabalhado na zona de desenvolvimento próximo deles. Assim, eles são induzidos a confrontar as informações que trazem com as que são apresentadas, criar conflitos cognitivos e gerar aprendizagem. Se bem planejada e realizada, essa atividade pode ser o melhor meio para apresentar informações de forma organizada, relacionar diferentes visões sobre um mesmo tema, expor um conteúdo distante do dia a dia dos alunos, sistematizar conteúdos já trabalhados, trabalhar tópicos obscuros da teoria, rever conceitos não compreendidos e apresentar exemplos e questões-chaves do conteúdo trabalhado. A aula expositiva dialógica permite desenvolver no aluno a habilidade verbal de argumentação, melhorar sua

autoestima e construir sua identidade. E quanto mais dialogada for a aula, melhores condições estarão sendo criadas para cada aluno aprender a conviver com a pluralidade de opiniões, a respeitar o ponto de vista do outro e a fazer negociação.

Para Rugunnett et al (2017) o didatismo é fonte de grande polêmica, pois para algumas vertentes do construtivismo, o aprendiz deve alcançar o conhecimento de maneira independente, sempre chegando a uma conclusão exclusivamente por si próprio e assim, a exposição de conteúdos deve ser evitada. Isso ocorre, segundo Chakur (2014), devido a uma má interpretação dada aos trabalhos de Piaget, pois o que ele realmente condena é o excesso de verbalismo na transmissão de conteúdos escolares. Seguindo essa ideia, existem outros construtivistas que defendem a possibilidade de que haja momentos mais adequados de aulas expositivas, de transmissão direta da informação organizada.

2.2 CLASSIFICAÇÕES DAS ATIVIDADES EXPERIMENTAIS

É fácil encontrar na literatura especializada em ensino de física, diversas formas de classificar as atividades experimentais. Para fazer essa classificação diversos critérios podem ser considerados. Assim, a partir dos trabalhos de Carvalho (2010), Azevedo et al. (2009), Araújo e Abib (2003) e Moreira e Gonçalves (1980) os experimentos foram agrupados segundo o Quadro 2.1.

Quadro 2.1 – Critérios de classificação dos experimentos de física.

| Equipamento | Ênfase matemática | Direcionamento | Grau de Liberdade |
|--------------|-------------------|------------------|-------------------|
| Profissional | Qualitativo | Demonstrativo | Estruturado |
| Alternativo | Quantitativo | Verificacionista | Não estruturado |
| | | Investigativo | |

Fonte: elaborado pelo autor.

Usando como critério o tipo de equipamento, existem os experimentos que usam aparatos de origem profissional (AZEVEDO et al., 2009, p. 4) que são produzidos por empresas conceituadas e especializadas na construção de equipamentos de laboratório para o ensino de ciências. Esses equipamentos têm como vantagens a sua rápida montagem para utilização e a precisão das medidas obtidas. Porém, tendem a fazer uso mais intenso das fichas de laboratório com roteiros prontos e de ter sua aquisição tornada proibitiva para as escolas de menor poder aquisitivo, principalmente as instituições públicas de ensino fundamental e médio.

Em contrapartida, os experimentos alternativos, que Azevedo et al. (2009) chamam de aparatos de montagem simples, são constituídos por montagens em geral simples, com o uso de materiais de baixo custo e fácil aquisição, que podem ser objetos do cotidiano, reciclados e até sucatas. O aparato muitas vezes é construído apenas pelo professor. Mas quando os educandos estão envolvidos no seu desenvolvimento, o engajamento necessário a essa construção estimula a criatividade e o espírito colaborativo e os leva a um forte aprendizado, não só dos conteúdos diretamente ligados ao experimento, mas também de muitos outros conteúdos e habilidades necessários a produção das estruturas desses artefatos.

As atividades experimentais qualitativas privilegiam os aspectos de natureza conceitual enquanto aquelas quantitativas são propostas com objetivo de salientar aspectos formais relacionados com teorias e modelos matemáticos (ARAÚJO e ABIB, 2003).

Ainda segundo Araújo e Abib (2003), as atividades experimentais demonstrativas são aquelas nas quais o os alunos observam o professor realizar o experimento com objetivo de tornar o aprendizado mais interessante, fácil e agradável. Como em geral demandam um pequeno tempo de realização, podem ser usadas no início de uma aula expositiva, procurando despertar o interesse do aluno para o tema que será abordado ou podem ser utilizadas como um

fechamento da aula como forma de reforçar o aprendizado dos conteúdos apresentados.

Como o próprio nome sugere, as atividades de verificação são aquelas empregadas com a finalidade de se verificar ou confirmar alguma lei ou teoria e assim estimular a capacidade dos alunos de analisar os parâmetros dos fenômenos observados, articulando-os com os conceitos científicos e de efetuar generalizações. Por necessitar da abordagem prévia do conteúdo, essa atividade deve ser realizada após a aula expositiva (ARAÚJO e ABIB, 2003).

Os experimentos do tipo investigativo estimulam o estudante a ter uma postura mais ativa no processo de aprendizagem. Normalmente eles são realizados antes da exposição dos conteúdos correlacionados à atividade, de modo que os resultados não são totalmente previsíveis aos estudantes, forçando-os a buscar as respostas que faltam e incentivando-os a refletir, questionar e argumentar sobre os fenômenos e conteúdos científicos presentes no experimento. Por esse motivo, Azevedo et al. também os chamam de experimentos problematizadores. Cabe destacar que essas atividades frequentemente exigem um tempo maior, pois envolvem uma série de etapas a serem desenvolvidas pelos alunos.

Moreira e Gonçalves (1980) descrevem que na atividade estruturada o educando recebe um roteiro composto por instruções que o guiam com objetivo de produzir certos resultados específicos. Já numa atividade completamente não estruturada, o aluno diante de um problema fica inteiramente livre para usar o procedimento que achar melhor, organizar seus dados e chegar as suas próprias conclusões. O grau de liberdade que é dado ao educando é extremamente variável e baseado em Carvalho (2010, p. 50) pode ser resumido no Quadro 2.2.

Assim, no nível 0 estariam as atividades em que o professor apresenta o experimento e os educandos apenas visualizam os fenômenos que obedecem à

lógica da teoria apresentada. Já no nível V estão atividades em que desde a definição do problema até as conclusões da pesquisa são definidas e realizadas pelos educandos.

Quadro 2.2 – Graus de liberdade do aluno em aulas de laboratório

| Grau | 0 | I | II | III | IV | V |
|-------------------|---|---|----|-----|----|---|
| Problema | P | P | P | P | P | A |
| Hipótese | P | P | P | P | A | A |
| Plano de trabalho | P | P | P | A | A | A |
| Obtenção de dados | P | A | A | A | A | A |
| Conclusões | P | P | A | A | A | A |

P = Professor A = aluno

Adaptado de Carvalho (2110, p. 55)

É importante observar que, sob o ponto de vista construtivista, quaisquer dessas categorias de experimentos podem ser utilizadas desde que os educandos tenham um efetivo aprendizado induzido pelo processo de assimilação, equilíbrio e acomodação.

Assim um simples experimento de demonstração, com grau de liberdade 0 pela Tabela 2.2, que atraia a atenção dos educandos e permita a eles, ao visualizar os fenômenos apresentados, perceber sua lógica e entender de fato a teoria correspondente, conduz a uma aprendizagem efetiva. Já a realização de uma atividade investigativa não estruturada, com grau de liberdade V pela tabela 2.2, apesar de sob o ponto de vista construtivista ter um grande potencial de produzir forte aprendizagem, pode não atingir esse objetivo por se tornar muito difícil para alunos com déficit de conhecimentos básicos e sem treino na realização de experimentos.

Por isso, Oliveira (2010) e outros pesquisadores sugerem que inicialmente as aulas experimentais tenham uma abordagem mais simples, do tipo

demonstração e/ou verificação e sejam bem estruturadas (grau 0 ou 1 pela Tabela 2.2). À medida que os educandos se familiarizem com essa estratégia de ensino, gradativamente passem a realizar experimentos cada vez mais abertos e complexos até atingirem um nível que os permita desenvolver atividades de investigação totalmente não estruturadas (grau V da Tabela 2.2).

Por fim, a escolha de determinada atividade experimental depende de vários aspectos, entre os quais, dos objetivos específicos do problema em estudo, das competências que se quer desenvolver e da disponibilidade de tempo e recursos materiais. Assim, para explorar adequadamente todas as potencialidades de determinada categoria de experimento é importante que o professor entenda as diferenças entre elas e saiba quando e como utilizá-las.

2.3 PROPOSTA DE ATIVIDADE EXPERIMENTAL NA CONCEPÇÃO CONSTRUTIVISTA

Num contexto construtivista toda atividade teórica ou experimental deve estar intimamente ligada à dinâmica da mediação planejada pelo professor e com objetivo da elaboração do conhecimento científico na sala de aula e não pela simples realização de um conjunto de procedimentos (ROSA, 2012).

Para Carvalho et al (1998, p. 20), numa perspectiva construtivista, não se deve esperar que o educando descubra conhecimentos inéditos por meio do trabalho experimental. A principal função das experiências é aumentar o conhecimento do aluno sobre os fenômenos naturais e fazer com que ele relacione tais experimentos com sua visão do mundo, sempre a partir das hipóteses e conhecimentos anteriores e com o auxílio do professor quando necessário. Assim, em uma atividade é inicialmente necessária a apresentação de um problema pelo professor. Dessa forma, a busca da solução do problema é a mola propulsora de todas as ações dos alunos. Essa busca produz discussões, gera motivação e desafio e desperta o interesse. Solucionar um

problema intrigante é motivo de euforia, pois promove a autoconfiança necessária para o aluno divulgar suas ações e sugerir explicações. Qualquer experimento deve ter a função de gerar uma situação problemática, indo muito além da simples manipulação de materiais. Por isso, é necessário um tempo especial para que o aluno possa refletir e tomar consciência de suas ações de modo a compreender a importância de cada passo executado e, por fim, propor explicações causais.

Numa concepção construtivista, Rosa (2012) apresenta uma proposta em que na realização de uma atividade experimental ocorrem três etapas distintas: pré-experimental, experimental e pós-experimental. Ela salienta que para serem atingidos todos os objetivos desejados é de fundamental importância destinar um tempo adequado para cada uma dessas etapas e não privilegiar apenas a etapa experimental.

A etapa pré-experimental envolve inicialmente a apresentação da teoria, ou seja, é mostrado aos estudantes o conteúdo e estabelecido o foco de estudo. Pode haver mais de um objetivo, porém todos claramente definidos, conduzindo os estudantes a entender o que se deseja analisar. Esse é um bom momento para apresentar os equipamentos disponíveis para realizar a atividade experimental, pois eles auxiliam o entendimento do objetivo. Também deve ser realizada a formulação das hipóteses, as quais devem sempre anteceder a observação a ser realizada e nunca o contrário. Elas têm papel indispensável e servem como um guia da atividade e mostram que há “algo” a ser testado, verificado. As hipóteses também permitem aos estudantes construir e reconstruir contínua e progressivamente os conhecimentos já existentes em suas estruturas cognitivas. Para finalizar esta etapa é necessário fazer o planejamento das ações. O educando deve, antes de realizar o experimento, refletir sobre o que será feito, registrando cada passo a ser realizado em um roteiro ou guia de ações. Por exemplo, selecionar os conhecimentos necessários à atividade experimental, como uma expressão matemática que relacione variáveis do problema. Esse “modo de fazer” deve ser apresentado de forma a levar o

estudante a pensar e planejar suas ações entendendo o que e por que proceder de determinada forma.

Na etapa experimental são desenvolvidas as atividades listadas no roteiro anteriormente planejado. Geralmente nela são manipulados equipamentos para testar as hipóteses previstas e registrados os dados obtidos relativos ao objetivo a alcançar. Para obter sucesso nessa atividade o educando precisa apresentar uma postura ativa, estar engajado com a atividade e ser capaz de interagir socialmente, pois geralmente as atividades experimentais são realizadas em grupos de trabalho, o que promove o diálogo e a troca de conhecimento entre os colegas e com o professor.

A etapa pós-experimental é aquela em que deve ocorrer o fechamento da atividade experimental, representado pela conclusão. Nela não ocorre apenas a apresentação dos resultados, mas principalmente é o momento em que o educando é estimulado a analisar, interpretar, confrontar e discutir os resultados obtidos, representando um momento de construção do conhecimento. Neste ponto de vista, concluir significa rever o que foi realizado, verificar a ocorrência de possíveis falhas no processo, sintetizar e revisar a atividade. Como envolve uma série de atividades mentalmente complexas, essa etapa necessita de um tempo adequado para serem obtidos resultados satisfatórios.

3 FUNDAMENTAÇÃO FÍSICA

3.1 MOVIMENTO PERIÓDICO

O movimento periódico é aquele em que a posição e a velocidade se repetem em intervalos de tempo regulares e sucessivos. O intervalo de tempo decorrido entre duas repetições sucessivas do movimento é chamado de período T do movimento.

Sendo Δt o tempo gasto para ser realizado um determinado número de repetições n , o período será dado por:

$$T = \frac{\Delta t}{n} \quad (3.1)$$

Ainda, segundo Marques (2014, p. 277), supondo que em instante de tempo t , um corpo tem posição $\vec{R}(t)$ e velocidade $\vec{v}(t)$, decorrido um período de tempo T , valem as relações:

$$\vec{R}(t) = \vec{R}(t + T) \quad (3.2)$$

$$\vec{v}(t) = \vec{v}(t + T) \quad (3.3)$$

A rotação da Terra em torno do seu eixo, o movimento da Terra em torno do Sol, o movimento de um pêndulo e do sistema massa-mola são exemplos de movimentos periódicos.

Outra propriedade importante do movimento periódico é a sua frequência f , que representa o número de vezes que o movimento se repete por unidade de tempo e é definida pelo inverso do seu período:

$$f = \frac{1}{T} \quad (3.4)$$

3.2 MOVIMENTO OSCILATÓRIO

O movimento oscilatório, também chamado vibratório, é um caso particular de movimento periódico. Nele, o sentido da velocidade é invertido a cada intervalo de tempo. Essa inversão se dá quando a velocidade do corpo se anula (MARQUES 2014, p. 278).

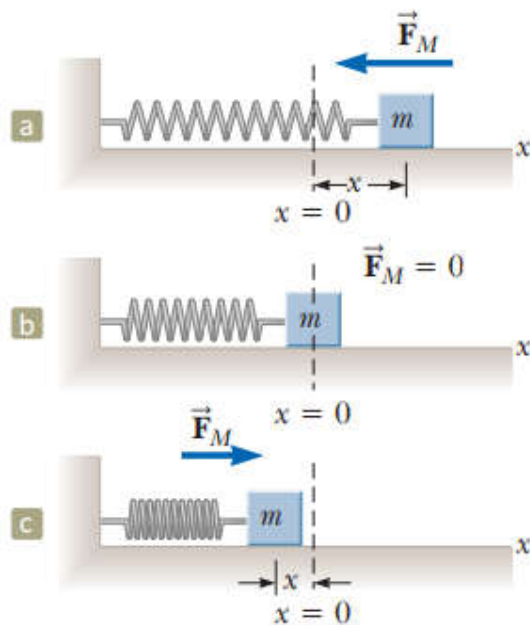
Existem inúmeros exemplos de objetos que se movem repetidamente de um lado para outro, entre os quais, o movimento de um pêndulo e o movimento de um corpo preso a uma mola. Como a Terra não inverte o sentido do giro em torno do Sol, seu movimento é periódico, mas não é oscilatório.

3.3 PERÍODO DO MOVIMENTO HARMÔNICO SIMPLES (MHS)

O MHS é um tipo especial de movimento oscilatório em que um corpo oscila em torno de sua posição de equilíbrio por sofrer a ação de uma força proporcional ao deslocamento sofrido. Como esta força é sempre orientada para a posição de equilíbrio, ela é conhecida como força restauradora. Tal comportamento linear acontece em geral apenas para pequenos valores de deslocamento (SERWAY; JEWETT JR, 2011, p. 3).

Um exemplo de modelo de MHS bem comum é um bloco de massa m preso à ponta de uma mola, com o bloco livre para se mover sobre uma superfície horizontal, sem atrito (Figura 3.1). A posição de equilíbrio do sistema é em $x = 0$, situação em que a mola está com seu tamanho natural e a força que ela atua sobre o bloco é nula como mostra a Figura 3.1 (b).

Figura 3.1 – Sistema massa-mola



Adaptado de: Serway e Jewett Jr (2011, p. 3).

Quando o bloco é deslocado para uma posição x , como mostram as figuras 3.1 (a) e 3.1 (c), a mola exerce sobre ele uma força proporcional a sua posição, dada pela Lei de Hooke:

$$F_M = -k \cdot x \quad (3.5)$$

F_M é chamada de força restauradora, porque ela sempre aponta para a posição de equilíbrio e, portanto, tem sentido oposto ao deslocamento do bloco a partir $x = 0$. Como resultado, se tirado de sua posição de equilíbrio, o sistema oscila para frente e para trás. Aplicando a Segunda Lei de Newton ao movimento do bloco:

$$F_M = m \cdot a \quad (3.6)$$

$$-k \cdot x = m \cdot \frac{d^2x}{dt^2} \quad (3.7)$$

A equação de movimento é:

$$\frac{d^2 x}{dt^2} + \frac{k}{m} \cdot x = 0 \quad (3.8)$$

Uma forma de resolver esta equação diferencial é usar uma propriedade que a derivada de segunda ordem da função exponencial apresenta:

$$\frac{d^2}{dt^2} (e^{\beta t}) = \frac{d}{dt} (\beta e^{\beta t}) = \beta^2 \cdot e^{\beta t} \quad (3.9)$$

E para um número complexo $i\omega$, em que $i = \sqrt{-1}$ e ω é real:

$$(i\omega)^2 = (\sqrt{-1})^2 \omega^2 = -\omega^2 \quad (3.10)$$

Combinando as equações 3.9 e 3.10, é possível para a equação 3.8 encontrar soluções do tipo:

$$x(t) = Ae^{i(\omega t + \varphi_0)} \quad (3.11)$$

Sendo A e φ_0 constantes quaisquer. Mas, pela fórmula de Euler:

$$e^{i\beta} = \cos\beta + i\sin\beta \quad (3.12)$$

E usando $Re(x(t))$ e $Im(x(t))$ para representar as partes real e imaginária de $x(t)$, respectivamente, encontra-se:

$$x(t) = Re(Ae^{i(\omega t + \varphi_0)}) = A\cos(\omega t + \varphi_0) \quad (3.13)$$

ou

$$x(t) = Im(Ae^{i(\omega t + \varphi_0)}) = A\sin(\omega t + \varphi_0) \quad (3.14)$$

Como as equações 3.13 e 3.14 são equivalentes do ponto de vista físico, ambas podem ser adotadas como solução da eq. 3.8.

O termo A é a amplitude do movimento, φ_0 é a fase inicial, ambos arbitrários e que podem ser determinados a partir das condições iniciais. Já o termo ω , chamado muitas vezes de frequência angular, pode ser encontrado calculando a segunda derivada da posição, por exemplo, dada pela eq. 3,13:

$$x(t) = A\cos(\omega t + \varphi_0) \quad (3.15)$$

$$\frac{d}{dt}x(t) = -A\omega\sin(\omega t + \varphi_0) \quad (3.16)$$

$$\frac{d^2}{dt^2}x(t) = -A\omega^2\cos(\omega t + \varphi_0) \quad (3.17)$$

E aplicando as equações 3.15 e 3.17 na equação de movimento 3.8:

$$-A\omega^2\cos(\omega t + \varphi_0) + \frac{k}{m} \cdot A\cos(\omega t + \varphi_0) = 0 \quad (3.18)$$

$$\omega^2 = \frac{k}{m} \quad (3.19)$$

$$\omega = \sqrt{\frac{k}{m}} \quad (3.20)$$

O período do MHS é então calculado pela condição de periodicidade:

$$x(t) = x(t + T) \quad (3.21)$$

$$A\cos(\omega t + \varphi_0) = A\cos(\omega t + \omega T + \varphi_0) \quad (3.22)$$

Mas como a função cosseno tem período 2π :

$$\omega T = 2\pi \quad (3.23)$$

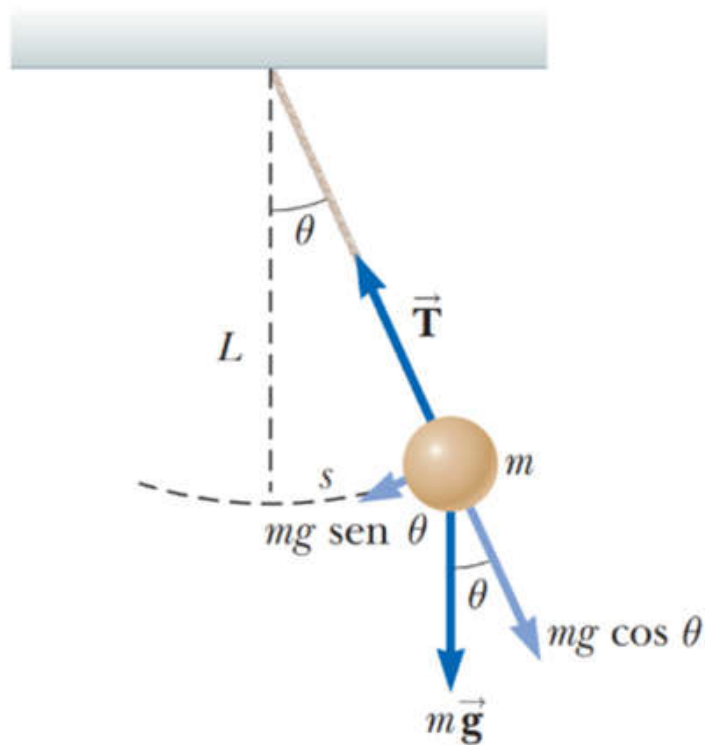
Finalmente, combinando 3.20 e 3.23, chega-se a:

$$T = 2\pi \sqrt{\frac{m}{k}} \quad (3.24)$$

3.4 PERÍODO DO PÊNDULO SIMPLES

O pêndulo simples é um sistema constituído por uma partícula de massa m , suspensa por um fio de massa muito inferior a massa da partícula, com comprimento L e fixado à extremidade superior, como mostrado na Figura 3.2.

Figura 3.2 – O Pêndulo Simples



Fonte: Serway e Jewett Jr (2011, p. 15).

O movimento ocorre em um plano vertical e descreve um arco de circunferência. Se o ângulo θ for pequeno (menor que aproximadamente 10°), o movimento é muito próximo do oscilador harmônico simples (SERWAY; JEWETT JR, 2011, p. 3).

Na partícula atuam apenas a força de tração \vec{T} exercida pelo fio e a força de gravidade $m\vec{g}$. Como a tração é equilibrada pela componente perpendicular da força gravitacional, a força restauradora é:

$$F_{tan} = -m \cdot g \cdot \text{sen}\theta \quad (3.25)$$

E aplicando a Segunda Lei de Newton para o movimento tangencial:

$$F_{tan} = m \cdot a_{tan} \quad (3.26)$$

$$-m \cdot g \cdot \text{sen}\theta = m \cdot \frac{d^2s}{dt^2} \quad (3.27)$$

Mas $s = L\theta$, então:

$$-g \cdot \text{sen}\theta = L \cdot \frac{d^2\theta}{dt^2} \quad (3.28)$$

Assim, encontra-se a seguinte equação de movimento:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \cdot \text{sen}\theta = 0 \quad (3.29)$$

A presença da função seno torna essa equação não linear, impedindo a obtenção de uma equação analítica exata formada apenas com funções elementares. Aproximações analíticas de alta precisão para o período podem ser obtidas utilizando diversas abordagens. Uma delas é o uso das séries de

potências que produzem razoável precisão. Outra técnica, a média aritmético-geométrica que usa a integral elíptica completa do primeiro tipo é tão eficiente que fazendo apenas três iterações já se obtém uma aproximação com erro abaixo de 1% do valor exato para amplitudes com até 179° (CARVALHAES; SUPPES, 2009).

Para se evitar toda essa complexidade extra é muito comum limitar o movimento a ângulos pequenos de modo a se poder fazer a aproximação:

$$\text{sen}\theta \approx \theta \quad (3.30)$$

Assim o movimento passa a ser harmônico simples com equação:

$$\frac{d^2\theta}{dt^2} + \frac{g}{L} \cdot \theta = 0 \quad (3.31)$$

E por comparação direta com as equações 3.8 e 3.19 chega-se a:

$$\omega^2 = \frac{g}{L} \quad (3.32)$$

O período é então dado por:

$$T = 2\pi \sqrt{\frac{L}{g}} \quad (3.33)$$

Essa aproximação harmônica apresenta dois problemas básicos em aplicações fora do campo didático. O primeiro é que ela produz resultados numéricos imprecisos para amplitudes de oscilação fora do regime de pequenas oscilações (CARVALHAES; SUPPES, 2009).

Para corrigir esse problema, PALMA et al. (2009), usa a expansão em série de potências da função seno:

$$\text{sen}\theta \approx \theta - \frac{\theta^3}{3!} + \frac{\theta^5}{5!} - \frac{\theta^7}{7!} + \dots \quad (3.34)$$

Assim, o período do pêndulo passa a ser determinado por:

$$T = 2\pi \sqrt{\frac{L}{g}} \left[1 + \frac{1}{16} \theta_{máx}^2 + \frac{11}{3072} \theta_{máx}^4 + \frac{25}{18432} \theta_{máx}^6 \dots \right] \quad (3.35)$$

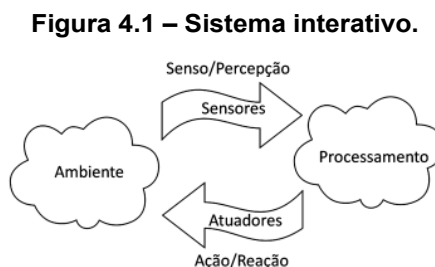
A equação 3.35 mostra que as correções vão diminuindo continuamente. Para fazer um cálculo mais preciso, basta adicionar mais um termo da expansão, tendo como erro o termo seguinte.

O segundo problema da aproximação harmônica é que ela apresenta isocronismo do pêndulo, ou seja, para pequenas oscilações o seu período independe da amplitude do movimento. Esse comportamento contrasta com o do pêndulo real, chamado de anisocronismo, para o qual o período cresce com a amplitude e é facilmente verificado pela equação 3.35 (CARVALHAES; SUPPES, 2009).

4 A PLATAFORMA ARDUINO

4.1 AQUISIÇÃO DE DADOS, MONITORAMENTO E CONTROLE

A aquisição de dados pode ser definida, de maneira simples, como a medição de informações do mundo real. Um sistema de aquisição de dados realiza a coleta de parâmetros de um fenômeno físico real, faz sua conversão num sinal elétrico e o formata de modo conveniente para seu armazenamento, processamento, visualização e análise. Os sistemas de monitoramento e controle verificam continuamente os dados recebidos e enviam sinais em resposta aos valores recebidos e executam ações que dependem desses valores (PAMBOUKIAN; PEDROTTI, 2012), ao que Banzi (2011, p. 27) chama de processo interativo e está esquematizado na Figura 4.1.



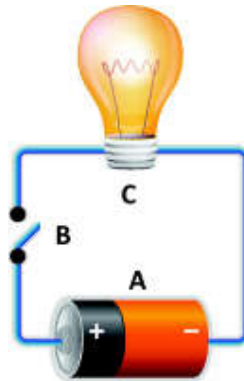
Adaptado de Banzi (2011, p. 27).

As quantidades físicas a serem medidas podem ser as mais variadas (luminosidade, temperatura, pressão, força, deslocamento, etc.), mas todas relacionadas com algum tipo de energia. Os dispositivos que convertem uma forma de energia em outra são denominados transdutores. Assim, um transdutor elétrico converte alguma grandeza não elétrica em sinais elétricos e vice-versa e são classificados em sensores e atuadores. Os sensores convertem alguma grandeza medida no mundo real em sinais elétricos. Já os atuadores são transdutores que podem converter um sinal elétrico em uma ação física de reação (BANZI, 2011, p. 27).

Para realizar o processamento e a análise dos dados é necessário um sistema eletrônico, que não necessariamente precisa usar a tecnologia da informática. Mas, com o barateamento e aumento de capacidade e velocidade dos microcontroladores, é cada vez mais comum o seu uso juntamente com sensores e atuadores no desenvolvimento de dispositivos interativos eletrônicos capazes de realizar aquisição de dados, monitoramento e controle de sistemas (BANZI, 2011, p. 27).

De um ponto de vista da física elétrica, qualquer dispositivo interativo eletrônico se baseia no funcionamento de um circuito elétrico simples, como o mostrado na Figura 4.2. Quando a chave interruptora manual (item B da Figura 4.2) está aberta não há corrente elétrica circulando. Porém, quando ela é fechada, passa a fluir uma corrente elétrica que leva ao funcionamento do receptor elétrico (item C da Figura 4.2).

Figura 4.2 – Circuito elétrico simples



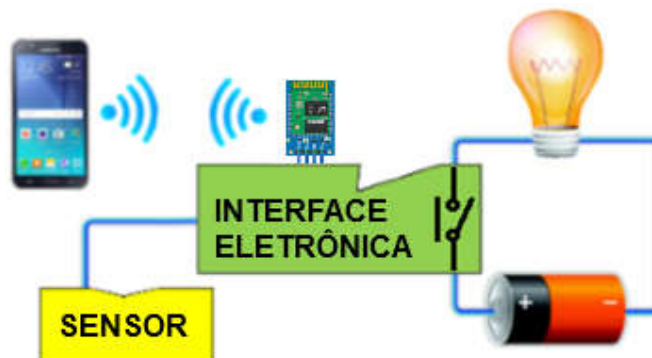
A = Gerador elétrico B = Chave interruptora C =Receptor elétrico

Adaptado de: <https://vestibular.brasilecola.uol.com.br/enem/abordagem-dos-circuitos-eletricos-no-enem.htm>. Acesso em: 25 jun 2020.

A substituição da chave interruptora manual da Figura 4.2 por uma interface eletrônica controlada por um computador ou microcontrolador, como mostra a Figura 4.3, leva a automatização do funcionamento circuito, com o seu

controle podendo ser realizado com uso de sensores e também com equipamentos de comunicação sem fio, como por exemplo, via Bluetooth.

Figura 4.3 – Circuito elétrico automatizado.



Adaptado de Brasil Escola¹, Samsung² e Embarcados³.

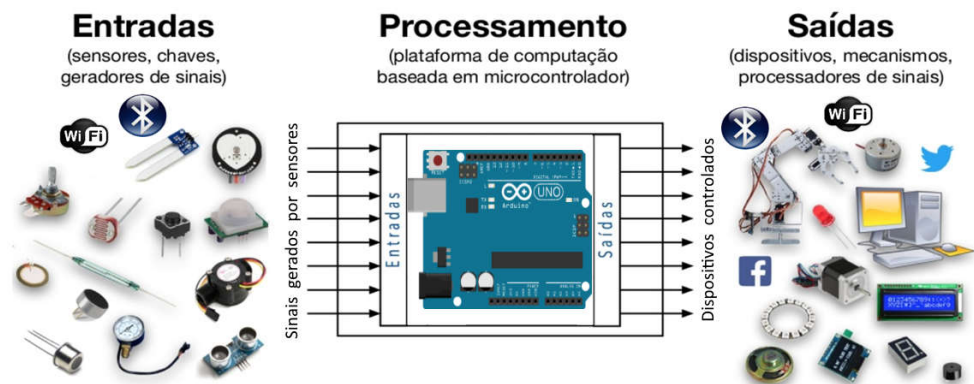
É importante observar que, como mostra a Figura 4.4, existe uma diversidade de sensores, chaves e geradores de sinais que podem ser usados como entrada de dados e imensa quantidade de dispositivos, mecanismos e processadores de sinais usados como saída de dados e é possível usar comunicação por USB, ethernet, Bluetooth, wifi, GSM, etc. Além disso, todos esses dispositivos e tecnologias têm custo cada vez mais baixo e são cada vez mais fáceis de serem adquiridos no mercado especializado. Tudo isso permite o desenvolvimento de uma infinidade de projetos.

¹ Disponível em: <https://vestibular.brasilecola.uol.com.br/enem/abordagem-dos-circuitos-eletricos-no-enem.htm>. Acesso em: 25 jun. 2019.

² Disponível em: <https://www.samsung.com/br/smartphones/galaxy-j5-j500m/SM-J500MZZTO/>. Acesso em: 25 jun. 2019.

³ Disponível em: <https://www.embarcados.com.br/modulos-bluetooth-hc-05-e-hc-06/>. Acesso em: 21 jul. 2019.

Figura 4.4 – Dispositivos de entrada e saída em um projeto Arduino.



Adaptado de: <https://pt.slideshare.net/helderarocha/introduo-ao-arduino-81351703>.
Acesso em: 30 jul. 2019.

4.2 O MICROCONTROLADOR ARDUINO

O Arduino foi desenvolvido em 2005 no *Interaction Design Institute*, na cidade italiana de Ivrea e o nome Arduino foi dado à nova placa em referência a um bar frequentado por professores e estudantes desse instituto. Foi idealizado por Massimo Banzi, um professor preocupado em encontrar um meio barato e fácil de seus estudantes de arte e design trabalharem com tecnologia, em contrapartida aos produtos existentes até então, caros e relativamente difíceis de usar. As duas exigências básicas eram, primeiro que fosse barato e com preço não podendo ser maior que o custo da pizza que um estudante fosse comprar e em segundo lugar que fosse um sistema que qualquer um pudesse utilizar. Assim Banzi solicitou a David Cuartielles, pesquisador visitante da Universidade sueca de Malmö, desenhar os circuitos da placa, ao mesmo tempo em que David Mellis, aluno de Banzi, desenvolvia o software de controle da placa. Em seguida, um engenheiro local chamado Gianluca Martino foi contratado para orientar os alunos do *Design Institute* no desenvolvimento dos projetos (EVANS; NOBLE; HOCHENBAUM, 2013, p. 25-26)

Como era trabalhoso para os estudantes encontrar todos os componentes e dispositivos necessários, foram produzidos kits contendo as placas e os itens eletrônicos mais comuns dos seus projetos. Foi um grande sucesso e o estoque

inicial de duzentos kits foi rapidamente vendido, sendo necessária a fabricação de mais unidades para suprir a demanda. Designers, artistas e profissionais de outras áreas se interessaram pela novidade e passaram a usá-lo em seus projetos e a sua popularidade cresceu ainda mais. O sucesso foi confirmado com a obtenção de uma menção honrosa na categoria Comunidades Digitais em 2006, pela *Prix Ars Electronica*, além da marca de mais de 50.000 placas vendidas até outubro de 2008. O grande público percebeu que além de sua relativa facilidade de uso e custo baixo ele era uma excelente introdução para programação de microcontroladores e dessa forma poderia ser usado por qualquer pessoa interessada (WIRED, 2019).

O Arduino é uma plataforma eletrônica baseada em hardware livre e software de código aberto. Assim, pode-se comprar uma placa Arduino pronta ou adquirir seus componentes separadamente e conectá-los de acordo com as necessidades do projeto a ser desenvolvido. É possível encontrar, principalmente em sites da internet, todas as rotinas necessárias ao projeto já prontas para uso ou somente parte delas e desenvolver só o restante que falta ou ainda desenvolver totalmente os algoritmos necessários ao projeto. Tudo dependendo da necessidade, tempo e interesse do usuário (ARDUINO INTRODUÇÃO, 2019).

Ele pode ser usado para desenvolver objetos interativos independentes ou ser conectado a um computador, a uma rede ou até mesmo à internet para recuperar e enviar dados do Arduino e trabalhar com eles. Assim, pode ser usado em soluções IoT⁴, *wearable* (vestível), impressão 3D e ambientes incorporados. Como exemplo simples, pode enviar um conjunto de dados recebidos de sensores para um site e exibi-los na forma de tabela ou de um gráfico (ARDUINO INTRODUÇÃO, 2019).

⁴ Acrônimo de *Internet of Things* e significando internet das coisas.

Com o passar dos anos, o Arduino tem permitido o desenvolvimento de milhares de projetos, desde objetos do cotidiano até instrumentos científicos complexos. Uma crescente comunidade mundial de entusiastas (estudantes, amadores, artistas, programadores e profissionais de diversas áreas) utilizando de forma intensa essa plataforma contribuiu para o desenvolvimento de uma incrível quantidade de literatura livre, de fácil acesso e de grande ajuda para iniciantes e especialistas (ARDUINO INTRODUÇÃO, 2019).

Uma pesquisa na Internet por “Arduino” mostrará um vasto número de sites dedicados ao Arduino e que apresentam projetos interessantes em que ele foi utilizado para ler dados e controlar uma enorme quantidade de dispositivos. Por isso ele é um produto incrível e com um pouco de vontade para aprender como usá-lo será possível a criação de quase tudo, desde obras de arte interativas até robôs (MCROBERTS 2011).

O Arduino original tem sua programação feita diretamente pelo IDE⁵ oficial do Arduino, o layout da placa é padronizada, está documentado no site oficial (ARDUINO HOME, 2019), está devidamente licenciado para exibir o nome e o logotipo do Arduino e é produzido por fabricantes autorizados e oficiais que atualmente são SmartProjects na Itália, Sparkfun nos EUA e DogHunter em Taiwan / China (ARDUINO CLONES, 2019).

Como ele tem hardware aberto, há muitas placas não oficiais disponíveis para compra com base no Arduino original, ou que podem ser criadas a partir de um diagrama de fácil compreensão. Assim é possível comprar os componentes apropriados e montar uma placa Arduino completamente funcional em uma matriz de pontos ou em uma PCB⁶ feita em casa. A equipe do Arduino impõe apenas que não seja utilizado o nome “Arduino” nem o seu logotipo, pois eles são de uso exclusivo da placa oficial (MCROBERTS, 2011).

⁵ Acrônimo de *Integrated Development Environment* e significando ambiente de desenvolvimento integrado.

⁶ Acrônimo de *Printed Circuit Board* e significando placa de circuito impresso.

O Arduino original ou oficial usa um microcontrolador da família ATmega (Oliveira 2018). Apenas no site do Arduino estão disponíveis para compra dezenas de diferentes tipos de placas, cada uma com uma finalidade específica, além de módulos para expansão (*shields*) e kits que estão agrupados em *Entry Level* (Nível de Entrada), *Enhanced Features* (Recursos Aprimorados), *IoT*, *Education* (Educação), *Wearable* (Vestível), além do grupo *Retired* (Aposentado) que mostra a história de diversos modelos de Arduino e placas auxiliares que foram descontinuados por ficarem obsoletos (ARDUINO PRODUTOS, 2019).

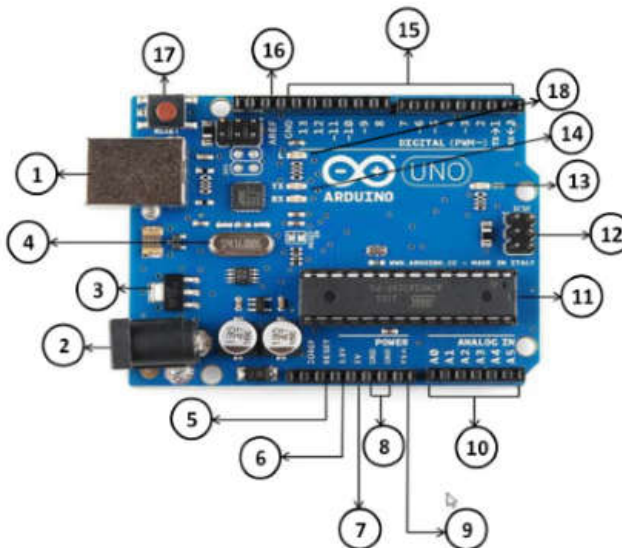
A placa Arduino Uno (Figura 4.5) é a primeira de uma série de placas USB Arduino com microcontrolador ATmega328P (componente 11 da Figura 4.5). Pertencente ao grupo *Entry Level*, é atualmente o modelo de referência para a plataforma Arduino, sendo a placa mais utilizada nos projetos em geral e a mais fácil de fazer a iniciação em Arduino (ARDUINO UNO, 2019).

Através do conector USB (componente 1 da Figura 4.5) o Arduino pode ser alimentado e controlado diretamente pelo computador. Para tornar-se autônomo, um programa deve ser desenvolvido no computador e transferido para a memória do Arduino também através da conexão USB. Depois de desconectado, é necessário que através do conector de alimentação (componente 2 da Figura 4.5) a placa receba energia de alguma fonte com tensão contínua entre 6 V e 20 V. (MONK, 2014, p. 1).

É importante observar que os pinos digitais de 0 a 13 podem ser configurados como pinos de entrada para ler valores lógicos (0 ou 1) ou como pinos de saída para acionar diferentes dispositivos como LEDs, relés, etc. Os pinos 3, 5, 6, 9, 10 e 11 são rotulados com “~” pois podem ser usados também para fornecer saída do tipo PWM⁷ (TUTORIALSPPOINT, 2019).

⁷ Acrônimo de *Pulse Width Modulation* e significando Modulação de Largura de Pulso.

Figura 4.5 – Arduino Uno



| Componente | Descrição |
|------------|--|
| 1 | Conector USB – Permite comunicação com um PC via cabo USB |
| 2 | Conector de alimentação – Entrada de tensão (6 V a 20 V DC). |
| 3 | Regulador de voltagem – Estabiliza as tensões DC em toda a placa. |
| 4 | Oscilador de cristal de 16 MHz – Temporizador do sistema. |
| 5 | Pino de reset da placa – Reinicializa a placa com botão externo. |
| 6 | Pino de 3,3 V – Fornece 3,3 V de saída. |
| 7 | Pino de 5 V – Fornece 5 V de saída. |
| 8 | Pino GND (Ground) – Pino terra e de retorno de corrente. |
| 9 | Pino Vin – Entrada de tensão por fonte de energia externa AC. |
| 10 | Pinos analógicos de A0 a A5. |
| 11 | Microcontrolador da placa – Atualmente no Uno é o ATmega328P. |
| 12 | Pino ICSP – Usado para transferir programas/firmwares e também para executar tarefas administrativas. |
| 13 | LED indicador – Indica que a placa está ligada corretamente. |
| 14 | LEDs TX e RX – O led TX pisca enquanto envia os dados seriais. A O led RX pisca durante o processo de recebimento. |
| 15 | Pinos digitais – A placa Arduino UNO possui 14 pinos de E/S digitais |
| 16 | Pino AREF – Define uma tensão de referência externa (entre 0 e 5 Volts) como o limite superior para os pinos de entrada analógica. |
| 17 | Botão de reset da placa – Reinicializa a placa. |
| 18 | LED built-in – conectado geralmente ao pino digital 13. |

Adaptado de: TUTORIALSPPOINT (2019).

4.3 A IDE ARDUINO

Para controlar a placa do Arduino é necessário desenvolver um *sketch* (código) que é um algoritmo de controle e fazer sua gravação na placa através de uma conexão USB. Esse desenvolvimento é feito no Arduino Software (IDE) e atualmente está disponível em duas opções (ARDUINO START, 2019):

a) IDE *on-line* (*Arduino Web Editor*) – Está disponível para execução na seção Software do site oficial do Arduino (ARDUINO SOFTWARE, 2019). Deve-se usar essa opção somente quando houver disponível uma conexão de Internet confiável. As placas originais e algumas placas clones licenciadas funcionam prontamente sem necessidade de nenhuma instalação adicional. Essa interface permite que os *sketchs* criados sejam salvos na internet, disponibilizando-os a partir de qualquer dispositivo online e fazendo backup automático. Sempre será executada a versão mais atualizada do IDE sem a necessidade de instalar atualizações ou bibliotecas geradas pela comunidade.

b) IDE *off-line* – Também está disponível na seção Software do site oficial do Arduino (ARDUINO SOFTWARE, 2019). Essa opção é indicada quando não houver disponível uma conexão confiável com a internet ou forem utilizadas placas clones não compatíveis com a opção *on-line*. É necessário baixar a IDE de acordo com o sistema operacional usado (Windows, Linux ou OS) e o próprio site sugere o uso da versão mais recente. Após o download deve-se fazer a instalação no computador que pode ser feita através do hiperlink **Getting Started**, disponível na própria página de download da IDE. Após a instalação aparece o ícone da IDE do Arduino indicado pela Figura 4.6 (ARDUINO START, 2019).

Figura 4.6 – Ícone da IDE do Arduino

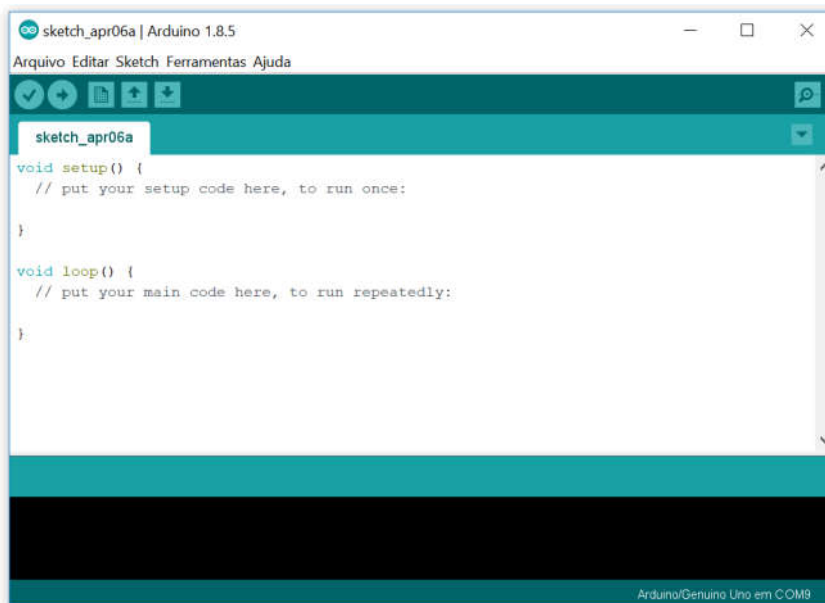


Fonte: print screen da IDE no sistema operacional Windows 10.

4.4 CONFIGURANDO A IDE OFF-LINE DO ARDUINO

Assim que a IDE for aberta aparece algo similar a Figura 4.7.

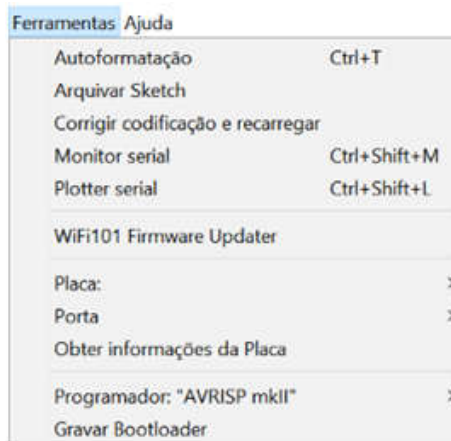
Figura 4.7 – IDE do Arduino quando aberto pela primeira vez



Fonte: print screen da IDE no sistema operacional Windows 10.

Acionando o item de menu **Ferramentas** clicar em **Placa** (Figura 4.8).

Figura 4.8 – Menu Ferramentas do Arduino



Fonte: print screen da IDE no sistema operacional Windows 10.

A seguir será apresentada a uma lista de placas (Figura 4.9). Deve-se selecionar a opção apropriada a partir dessa lista.

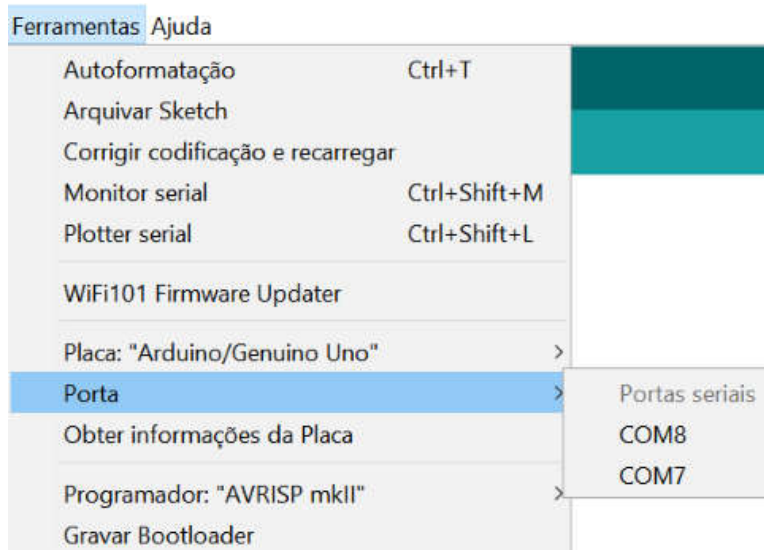
Figura 4.9 – Item de menu Placas do Arduino



Fonte: print screen da IDE no sistema operacional Windows 10.

Como último passo da configuração, deve-se acionar novamente o menu **Ferramentas**, clicar em **Porta** e escolher na lista a porta apropriada para a placa do Arduino (Figura 4.10).

Figura 4.10 – Item de menu Porta do Arduino



Fonte: print screen da IDE no sistema operacional Windows 10.

O IDE do Arduino é bem simples, e permite rápido aprendizado à medida que é utilizado. Conforme o usuário se torna mais hábil no uso da IDE padrão, pode ser considerado experimentar alguma IDE profissional, como Eclipse, ArduIDE, GNU/Emacs, AVR-GCC, AVR Studio e até mesmo o XCode da Apple (MCROBERTS, 2011).

4.5 A LINGUAGEM DE PROGRAMAÇÃO DO ARDUINO

A linguagem de programação do Arduino é baseada em Wiring (ARDUINO INTRODUÇÃO, 2019). Wiring é uma linguagem de programação de código aberto para microcontroladores e se baseia no Processing, um projeto aberto desenvolvido a partir das idéias surgidas no *MIT Media Lab* (WIRING, 2019).

Um programa em Arduino é chamado *sketch* e é composto por uma sequência de instruções que apenas podem ser escritas com palavras-chave e que seguem um conjunto bem definido de regras da linguagem (MCROBERTS, 2011).

O termo função na programação do Arduino é extremamente importante e tem vários significados. Primeiro, ele pode ser usado com o mesmo sentido de uma função matemática, pois recebe parâmetros como entrada (o domínio da função) e retorna um valor de saída (a imagem da função). Como exemplo simples, a execução da instrução com sintaxe **sqrt(25)** calcula a raiz quadrada de 25. Ela tem como parâmetro de entrada o valor 25 e como parâmetro de retorno ou de saída o valor 5. Aqui o termo função está associado a apenas uma única instrução pertencente ao grupo de palavras-chave do Arduino. Podem existir funções sem parâmetros de entrada e nesse caso não há nada entre seus parênteses e podem existir funções sem parâmetros de saída em que é obrigatório colocar a palavra-chave **void** antes de serem declaradas. Assim, a instrução **void setup()** indica que `setup` é uma função sem parâmetros de entrada e também sem parâmetros de saída (MCROBERTS, 2011).

Em um sentido diferente, uma função pode ser também um conjunto de comandos agrupados em um bloco de código que se inicia com o símbolo { e termina com o símbolo }. Tudo entre esses dois símbolos fará parte da função. Neste sentido, uma função pode ter muitas linhas de código, cada linha contendo uma ou várias instruções e pode-se executar esse código inúmeras vezes simplesmente chamando o nome da função, em vez de ter de reescrever todo o código cada vez que for usá-lo. É obrigatório para um *sketch* Arduino ter uma função **setup()** e uma função **loop()**, do contrário, ele não funcionará. Por não terem nenhum parâmetro de entrada não há nada entre seus parênteses. Ambas devem ser precedidas da palavra-chave **void** para indicar que são funções que não retornam nenhum parâmetro. A Figura 4.7 mostra uma possível imagem da IDE logo após sua abertura e nela as duas funções podem ser vistas (MCROBERTS, 2011).

Assim que o Arduino é energizado ou imediatamente após o término da transferência do *sketch* entre o computador e a memória do Arduino, a função **setup()** é a primeira parte do *sketch* a ser executada e sempre uma única vez. Nela estão contidas as instruções de inicialização do programa, como a definição dos modos dos pinos, das taxas de transmissão serial etc. Ela inicia com **void setup()** para informar ao compilador que é uma função chamada **setup**, que não retorna nenhum dado (**void**) e que não recebe nenhum parâmetro (parênteses vazios) (MCROBERTS, 2011).

A função **loop()** é a principal função do *sketch* e inicia sua execução logo após o **setup**. As declarações dentro dessa função são executadas uma por vez até a sua última linha, quando é reiniciando o loop num processo repetitivo até que o Arduino seja desligado ou o botão Reset pressionado. De modo similar ao **setup**, ela inicia com **void loop()** para informar ao compilador que não retorna nenhum dado e que não recebe nenhum parâmetro (MCROBERTS, 2011).

Por exemplo, se num programa for desenvolvida uma função (ou um bloco de código) de nome **myFunc**, em que a sintaxe da escrita seja algo como:

```
int myFunc (int x, int y) {  
    // instruções da função  
}
```

Então a função **myFunc** recebe como entrada dois números inteiros, x e y. E quando seu processamento for concluído, será retornado um valor inteiro no ponto de execução imediatamente após onde ela tiver sido chamada no programa. Para indicar que ela retorna um valor inteiro é obrigatório preceder seu nome com a palavra-chave `int`. Pelo mesmo motivo deve-se fazer isso com cada parâmetro de entrada inteiro (MCROBERTS, 2011). É sempre necessário preceder o nome da função e cada um de seus parâmetros de entrada com a correspondente palavra-chave indicativa tipo de dado (ARDUINO LINGUAGEM, 2019).

À medida que um *sketch* se torna mais complexo, e passar a ter dezenas, centenas ou talvez milhares de linhas, comentários serão vitais para facilitar a compreensão de como cada seção funciona. Pode-se desenvolver um trecho incrível de código, mas provavelmente não será possível lembrar-se de como ele funciona quando for tentar entendê-lo dias, semanas, meses ou até anos após a sua criação. Por isso, inserir comentários é essencial para a compreensão do código. Além disso, faz parte do espírito do Arduino, e de toda a comunidade de fonte aberta, o compartilhamento de código e projetos. E o uso dos comentários pode ajudar muito a alguém interessado na compreensão de um determinado algoritmo (MCROBERTS, 2011).

Um comentário que ocupa toda uma linha ou parte dela deve iniciar com `//`, e qualquer texto a direita será desconsiderado no momento da execução dessa linha do programa. Caso o comentário possua várias linhas é melhor usar outro formato: um bloco com linhas de texto entre os sinais `/*` e `*/`. Como exemplo, as duas formas são usadas para fazer comentários no *sketch* Blink (ARDUINO BLINK, 2019).

As instruções da linguagem de programação do Arduino estão agrupadas em três partes principais: **functions** (funções), **values** (valores variáveis e constantes) e **structure** (estrutura). Na IDE do Arduino as instruções do grupo **functions** são escritas em vermelho, as do grupo **values** então em azul e aquelas do grupo **structure** em marrom. Já os comentários ficam em cinza e os demais termos numéricos e alfanuméricos não pertencentes ao grupo de palavras-reservadas ficam na cor preta (ARDUINO LINGUAGEM, 2019).

O grupo **functions** reúne as instruções que servem para controlar a placa do Arduino e realizar cálculos matemáticos. Como exemplo, no trecho de código:

```
void setup() {  
  pinMode(13, OUTPUT); // configura o pino digital 13 como saída  
}
```

A função **setup** é usada para configurar o pino digital 13 para o modo de saída de dados através do uso da função **pinMode**, que pertence ao grupo **functions** da linguagem do Arduino. É importante observar que por exigência da linguagem, é obrigatório finalizar cada instrução com um ponto e vírgula. Se desejado, após uma instrução pode-se colocar outra instrução ou um comentário (ARDUINO LINGUAGEM, 2019).

O grupo **values** contém as constantes e variáveis que serão usadas ao longo do programa. Constantes são expressões predefinidas na linguagem Arduino. Elas são usadas para tornar os programas mais fáceis de ler. É importante observar que a linguagem diferencia caracteres maiúsculos e minúsculos. Como no exemplo, na instrução **pinMode** é necessário especificar como primeiro parâmetro o número do pino e como segundo parâmetro o comportamento elétrico do pino, ou seja, se é um pino de entrada (INPUT) ou de saída (OUTPUT). Pinos configurados como INPUT adquirem um estado de alta impedância e consomem correntes extremamente pequenas no circuito que eles estão conectados, equivalente a um resistor em série de 100 MΩ conectado ao pino. Isso os torna úteis para ler um sensor. Pinos configurados como OUTPUT

passam a ter um estado de baixa impedância, podendo fornecer até 40 mA de corrente para outros circuitos. Isso os torna úteis para alimentar LEDs e dispositivos que usam menos de 40 mA de intensidade de corrente. Cargas maiores que 40 mA, como motores, exigirão um transistor ou outro circuito de interface (ARDUINO LINGUAGEM, 2019).

No grupo **values**, além das constantes, pode-se trabalhar com variáveis. Uma variável é um local na memória do computador em que é possível armazenar dados de diversos tipos, tais como texto (**string**), inteiro (**int**), decimal (**float**) etc. Assim, uma variável de tipo **int** é um número inteiro entre -32 768 e 32 767. Logo, segundo Mcroberts 2011, a instrução:

```
int ledPin = 10;
```

Atribui à variável de tipo inteiro o nome **ledPin** e dá a ela um valor de 10.

Pelas regras da linguagem Arduino, todas as variáveis devem iniciar com uma letra maiúscula ou minúscula e o restante do nome pode ser formado por letras, números e underscores. É proibido utilizar palavras-chave da linguagem como nomes de variáveis. Para ajudar a evitar isso, todas as palavras-chave no *sketch* serão mostradas em vermelho, azul ou verde (MCROBERTS, 2011).

Toda variável pode ter um escopo local ou global em relação ao programa. Se ela for global deverá ser declarada no início do algoritmo e antes da função setup e existirá enquanto o Arduino estiver em funcionamento, podendo ser acessada e alterada por qualquer função do algoritmo. Já uma variável local deverá ser declarada dentro do bloco de instruções de uma função. Poderá ser vista e modificada apenas pelo código dentro do seu próprio bloco e será destruída ao final da execução dessa função. Assim, no trecho de código:

```
int ledPin = 10;

void setup() {
  int x = 0;
  // instruções da função
}
```

São criadas a variável global inteira **ledPin** e a variável local inteira **x**. A variável **x** é uma variável local e só poderá ser usada dentro da função **setup**. Já a variável **ledPin** é uma variável global e poderá ser utilizada em qualquer outra função do algoritmo (MCROBERTS, 2011).

Structure é o terceiro grupo da linguagem Arduino e contém os elementos do *sketch* (**setup** e **loop**), de controle de estrutura (**if**, **for**, **while**, etc.), dos operadores (aritméticos, de comparação, booleanos, etc) e elementos de sintaxe adicional (**#define**, **#include**, etc.) (ARDUINO LINGUAGEM, 2019).

4.5.1 Controle de acionamento do LED built-in

Um LED é um *Light Emitting Diode* (Diodo Emissor de Luz). Um diodo é um dispositivo que permite a corrente elétrica passar em um único sentido por ele. Caso a corrente tente passar no sentido oposto, o diodo impede que ela o faça. Diodos podem ser úteis para prevenir que se faça uma conexão errada que reverta o sentido da corrente e danifique algum componente. Os LEDs impedem essa reversão e também emitem luz. Eles são fabricados com diferentes tamanhos, níveis de luminosidade e cores, incluindo o espectro ultravioleta e infravermelho como os LEDs dos controles remotos do televisor, condicionador de ar e outros aparelhos eletrônicos (MCROBERTS, 2011).

Na placa Arduino existem quatro LEDs. O primeiro é o *Power LED*, na cor verde, que permanece aceso enquanto o Arduino estiver em funcionamento. Outros dois LEDs, para comunicação serial, na cor laranja, só ficam acesos quando a placa envia ou recebe dados do computador através do cabo USB. O *TX LED* informa quando o Arduino envia dados para o PC e o *RX LED* informa

quando o Arduino recebe dados do PC. Finalmente, o *Status LED* ou *LED built-in* (embutido na placa) ou ainda *LED on-board*, também de cor laranja, está conectado eletricamente a um dos pinos digitais (ELECTRICARENA, 2019).

Para referir-se ao *LED built-in* em um comando é usada a constante predefinida LED_BUILTIN. Ela tem valor inteiro e igual ao do pino digital a que o LED está conectado e na grande maioria das placas é o pino 13. Se houver certeza do número do pino, esse valor pode ser usado no lugar de LED_BUILTIN (ARDUINO CONSTANTES, 2019).

A IDE Arduino apresenta, no menu Arquivo, item Exemplo, uma grande quantidade de exemplos de *sketchs* que podem ser abertos, analisados estudados e modificados. No subitem Exemplos embutidos 01.Basics, encontra-se o Blink (pisca), o exemplo ideal para início de estudo do Arduino. Sua codificação precisa de poucos comandos para fazer o *LED built-in* de forma repetitiva acender por um segundo e em seguida apagar por um segundo. Sem as linhas de comentário o seu algoritmo é:

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

A primeira e única linha do setup simplesmente informa através do comando **pinMode** que o pino LED_BUILTIN será de saída e por isso recebe o parâmetro OUTPUT. Essa indicação deverá ser sempre usada quando se desejar acionar um dispositivo conectado a qualquer porta digital. Caso o pino fosse usado para fazer leitura de um sensor deveria ser usado o parâmetro INPUT (ARDUINO CONSTANTES, 2019).

Dentro da função `loop`, **`digitalWrite`** é a primeira instrução e permite definir um valor HIGH ou LOW para o pino. Quando se define um pino como HIGH, ele passa a receber entre 4,5 a 5,0 volts do Arduino. Quando se define como LOW, o pino recebe entre 0 a 0,8 a volts, tornando-se um terra. Assim, a primeira instrução envia 5 V para o pino LED_BUILTIN e acende o *LED built-in* (ARDUINO CONSTANTES, 2019).

Quando um pino é configurado com saída, ele é posto em estado de baixa impedância e pode fornecer (drenar) até 40 mA de corrente para um circuito externo. Essa configuração é perigosa para a placa Arduino. Por isso, quando um pino for usado como saída de dados, deve sempre haver a preocupação em não ultrapassar o limite de corrente e danificar o pino ou mesmo queimar o microcontrolador (ARDUINO PINOS DIGITAIS, 2019).

É muito importante também observar que pelo *datasheet* do microcontrolador do Arduino Uno, o ATmega 3208, a soma das correntes drenadas por todos os pinos não pode ultrapassar 200 mA com temperatura de 25 °C e apenas 100 mA quando a 125 °C (ATMEGA, 2019, p. 15).

A segunda instrução do `loop` usa o comando **`delay`** e simplesmente informa ao Arduino para esperar 1000 milissegundos (um segundo) antes de executar a próxima instrução. A terceira instrução do `loop` anula a tensão que vai para o pino LED_BUILTIN apagando o LED. Por fim a quarta instrução, um **`delay`**, manda que se espere por mais 1000 milissegundos para ir para o final do `loop` e reiniciar o algoritmo num processo infinito (MCROBERTS, 2011).

Uma possível melhoria desse *sketch* é:

```
int pinoLED = LED_BUILTIN;

void setup() {
  pinMode(pinoLED, OUTPUT);
}
```

```

void loop() {
  digitalWrite(pinoLED, HIGH);
  delay(1000);
  digitalWrite(pinoLED, LOW);
  delay(1000);
}

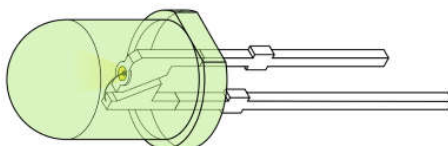
```

No trecho de código, através da instrução *int*, é criada uma variável global inteira para armazenar o número do pino digital usado como parâmetro nos comandos **pinMode** e **digitalWrite**. A grande vantagem dessa modificação é que se por qualquer motivo for necessário usar outro pino, basta alterar uma única linha, a primeira, simplesmente inserindo o número do pino desejado.

4.5.2 Controle de acionamento de um LED externo

Em um exame cuidadoso de um LED percebem-se dois detalhes: os terminais têm comprimentos diferentes, e um lado do LED é chanfrado, em vez de cilíndrico (Figura 4.11). Esses detalhes servem para indicar qual terminal é o ânodo (positivo) e qual é o cátodo (negativo): o terminal mais comprido (ânodo) é conectado à tensão positiva e o terminal mais curto e com o lado chanfrado (cátodo) vai para o pino terra ou pino negativo. Se o LED for conectado ao contrário, isso não o danificará se ele não tiver que conduzir correntes muito elevadas.

Figura 4.11 – Detalhe de um LED verde com 5mm de tamanho



Adaptado de: Wikipédia⁸

De acordo com o *datasheet* da Atmega, um pino digital emite 5 V com corrente máxima de 40 mA (Mcroberts 2011). Já o *datasheet* de um LED de alto

⁸ Disponível em: [https://commons.wikimedia.org/wiki/File:LED,_5mm,_green_\(en\).svg#/media/File:LED,_5mm,_green_\(unlabelled,_full\).svg](https://commons.wikimedia.org/wiki/File:LED,_5mm,_green_(en).svg#/media/File:LED,_5mm,_green_(unlabelled,_full).svg). Acesso em: 05 abr. 2019.

brilho branco de 5 mm (LED, 2019) indica que ele suporta uma corrente máxima de 20 mA, situação em que recebe entre seus terminais uma tensão da ordem de 3 V. Assim, ligar diretamente um LED a uma porta digital irá sobrecarregá-lo, sendo por isso necessário utilizar alguma estratégia para limitar a corrente que atravessa o LED. A mais simples é a utilização de um resistor em série a ele.

Um resistor é um dispositivo projetado para provocar resistência a uma corrente elétrica e quando conectado em série a um dispositivo, tem-se como resultado uma queda da corrente que atravessa a ambos e também a diminuição da tensão nos terminais do dispositivo. Assim, para limitar a corrente que atravessa o LED e evitar a sua queima, é necessário reduzir os 5 V do pino digital do Arduino para os 3 V requeridos pelo LED. Logo, o terminal curto do LED (cátodo) deve ser conectado ao terra (GND), o terminal longo (ânodo) deve ser ligado em série com um resistor e este deve ser conectado a um dos pinos digitais. Para determinar o valor R da resistência adequada é necessário usar a primeira lei de Ohm:

$$R = \frac{V}{I} \quad (4.1)$$

Em que V é a tensão nos terminais do resistor e I a corrente que o percorre. Mas a tensão no resistor deve ser a diferença entre a tensão fornecida pelo pino digital (V_S) e a tensão requerida pelo LED (V_L), então:

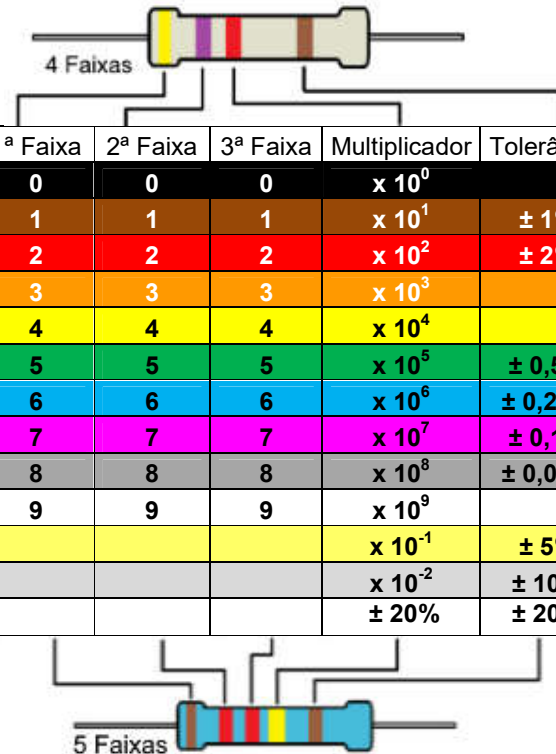
$$R = \frac{V_S - V_L}{I} \quad (4.2)$$

Como $V_S = 5$ V, $V_L = 3$ V e $I = 20$ mA, o valor necessário da resistência seria de $R = (5 - 3) / 0,02$, ou seja, um valor de 100 Ω . É prudente sempre escolher um resistor com valor mais próximo e maior ao valor necessário, caso contrário uma corrente muito intensa pode atravessar o resistor e danificar o LED (MCROBERTS, 2011). Mas, é comum o uso de resistores com valores bem

maiores, com 220Ω até $1 \text{ k}\Omega$ (ARDUINO BLINK, 2019), tendo como consequência apenas a diminuição da luminosidade emitida pelo LED.

Em geral os resistores são fabricados com valores-padrão e como são muito pequenos para conter informações de fácil leitura é utilizado um código de cores. Ao redor do resistor encontram tipicamente quatro anéis de cores e é possível descobrir o valor de sua resistência utilizando o código de cores contido no Quadro 4.1. Resistores com três, cinco ou seis faixas também são fabricados, mas são de uso muito mais restrito que aqueles com quatro faixas.

Quadro 4.1 – Código de cores dos resistores



| Cor | 1ª Faixa | 2ª Faixa | 3ª Faixa | Multiplicador | Tolerância |
|----------|----------|----------|----------|------------------|--------------|
| Preto | 0 | 0 | 0 | $\times 10^0$ | |
| Marrom | 1 | 1 | 1 | $\times 10^1$ | $\pm 1\%$ |
| Vermelho | 2 | 2 | 2 | $\times 10^2$ | $\pm 2\%$ |
| Laranja | 3 | 3 | 3 | $\times 10^3$ | |
| Amarelo | 4 | 4 | 4 | $\times 10^4$ | |
| Verde | 5 | 5 | 5 | $\times 10^5$ | $\pm 0,5\%$ |
| Azul | 6 | 6 | 6 | $\times 10^6$ | $\pm 0,25\%$ |
| Violeta | 7 | 7 | 7 | $\times 10^7$ | $\pm 0,1\%$ |
| Cinza | 8 | 8 | 8 | $\times 10^8$ | $\pm 0,05\%$ |
| Branco | 9 | 9 | 9 | $\times 10^9$ | |
| Dourado | | | | $\times 10^{-1}$ | $\pm 5\%$ |
| Prata | | | | $\times 10^{-2}$ | $\pm 10\%$ |
| Nenhuma | | | | $\pm 20\%$ | $\pm 20\%$ |

Adaptado de: Mcroberts (2011, p. 48) e de Pakéquis⁹

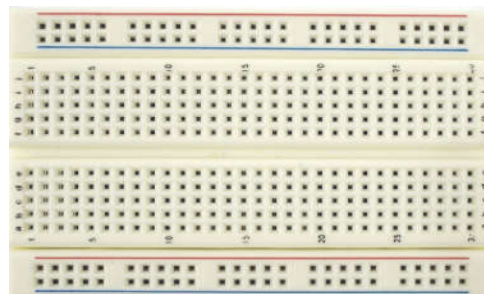
Assim, para um resistor de 220Ω é necessário um 2 na primeira faixa, que é vermelho, seguido por um 2 na faixa seguinte, também vermelho. A

⁹ Disponível em: <https://www.pakequis.com.br/p/calculadora-de-codigo-de-cores-de.html>. Acesso em: 05 abr. 2019.

terceira faixa se refere ao expoente da potência de 10 que deve ser multiplicada ao valor já computado nas duas primeiras faixas e que no caso é 10^1 . Ou seja, o expoente é 1, resultando em marrom na terceira faixa. A faixa final indica a tolerância do resistor. Caso o resistor tenha uma faixa dourada, ele tem uma tolerância de $\pm 5\%$ e isso significa que o valor, de fato, do resistor varia entre 209Ω e 231Ω . Dessa forma, um LED que requer 3 V de tensão e 20 mA de corrente, necessitará de um resistor com uma combinação de faixas vermelho, vermelho e marrom (MCROBERTS, 2011).

Para conectar um LED a um dos pinos digitais, será necessária também uma protoboard (Figura 4.12). Ela é uma placa muito usada para montagem de circuitos experimentais, pois é reutilizável e não necessita de realização de solda elétrica. O tipo mais comum e usado na maioria dos projetos é a que mede 16,5 cm por 5,5 cm e apresenta 840 furos (ou pontos) na placa. Os furos são conectados eletricamente por tiras de metal condutor alojados no interior da placa. Os furos das duas linhas de tiras paralelas ao topo e a base estão conectados eletricamente somente aos furos da mesma linha, e são projetados para carregar o barramento de alimentação e o barramento do terra. As tiras no centro correm a 90 graus dos barramentos de alimentação e do terra, e há um espaço vazio no meio para que se possa colocar circuitos integrados, de modo que cada pino do chip vá para um conjunto diferente de furos e, portanto, para um barramento diferente (MCROBERTS, 2011).

Figura 4.12 – Protoboard



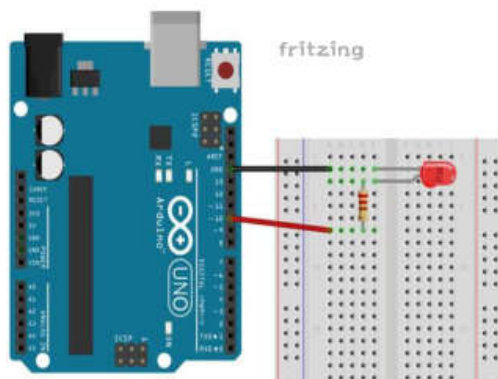
Fonte: Wikipédia¹⁰.

¹⁰ Disponível em: https://upload.wikimedia.org/wikipedia/commons/b/be/Protoboard_Unitec.jpg. Acesso em: 20 abr. 2019.

As conexões elétricas entre os dispositivos em geral são feitas com fios jumper. Eles são encontrados comercialmente com pontas moldadas para facilitar sua inserção na protoboard. Porém, é possível usar também pedaços de fios rígidos de núcleo único em que se retira cerca de 6 mm do material isolante de cada ponta (MCROBERTS, 2011).

A Figura 4.13 mostra o diagrama elétrico para acionar o LED. O jumper preto conecta o cátodo do LED ao pino GND. O ânodo do LED é conectado com o resistor de 220 Ω e este com o pino 10 do Arduino pelo jumper vermelho.

Figura 4.13 – Esquema de ligação de um LED externo ao Arduino



Adaptado de: Como Fazer as Coisas¹¹.

Com as conexões elétricas concluídas, é necessário desenvolver o algoritmo no Arduino. Como exemplo, um código para fazer o LED conectado ao pino digital 10, repetidamente, ficar aceso por um segundo e em seguida ficar apagado por um segundo é apresentado abaixo:

```
int pinoLED 10

void setup() {
  pinMode(pinoLED, OUTPUT);
}
```

¹¹ Disponível em: <http://www.comofazerascosas.com.br/projeto-arduino-pisca-led.html>. Acesso em: 21 abr. 2019.

```
void loop() {  
  digitalWrite(pinoLED, HIGH);  
  delay(1000);  
  digitalWrite(pinoLED, LOW);  
  delay(1000);  
}
```

Esse código tem uma única diferença em relação ao código para controle do *LED built-in*: a numeração da porta. E poderia até ser igual se o circuito fosse conectado ao pino do *LED built-in*. A reutilização do código ou de parte dele é uma das coisas mais úteis na programação do Arduino.

4.5.3 Eletroímã

Em abril de 1820, o cientista dinamarquês Hans Christian Orsted descobriu que se um fio condutor percorrido por uma corrente elétrica fosse colocado próximo a uma bússola, a sua agulha era defletida da posição de equilíbrio. Verificou assim que uma corrente elétrica ao atravessar um fio condutor cria ao seu redor um campo magnético (UNESP, 2019).

Enrolando-se um fio condutor de tal modo que forme uma sequência de voltas (espiras) muito próximas em forma de um tubo retilíneo produz-se uma bobina de fios chamado solenóide. Se por ele passar uma corrente elétrica, é gerado um campo magnético no sentido perpendicular à seção reta do solenóide. O resultado final é que o solenóide possui pólos norte e sul, tal como um ímã natural (UNESP, 2019).

O ferro doce (ferro puro) tem a propriedade de só se imantar enquanto estiver próximo de um campo magnético. Assim, usando um núcleo de ferro doce em um solenóide é criado um ímã temporário relativamente forte chamado eletroímã. Os eletroímãs apresentam inúmeras aplicações práticas tais como telégrafos, fones, alto falantes, telefones, campainhas e relés entre outras (USP, 2019), mais aqui será dada atenção a apenas duas aplicações: o eletroímã solenóide de tracionamento e o relé eletromecânico.

O Eletroímã Solenóide de tracionamento ou de levantamento é um dispositivo capaz de utilizar corrente elétrica para gerar um campo magnético e atrair materiais magnéticos próximos a ele e controlar a movimentação e transporte do material. Possui uma larga faixa de capacidade de tração, existindo desde aqueles com capacidade de levantar uns poucos newtons até os usados para erguer cargas com várias toneladas. É possível construir um para controlar cargas pequenas, mas é muito mais simples, por exemplo, entrar com “Eletroímã Solenoide 2,5kg” ou “Ímã Elétrico 12v Dc” em um buscador de páginas da internet e se obter uma lista de sites que possibilitam a aquisição de um deles.

Já um relé do tipo eletromecânico (EMR – *Electromechanical Relay*) é simplesmente um interruptor eletromecânico. Como mostra a Figura 4.14, ele é composto por um eletroímã, uma armadura (haste de material ferromagnético) e pelos contatos: comum (C), normalmente aberto (NA) e normalmente fechado (NF).

Figura 4.14 – Esquema de um relé EMR simples

| Situação do relé | Imagem | Esquema elétrico |
|------------------|--------|------------------|
| Desligado | | |
| Acionado | | |

Fonte: <http://autocorerobotica.blog.br/controlando-lampadas-por-control-remoto-com-arduino/>.
Acesso em: 10 maio 2019.

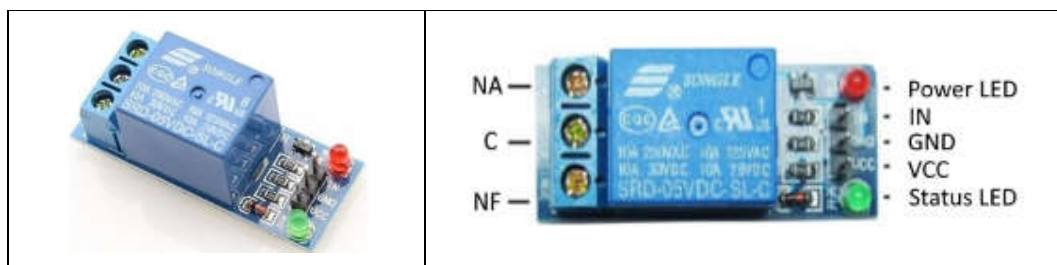
4.5.4 Controle de acionamento de um módulo relé

Quando for desejado acionar um dispositivo que utilize um valor de tensão superior aos 5 V ou correntes maiores que 40 mA fornecidos pela placa do Arduino Uno, é necessário utilizar alguma interface que suporte tal carga. Um relé seria uma opção, mas quando se liga e desliga uma bobina, ela induz correntes de valores bem superiores aos limites suportados pelo Arduino e para evitar isso é necessário o uso de circuito de proteção. Esse circuito, não é trivial de ser construído, mas pode ser encontrado pronto num módulo relé, que é um relé montado em uma placa junto com o circuito auxiliar para seu acionamento, e por isso é frequentemente usado com o Arduino (EMBARCADOS LÂMPADA, 2019).

Num módulo relé, além dos pinos C, NA e NF, existem os pinos VCC, GND e IN. Os pinos VCC e GND devem ser conectados aos pinos +5 V e GND do Arduino, respectivamente. Já o pino IN deve ser conectado à porta digital do Arduino responsável pelo controle de acionamento (VIDADESILÍCIO RELÉ, 2019).

Atualmente, são fabricados vários modelos de módulos relés com diversas quantidades de relés, cada um deles podendo controlar um circuito diferente dos outros. Cada circuito controlado por um relé é chamado canal. Comercialmente, encontram-se placas com 1 (Figura 4.15), 2, 4, 8 e até 16 canais (VIDADESILÍCIO RELÉ, 2019).

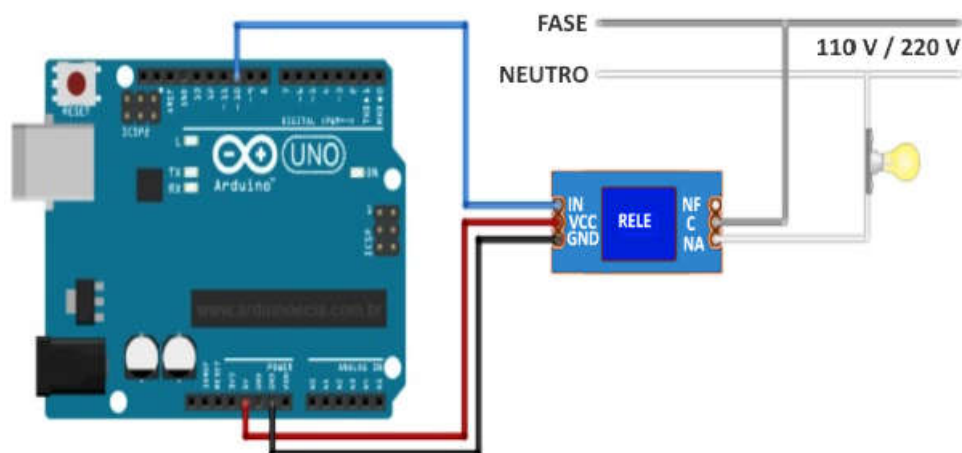
Figura 4.15 – Módulo relé de 1 canal



Fonte: <http://mekhos.com.br/mekhos/produto/modulo-rele-de-1-canal/>. Acesso em: 10 maio 2019.

Como exemplo, a Figura 4.16 mostra um esquema elétrico em que o pino 10 do Arduino é conectado ao pino IN do módulo relé para controlá-lo. A lâmpada tem um de seus terminais conectado ao contato NA do relé e o outro ao neutro da rede. Finalmente, o contato C do relé é conectado ao fase da rede. Desta forma, se for usado o algoritmo desenvolvido para o LED externo piscar, a lâmpada ficará acendendo e apagando com intervalo de um segundo.

Figura 4.16 – Acionamento de uma lâmpada por um módulo relé.



Fonte: Arduino e Cia¹².

4.5.5 Controle da recepção de sinal do sensor de toque

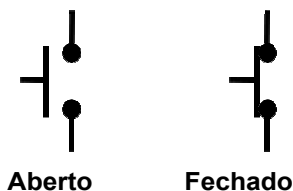
Roggia (2016, p. 23) define um sensor como um dispositivo sensível a uma forma de energia do ambiente (energia luminosa, cinética, sonora, térmica, entre outras), relacionando informações sobre uma grandeza física que precisa ser medida como luminosidade, temperatura, pressão, vazão, posição e corrente. E de acordo com a natureza do sinal que geram, os sensores podem ser classificados em digitais (discretos) e analógicos (contínuos).

¹² Disponível em: <https://www.arduinoocia.com.br/ligando-uma-lampada-com-modulo-rele-arduino/>. Acesso em: 20 maio 2019.

Ainda segundo Roggia (2016, p. 23), os sensores digitais são utilizados para monitorar a ocorrência ou não de um determinado evento. Apresentam como sinal apenas dois estados: O primeiro, quando a corrente pode fluir por ele, é chamado de fechado ou ligado (ON) ou alto (HIGH) ou nível 1; O segundo, quando não há corrente passando por ele, é chamado de aberto ou desligado (OFF) ou nível baixo (LOW) ou nível 0.

Os sensores digitais são de ativação/desativação (ON/OFF) e por isso são chamados comutadores, pois comutam entre os estados aberto (desligado) e fechado (ligado), como mostrado na Figura 4.17.

Figura 4.17 – Estados de um comutador elétrico.



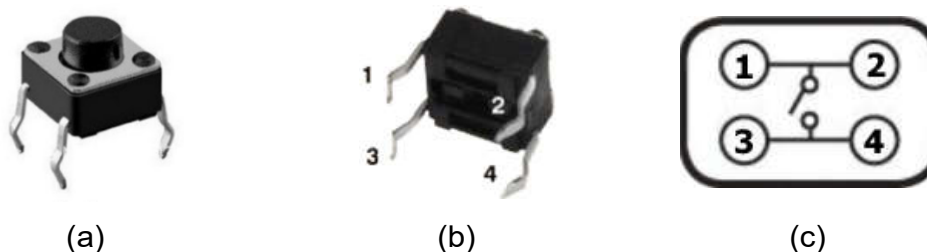
Adaptado de: Eletrônica para Artistas¹³

O comutador alternado é o tipo mais usado, mudando de um estado para outro apenas quando acionado. O Interruptor elétrico e a chave liga/desliga são seus exemplos mais comuns. Já os comutadores do tipo momentâneo, quando pressionados passam do estado aberto para o fechado e retornam ao estado anterior quando a pressão termina. Seus exemplos mais comuns são o botão da campainha elétrica e o botão de pressão ou *push button* (BANZI, 2011, p. 40).

O botão de pressão também é conhecido como botão momentâneo, botão tátil ou botão de apertar. O tipo mais comum tem quatro pinos e seu aspecto é mostrado nas Figura 4.18 (a) e Figura 4.18 (b) e seu símbolo elétrico é visto na Figura 4.18 (c).

¹³ Disponível em: <http://eletronicaparaartistas.com.br/arduino-5-resistores-de-pull-up/>. Acesso em: 11 jul. 2019.

Figura 4.18 – Aspecto e símbolo do botão de pressão de quatro pinos.



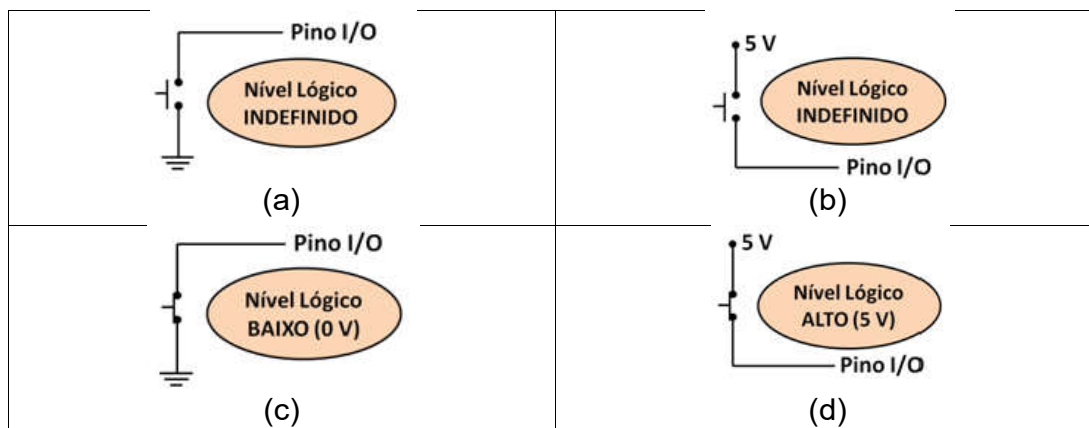
Adaptado de: Gueddes (2017, p. 44) e Kamei (2019, p. 7).

Com o botão não pressionado, só os pinos 1 e 2 estão em contato elétrico entre si, assim como os pinos 3 e 4. Os quatro pinos ficarão conectados eletricamente enquanto o botão for pressionado, (Kamei 2019, p. 8).

Um sensor digital é comumente conectado ao Arduino em um pino digital configurado como de entrada de dados. Nesse modo o pino é posto em um estado de alta impedância, equivalente a um resistor de 100 MΩ em série com o circuito a ser monitorado e consumindo corrente extremamente pequena. Isso significa que é necessária uma corrente muito baixa para mudar seu estado lógico de um valor para outro. E significa também que um pino configurado como entrada sem nada conectado a ele, ou com fios conectados a ele que não estão conectados a outros circuitos, estará em um estado lógico indefinido e seu nível de tensão apresentará alterações aparentemente aleatórias, captando ruído elétrico do ambiente ou acoplamento capacitivo do estado de um pino próximo (ARDUINO PINOS DIGITAIS, 2019).

Uma tentativa não eficiente de evitar que o estado do pino seja indefinido é mostrada na Figura 4.19 (a) e na Figura 4.19 (b), em que o botão não está pressionado e o pino encontra-se com estado indefinido. Com o botão pressionado, o estado do pino passa a ser LOW ou HIGH como é visto, respectivamente, na Figura 4.19 (c) e Figura 4.19 (d).

Figura 4.19 – Controle do nível lógico com botão de pressão.

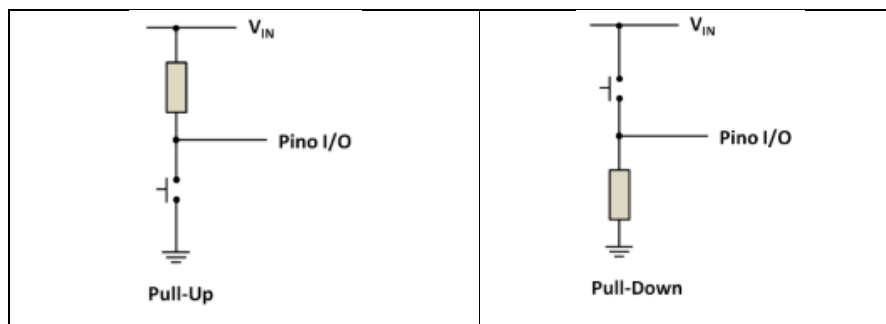


Adaptado de: <http://eletronicaparaartistas.com.br/arduino-5-resistores-de-pull-up/>.
Acesso em: 11 jul. 2019.

As conexões com resistores *pull-up* e *pull-down* mostradas na Figura 4.20 são as soluções mais simples para o pino nunca ficar em um estado indefinido.

Em ambas é usado um resistor geralmente de 10 K Ω conectado a um botão de pressão. Na ligação do tipo *pull-up*, enquanto o botão está livre o estado lógico do pino é HIGH e enquanto o botão está pressionado o pino está no estado LOW. Já na conexão *pull-down* ocorre o contrário, com o pino digital no estado LOW com o botão livre e HIGH com o botão pressionado (AUTOCOREROBOTICA, 2019).

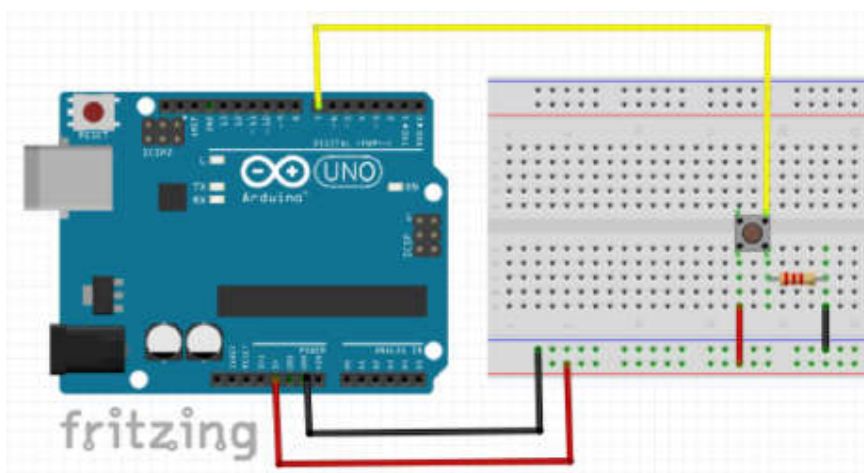
Figura 4.20 – Resistor de pull-up e resistor de pull-down.



Adaptado de: <https://www.embarcados.com.br/pinos-digitais-do-arduino/>.
Acesso em: 15 jul. 2019.

Em Arduino Button (2019) é apresentado um projeto que usa uma ligação *pull-down* conectada ao pino digital 7 para controlar o *LED built-in* e a Figura 4.21 mostra o esquema elétrico do circuito.

Figura 4.21 – Botão pull-down para controlar acionamento do *LED built-in*.



Adaptado de: ARDUINO BUTTON (2019).

Num *sketch* do Arduino, se for desejado que o pino digital de número `val_pino` seja usado para fazer a entrada de dados, ele deve ser previamente configurado no `setup` com a instrução `pinMode(val_pino, INPUT)`, e para ler seu valor lógico deve ser usada a instrução `digitalRead(val_pino)`. O retorno dessa instrução será HIGH ou LOW. Em Arduino Digital Read (2019) é encontrado um exemplo de código para controle do circuito mostrado na Figura 4.21:

```
int ledPin = 13; // Atribui a variável ledPin o valor do pino do LED built-in.
int inPin = 7;   // Atribui a variável inteira inPin o valor do pino digital 7.
int val = 0;    // Atribui a variável inteira val o valor 0.

void setup() {
  pinMode(ledPin, OUTPUT); // Configura o pino 13 como saída de dados
  pinMode(inPin, INPUT);  // Configura o pino 7 como entrada de dados
}

void loop() {
  val = digitalRead(inPin); // Lê o pino de entrada
  digitalWrite(ledPin, val); // Aciona o LED com o valor lido do botão
}
```

Este programa é bem simples: inicialmente são criadas as variáveis inteiras **ledPin** com valor 13, **inPin** com valor 7 e **val** com valor 0. Dentro do setup do sketch, **ledPin** é configurada para saída de dados e **inPin** para entrada de dados. Finalmente, na primeira instrução do loop, **val** recebe constantemente o valor do estado de **inPin** e na segunda instrução este valor é colocado em **ledPin**.

Pode-se escrever uma versão simplificada para esse algoritmo, sem uso de variáveis. Nessa versão, o valor do pino 7 é colocado diretamente como valor do pino 13.

```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(7, INPUT);  
}  
  
void loop() {  
  digitalWrite(13, digitalRead(7));  
}
```

É importante observar que em um *sketch* mais extenso, em que varias portas são usadas em muitos comandos, é muito comum a necessidade de troca de uma porta ou de varias delas e o uso de variáveis torna muito mais simples a realização dessas mudanças.

Em Arduino Button (2019) e Arduino Pushbutton (2019) encontram-se outras versões para este algoritmo que usam um teste lógico para saber se o botão está pressionado e acender o LED caso o resultado seja verdadeiro com nível lógico HIGH, ou apagar o LED em caso contrário, com nível lógico LOW. A instrução **if** serve para verificar se uma condição é verdadeira ou falsa e, segundo Arduino IF (2019), sua sintaxe é:

```
if (condição) [bloco de instruções];
```

A condição deve sempre estar entre parêntesis. Ela é um teste lógico que só possui dois resultados possíveis: **true** (verdadeiro) ou **false** (falso). O bloco de instruções dentro dos colchetes somente será executado se o resultado da condição for **true**.

Caso seja necessário executar alguma instrução somente se o resultado do teste for falso, deve-se, segundo Arduino Else (2019), usar:

```
if (condição) [bloco de instruções]
else [bloco de instruções];
```

O uso de **if..else** traz maior controle sobre o fluxo do código quando comparado com a instrução **if** sozinha. Permite também que vários testes sejam agrupados. O **else** pode realizar outro teste **if**, criando a possibilidade de executar simultaneamente vários testes mutuamente exclusivos.

Assim, uma versão reduzida do *sketch* com uso de teste lógico, encontrado em Arduino Button (2019), pode ser

```
void setup() {
  pinMode(13, OUTPUT);
  pinMode(7, INPUT);
}

void loop() {
  if (digitalRead(7) == HIGH) [digitalWrite(13, HIGH)]
  else [digitalWrite(13, LOW)];
}
```

Em um teste lógico de igualdade é necessário usar dois sinais seguidos de igual, pois na linguagem do Arduino um único sinal de igual é usado para realizar atribuição de variáveis (ARDUINO IF, 2019).

É fácil perceber que no projeto que usa uma ligação *pull-down* para controlar o *LED built-in* o teste condicional além de ser opcional, traz uma

complexidade desnecessária. Porém, existem incontáveis situações em que seu uso é indispensável.

4.5.6 Controle da recepção de sinal do sensor de luz

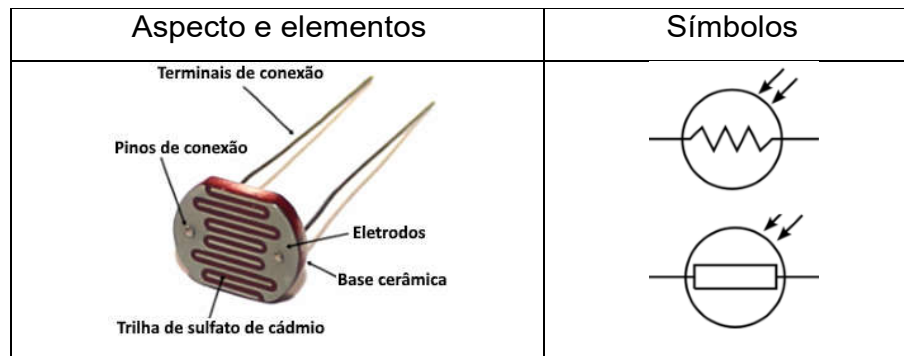
De acordo com Roggia (2016, p. 23), além do sensor digital, existe o sensor analógico que serve para medir uma grandeza analógica, ou seja, a grandeza física com valor compreendido entre um mínimo e um máximo de uma faixa contínua de valores. O sensor digital apresenta em sua saída um sinal de tensão, corrente ou resistência proporcional ao valor da grandeza física monitorada.

Existem diversos tipos de sensores analógicos e entre eles, Braga (2005, p. 106) define sensores resistivos como transdutores que apresentam a característica de mudar sua resistência elétrica na presença de um sinal de energia qualquer, normalmente luz, temperatura ou pressão e convertendo-o em um sinal elétrico analógico.

Segundo Braga (2005, p.107), o tipo mais comum de sensor resistivo de luz é o LDR (Ligh Dependent Resistor), também conhecido como fotorresistor ou ainda célula de sulfeto de cádmio (CdS). Os LDRs têm diferentes formas, tamanhos e valores de mínima e máxima resistência. Têm o aspecto mais comum e símbolos mostrados na Figura 4.22.

Um LDR é constituído por uma base de cerâmica isolante sobre a superfície da qual se encontram duas placas condutoras separadas por uma fina trilha ondulada de sulfeto de cádmio. Cada placa está conectada a um dos dois eletrodos que constituem os terminais de conexão do LDR.

Figura 4.22 – Aspecto, elementos e símbolos de um LDR.



Adaptado de: Bóson Treinamentos¹⁴ e Proesi¹⁵.

O sulfeto de cádmio apresenta resistividade extremamente alta na escuridão total, levando a resistência do LDR a ter de milhões de ohms, mas como essa substância é sensível a todo espectro de luz visível, o aumento da luminosidade incidente sobre ela produz a liberação cada vez maior de grandes quantidades de portadores de cargas elétricas, diminuindo drasticamente sua resistividade, o que faz a resistência do LDR atingir algumas dezenas de ohms. Assim, ele permite detectar se o ambiente está claro, escuro ou com qualquer valor de luminosidade. É importante observar que o LDR, ao contrário do LED, não tem polaridade (BRAGA, 2005, p. 107).

Como a resistência do LDR é uma grandeza analógica, para medi-la com o Arduino é necessário usar uma porta analógica. Diferentemente de um pino digital, não é necessário definir um pino analógico como de entrada ou saída e para ler o seu valor é utilizado o comando **analogRead(num_pino)** em que num_pino é um dos seis pinos analógicos, ou seja, A0, A1, A2, A3, A4 ou A5. Cada pino analógico tem um conversor analógico-digital de 10 bits, o que corresponde a 2^{10} valores inteiros. Assim, um pino analógico ao receber

¹⁴ Disponível em: <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/curso-de-eletronica-o-que-e-um-ldr-light-dependent-resistor/>. Acesso em: 21 jul. 2019.

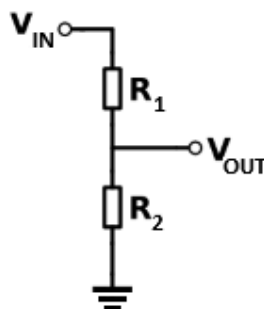
¹⁵ Disponível em: <https://proesi.com.br/sensor-ldr-10mm.html>. Acesso em: 21 jul. 2019.

voltagens entre 0 e 5 volts, converte-as em valores inteiros entre 0 (0 V) e 1023 (5 V) (MCROBERTS, 2011, p. 76).

Embora a principal função dos pinos analógicos nas aplicações em geral seja a leitura de sensores analógicos, eles também possuem toda a funcionalidade de pinos de entrada/saída de uso geral (GPIO), ou dos pinos digitais de 0 a 13. Assim, caso seja preciso de mais pinos digitais para uso geral, e nenhum pino analógico estiver em uso, os pinos analógicos poderão ser usados como se fossem GPIO (ARDUINO PINOS ANALOGICOS, 2019).

Praticamente todos os circuitos sensores usam de algum modo, um divisor de tensão ou divisor de potencial. Um divisor de tensão padrão utiliza dois resistores ligados em série, como mostra a Figura 4.23 (NETTO, 2019, p. 52).

Figura 4.23 – Circuito do divisor de tensão.



Adaptado de: <http://nerdeletrico.blogspot.com/2011/04/circuito-divisor-de-tensao.html>.

Acesso em: 24 jul. 2019.

Sendo V_1 e V_2 as tensões em R_1 e R_2 , a voltagem de entrada V_{IN} será reduzida (dividida) para uma tensão de saída V_{OUT} , assim:

$$V_{IN} = V_1 + V_2 \quad (4.3)$$

Se i_1 e i_2 são as correntes em R_1 e R_2 , é fácil perceber que $V_1 = i_1 \cdot R_1$ e $V_{OUT} = V_2 = i_2 \cdot R_2$ e admitindo que $i_{IN} = i_1 \approx i_2$, ou seja, que pouca corrente é drenada entre R_1 e R_2 , ou que $i_{OUT} \ll i_{IN}$, encontra-se:

$$V_{IN} = (R_1 + R_2) \cdot i_{IN} \quad (4.4)$$

Mas como:

$$i_{IN} \approx i_2 = \frac{V_2}{R_2} = \frac{V_{OUT}}{R_2} \quad (4.5)$$

Então:

$$V_{IN} = (R_1 + R_2) \cdot \frac{V_{OUT}}{R_2} \quad (4.6)$$

Assim:

$$V_{OUT} = \frac{R_2}{R_1 + R_2} V_{IN} \quad (4.7)$$

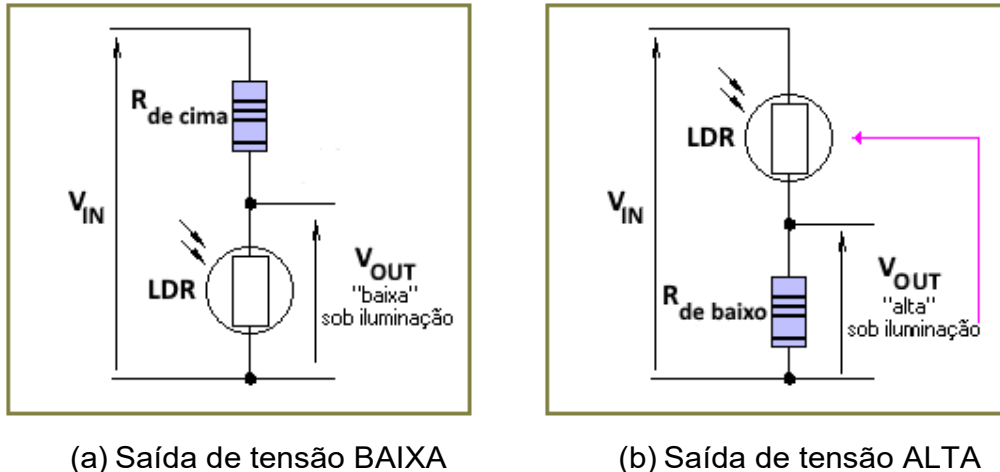
Logo:

$$V_{OUT} = \frac{V_{IN}}{1 + R_1/R_2} \quad (4.8)$$

Mostrando que é a razão entre os valores dos resistores e não os seus valores em si que importam para determinar V_{OUT} (MCROBERTS, 2011, p. 118).

Substituindo R_1 ou R_2 por um LDR no divisor de tensão, pode-se obter duas situações, como mostra a Figura 4.24. Na situação (a), obtém-se na saída uma tensão BAIXA quando o LDR está intensamente iluminado e uma tensão ALTA quando o LDR é posto no escuro. Já na situação (b), obtém-se o contrário, ou seja, uma tensão ALTA quando o LDR está intensamente iluminado e uma tensão baixa quando o LDR é posto no escuro. A escolha de uma delas só depende da leitura desejada no sensor (NETTO, 2019, p. 53-54).

Figura 4.24 – Divisor de tensão com LDR.



Adaptado de: Netto 2019, p. 53-54.

Como exemplo, a Tabela 4.1 mostra as voltagens obtidas conforme a luminosidade sobre o LDR se altera no modo de tensão de saída baixa, usando $V_{IN} = 5\text{ V}$ e um LDR com resistência de $100\text{ k}\Omega$ no escuro e $10\text{ k}\Omega$ quando bem iluminado.

Tabela 4.1 – Valores de V_{OUT} para um LDR com $V_{in} = 5\text{ V}$.

| R_1 | R_2 (LDR) | V_{OUT} | Luminosidade |
|---------------------|----------------------|-----------|--------------|
| $10\text{ k}\Omega$ | $100\text{ k}\Omega$ | 4,54 V | Mais escuro |
| $10\text{ k}\Omega$ | $73\text{ k}\Omega$ | 4,39 V | 25% |
| $10\text{ k}\Omega$ | $45\text{ k}\Omega$ | 4,09 V | 50% |
| $10\text{ k}\Omega$ | $28\text{ k}\Omega$ | 3,68 V | 75% |
| $10\text{ k}\Omega$ | $10\text{ k}\Omega$ | 2,5 V | Mais claro |

Fonte: Mcroberts, 2011, p. 119.

Em Vidadesilício LDR (2019) é apresentado um projeto simples capaz de detectar a luminosidade e acender o LED em um ambiente escuro, e o apagar em um ambiente claro. Como mostra a Figura 4.25, o LED está conectado com a porta digital 7 e em série com o um resistor de $300\ \Omega$. O divisor está conectado com a porta analógica A0, tem saída de tensão baixa sob iluminação e possui além do LDR, um resistor de $10\text{ k}\Omega$. Também foi definido que, uma leitura da porta analógica entre 0 e 500, indica muita luminosidade ambiente e o LED deve

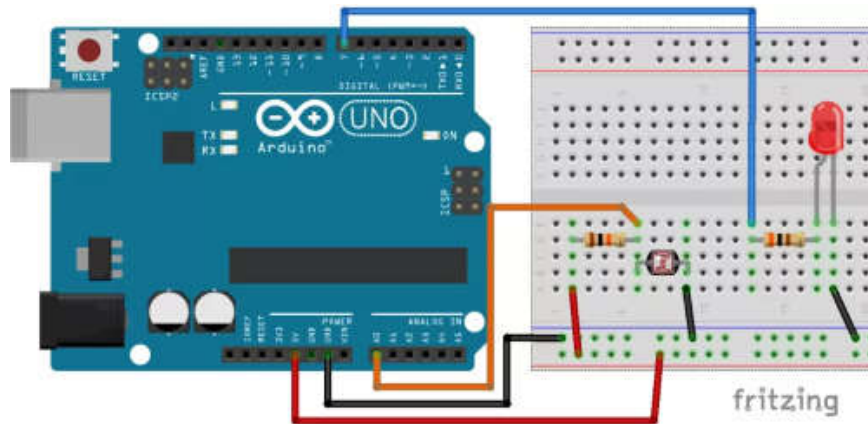
ser apagado. Já uma leitura entre 501 e 1023, significa pouca luminosidade e o LED deve ser aceso. O *sketch* do projeto é:

```
int ldrValor = 0; //Valor lido do LDR

void setup() {
  pinMode(7,OUTPUT); //define a porta 7 como saída
}

void loop() {
  ldrValor = analogRead(A5); //Lê valor do LDR: valor entre 0 e 1023
  if (ldrValor > 500) {
    digitalWrite(ledPin,HIGH); //Liga o LED se o valor > 500
  }
  else {
    digitalWrite(ledPin,LOW); // senão, apaga o LED
  }
  delay(100);
}
```

Figura 4.25 – Controle do acionamento de um LED com um LDR.



Adaptado de: <https://portal.vidadesilicio.com.br/sensor-de-luz-com-ldr/>.
Acesso em: 25 jul 2019.

Nesse algoritmo, primeiro é criada a variável inteira **ldrValor** para armazenar o valor da leitura do LDR e com valor inicial nulo. A seguir, dentro do *setup*, a porta digital 7 é configurada para saída de dados. Depois, dentro do *loop*, a primeira instrução faz a leitura da porta analógica e armazena seu valor em **ldrValor**. A instrução seguinte realiza o teste lógico: Se o valor de **ldrValor** for maior que 500 o LED será aceso, caso contrário será apagado. Por último, ocorre uma pequena espera de 100 ms antes do reinício do *loop*.

4.5.7 Transmissão via porta serial

Toda placa Arduino possui pelo menos uma porta serial, também chamada UART¹⁶ ou USART¹⁷. Essa porta serve para fazer a comunicação entre o Arduino e um computador através da conexão USB. Além disso, como está conectada aos pinos RX (de recepção) e TX (de transmissão), usa níveis lógicos TTL¹⁸ para fazer comunicação com outros dispositivos. Na placa Uno, o RX corresponde ao pino 0 e o TX ao pino 1. Como o upload do computador para o Arduino também é realizado pela conexão USB, durante sua realização nenhum outro dispositivo deve estar conectado ao RX ou ao TX, pois pode ocorrer interferência nessa comunicação, causando falhas no upload (ARDUINO SERIAL, 2019).

A plataforma Arduino possui diversas funções para manipulação da comunicação serial. Segundo Embarcados Serial (2019) as principais são: **Serial.begin(taxa)**, usada para iniciar a porta serial e definir a taxa de velocidade para transmissão de dados seriais em bits por segundo; **Serial.available()**, usada para obter o número de bytes (caracteres) disponíveis para leitura na porta serial; **Serial.read()**, que permite ler o byte mais recente apontado no buffer de entrada da serial; **Serial.print(texto)**, que permite escrever um texto na serial; e **Serial.println(texto)**, que além de escrever um texto na serial, acrescenta ao seu final o caractere de fim de linha e o caractere de nova linha.

Ainda, segundo Embarcados Serial (2019), o monitor serial Integrado a IDE do Arduino é uma das interfaces mais simples para fazer comunicação serial. Seu uso exige que o Arduino esteja devidamente conectado ao

¹⁶ Acrônimo de Universal Asynchronous Receiver/Transmitter e significando Receptor/Transmissor Universal Assíncrono.

¹⁷ Acrônimo de Universal Synchronous Asynchronous Receiver Transmitter e significando Transmissor/Receptor Universal Síncrono e Assíncrono.

¹⁸ Acrônimo de Transistor-Transistor Logic e significando Lógica Transistor-Transistor.

computador e para iniciá-lo, basta clicar no botão com ícone de lupa localizado na parte superior da janela e à direita da barra de ferramentas da IDE (Figura 4.26).

Figura 4.26 – Monitor serial integrado a IDE do Arduino.



Fonte: print screen da IDE no sistema operacional Windows 10.

Após a abertura do monitor, deve-se usar o botão de ajuste da velocidade de troca de dados, localizado na parte inferior direita da interface, para configurar a velocidade com o mesmo valor definido no *sketch* do Arduino. Para enviar dados é preciso fazer sua inserção na caixa de envio, localizada na parte superior da janela do monitor e acionar o botão enviar ou a tecla enter. Os dados de retorno da placa para o computador, caso existam, serão exibidos na caixa localizada no centro da janela do monitor.

Em *Arduino Read Serial* (2019) é encontrado um exemplo de aplicação para monitorar o estado de um botão momentâneo usando a comunicação serial. O circuito é praticamente igual ao mostrado na Figura 4.25, tendo como única diferença a substituição do pino 7 pelo pino 2.

O *sketch* sem os comentários para realizar a comunicação serial é:


```
int pushButton = 2;

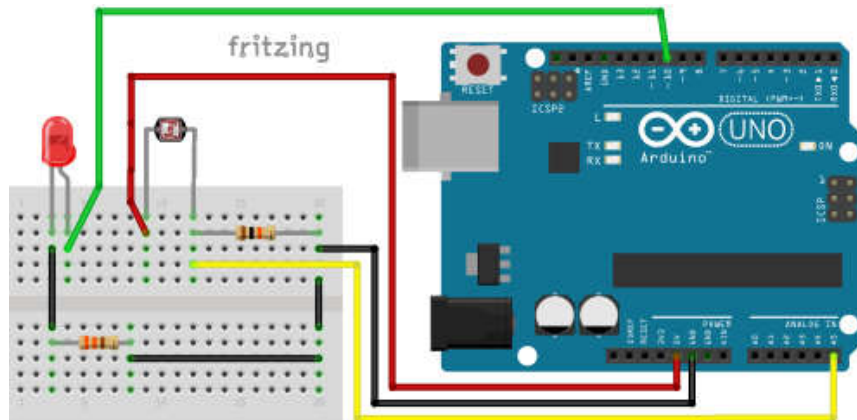
void setup() {
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}

void loop() {
  int buttonState = digitalRead(pushButton);
  Serial.println(buttonState);
  delay(1);
}
```

Inicialmente é criada a variável inteira **pushButton** com valor 2, o do pino de leitura do estado do botão. No setup, a velocidade de comunicação serial entre a placa e o computador é configurada para 9600 bits de dados por segundo. Depois o pino digital com valor definido na variável **pushButton** é configurado como uma entrada de dados. No loop, o estado do pino é lido e seu valor colocado na variável inteira **buttonState**. Em seguida, usando o comando **Serial.println**, este valor é enviado para a porta serial. Assim será escrito na caixa central do monitor serial uma sequência de "0"s, um abaixo do outro, se o botão estiver aberto ou de "1"s em caso contrário. Por último, A instrução **delay(1)** é usada para criar um atraso de um milissegundo entre as leituras e gerar estabilidade no programa.

Em outro exemplo, encontrado em Arduino LDR & LED (2019), similar ao encontrado em Vidadesilício LDR (2019) e cujo circuito elétrico é apresentado na Figura 4.27, um LDR controla o acionamento de um LED, acendendo-o quando a leitura na porta A5 é maior que 800 e apagando-o em caso contrário. Além disso, escreve na serial o valor encontrado na porta analógica.

Figura 4.27 – Circuito para monitorar o estado de um botão momentâneo.



Adaptado de Arduino LDR & LED (2019).

O seu *sketch* sem os comentários é:

```
int LedPin = 10;
int sensorPin = A5;
int sensorValor = 0;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(LedPin, OUTPUT);
}
```

```
void loop() {
  int sensorValor = analogRead(sensorPin);
  Serial.println(sensorValor);
  if (sensorValor > 800) {
    digitalWrite(LedPin, HIGH);
  }
  else {
    digitalWrite(LedPin, LOW);
  }
}
```

No algoritmo, primeiro é criada a variável inteira **LedPin**, com valor 10, o do pino de leitura do estado do botão. Depois é criada a variável inteira **sensorPin**, com valor A5, o pino analógico que será usado. Em seguida é criada a variável inteira **sensorValor** para armazenar o estado do sensor e inicialmente com valor nulo. No setup, a velocidade de comunicação serial entre a placa e o

computador é configurada para 9600 bits de dados por segundo. Depois o pino digital com valor definido na variável **sensorValor** é configurado como uma entrada de dados. No loop, o estado do pino é lido e seu valor colocado na variável **sensorValor**. Em seguida, usando o comando **Serial.println**, este valor é enviado para a porta serial. Por fim um teste é realizado e o led é aceso se o valor da leitura do sensor for maior que 800 e apagado em caso contrário.

4.5.8 Transmissão via Bluetooth

Bluetooth é uma tecnologia desenvolvida pela Ericsson em 1994. Usa ondas de rádio com frequências entre 2,4 GHz e 2,483 GHz. O objetivo inicial era fazer conexão sem fio (*wireless*) em substituição aos cabos que conectam a curta distância entre computadores e seus periféricos. Posteriormente, ganhou o suporte de empresas tais como Intel, IBM, Toshiba, Nokia, Lucent, Motorola entre outras e juntas criaram o *Bluetooth Special Interest Group* (SIG), para ser responsável por escrever as especificações do protocolo e lançar versões atualizadas e novas funcionalidades. Atualmente, suas aplicações incluem o controle de mídia (áudio, vídeo e imagens) e de sistemas de comunicações com diversos dispositivos tais como fones de ouvido, equipamentos de som, TVs, home theaters, computadores, telefones celulares, câmeras digitais, dispositivos de interface humana (mouses, joysticks, teclados), entre outras (TELECO, 2019).

Os principais requisitos que norteiam o desenvolvimento do Bluetooth são o baixo consumo de energia, o baixo custo, a capacidade de transmissão de voz, dados e sinalização e por último, uma área de cobertura pequena, entre um e cem metros, sendo por isso classificada como rede sem fio PAN (*Personal Area Network*), que não precisa de autorização governamental para ser utilizada (TELECO, 2019).

Ainda, segundo Teleco (2019), uma rede Bluetooth é chamada *piconet*. Ela é composta por um dispositivo mestre e até sete outros dispositivos,

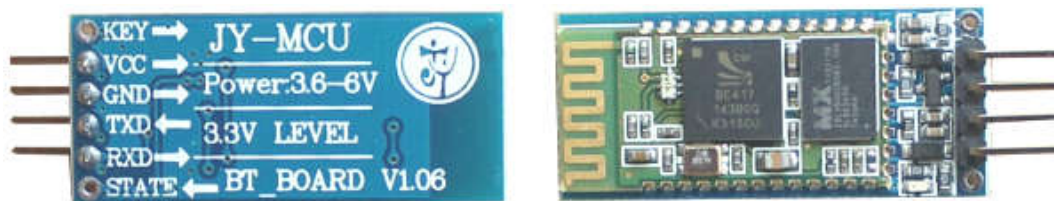
chamados escravos. O mestre pode solicitar e aceitar conexão (pareamento), enquanto o escravo só pode aceitar. Assim, toda comunicação ocorre somente entre o mestre e os escravos, não existindo comunicação direta entre dispositivos escravos.

Devido à facilidade de uso e preço relativamente baixo, o módulo Bluetooth HC-06 é um dos mais usados com Arduino. O HC-06 apresenta transmissão full duplex, ou seja, pode transmitir e receber ao mesmo tempo. Trabalha apenas em modo escravo com versão 2.0+EDR e o perfil SPP (*Serial Port Profile*). Tem potência de 2,5 mW e alcance aproximado de 10 metros (ELETROGATE, 2019).

Como mostra a Figura 4.28, ele apresenta quatro pinos:

- VCC que deve receber tensão de alimentação entre 3,6 V e 6,0 V;
- GND ou pino terra;
- TXD que envia sinais de transmissão com nível de tensão 3,3 V;
- RXD que recebe sinais de transmissão com nível de tensão de 3,3 V.

Figura 4.28 – Módulo HC-06.



Adaptado de: https://www.fecegypt.com/uploads/dataSheet/1522750308_bluetooth%20module%20hc06.pdf, p. 1. Acesso em: 28 jul. 2019.

O Arduino recebe sem problemas um sinal de 3,3 V vindo do pino TXD como nível lógico alto. Já no sentido de comunicação oposto, é necessário converter os 5 V saídos do pino TX do Arduino nos 3,3 V suportados pelo pino RXD. Para fazer essa redução a maneira mais simples e barata é usar um divisor de tensão em que, conforme a eq. 4.8, a resistência R_2 seja o dobro de

R_1 , pois essa é a razão aproximada entre a tensão de 3,3 V no pino RXD e a queda de tensão entre os pinos do Arduino e do HC-06:

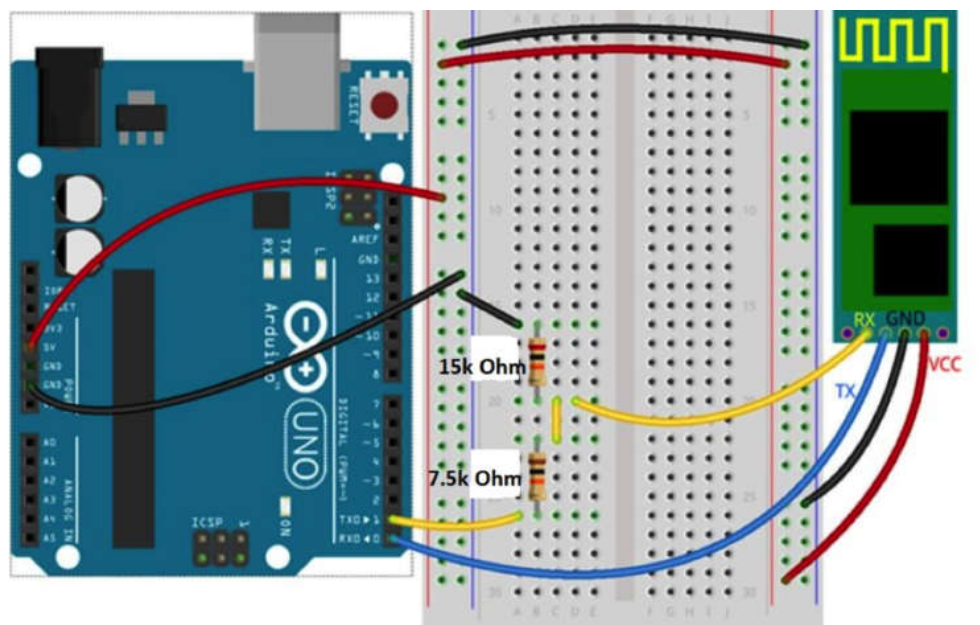
$$\frac{V_{out}}{V_{in} - V_{out}} = \frac{\frac{V_{in}}{1 + R_1/R_2}}{V_{in} - \frac{V_{in}}{1 + R_1/R_2}} = \frac{R_2}{R_1} \quad (4.9)$$

$$\frac{V_{out}}{V_{in} - V_{out}} = \frac{3,3 \text{ V}}{5 \text{ V} - 3,3 \text{ V}} = \frac{3,3}{1,7} \approx \frac{2}{1} \quad (4.10)$$

Por isso são usados para R_1 e R_2 , respectivamente, resistores de 1 k Ω e 2,2 k Ω em Eletrogate (2019), ou de 7,5 k Ω e 15 k Ω em Wustl (2019), ou ainda de 10 k Ω e 20 k Ω em Engineers (2019).

A Figura 4.29 mostra um circuito que usa o HC-06 como interface de conexão Bluetooth para enviar comandos de um smartphone para o Arduino ativar e desativar o LED *on-board* em sua porta digital 13.

Figura 4.29 – Circuito para acionamento da placa HC-06.



Fonte: [https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_\(HC-06\)_%2B_Arduino](https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_(HC-06)_%2B_Arduino). Acesso em: 30. Jul. 2019.

Em Filipeflop (2019) encontra-se o *sketch* do projeto:

```
char buf;

void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  while(Serial.available() > 0) {
    buf = Serial.read();
    if (buf == 'L') {
      digitalWrite(13, HIGH);
    }
    if (buf == 'D') {
      digitalWrite(13, LOW);
    }
  }
}
```

A primeira linha do algoritmo cria a variável **buf** que irá armazenar os valores lidos na porta serial. No setup, o pino 13 é configurado para saída e a taxa de comunicação da serial estipulada em 9600. No loop, a primitiva **while** verifica se existe algum valor a ser lido na porta serial e se existir coloca este valor na variável **buf**. Em seguida, no primeiro **if**, verifica se ela é igual a 'L', colocando o pino 13 em HIGH e acendendo o **LED on-board**. Já o segundo **if** verifica se **buf** tem valor 'D', colocando o pino 13 em LOW e desligando o LED em caso verdadeiro.

Como no Arduino Uno as comunicações com o computador e com o módulo Bluetooth utilizam os mesmos pinos, somente após o final do upload do *sketch* é que devem ser conectados os pinos TX e RX do Arduino aos pinos RX e TX do HC-06. Para evitar isso, pode-se incluir no *sketch* uma chamada a biblioteca externa **SoftwareSerial**, nativa da IDE Arduino, para simular por software uma porta serial em dois outros pinos do Arduino e manter livre sua serial em hardware para ser utilizada no debug e gravação do código a ser executado (VIDADESILÍCIO HC06, 2020).

Quando o HC-06 está energizado, seu LED pisca repetidamente, indicando que o mesmo está aguardando conexão com um dispositivo mestre. Neste momento, utilizando algum aplicativo para gerenciamento de conexão Bluetooth instalado num celular, ao acessar a lista de varredura de dispositivos Bluetooth no aplicativo, deve-se encontrar o nome HC-06. Ao selecioná-lo, caso seja a primeira tentativa de conexão, será solicitada uma senha (que é 1234 por padrão). Com o pareamento estabelecido, o LED para de piscar indicando que a placa está apta a transmitir e receber dados (VIDADESILÍCIO HC06, 2020).

5 SERIAL BLUETOOTH TERMINAL

5.1 USANDO O APP SERIAL BLUETOOTH TERMINAL

Para evitar a nada simples tarefa de desenvolver um aplicativo de comunicação entre o smartphone e o Arduino foi utilizado nos experimentos o aplicativo Serial Bluetooth Terminal. Simples de ser obtido e instalado no smartphone, esse app (aplicativo) para Android utiliza uma interface muito fácil e amigável para enviar e receber dados por conexão Bluetooth. Ele é completamente gratuito, foi desenvolvido por Kai Morich e seu download pode ser feito pelo serviço de distribuição digital de aplicativos Google Play¹⁹.

O app utiliza uma interface bem simples e intuitiva tipo terminal/console orientado a linha para comunicação de dispositivo Bluetooth pareado a um smartphone Android (MORICH, 2019).

Ele suporta comunicação com diferentes dispositivos Bluetooth e utiliza diferentes versões do Bluetooth, mas no presente trabalho foi apenas utilizada a comunicação com a placa HC-06 com a versão do Bluetooth clássico.

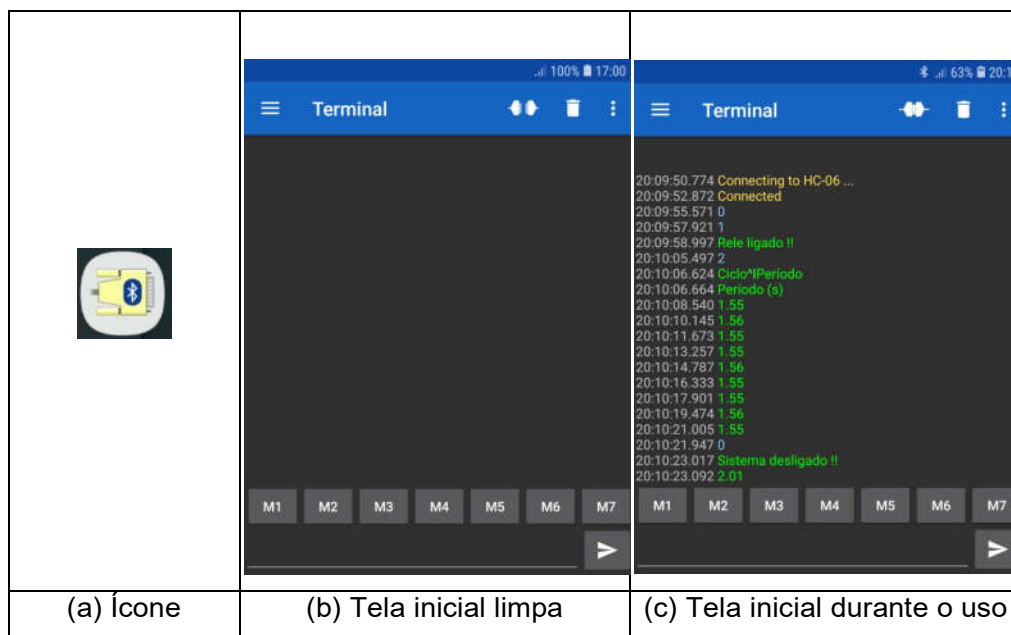
Após a instalação do app, o seu ícone, mostrado na Figura 5.1 (a), aparece na tela do smartphone.

Ao abrir o app, entra-se na tela do **Terminal**, mostrada na Figura 5.1 (b). A parte superior dessa tela contém quatro botões de comandos cujos ícones e funções encontram-se no Quadro 5.1. A região central da tela corresponde à área de mensagens e está inicialmente vazia. Ela registrará os dados de cada mensagem enviada e recebida pelo app, formando uma lista com o horário da mensagem, comandos e dados enviados/recebidos. Depois de algum tempo de

¹⁹Disponível em: https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal. Acesso em: 20 fev. 2019.

uso, a área de mensagens passa a ter um aspecto similar ao encontrado na Figura 5.1 (c). Finalmente, na parte inferior da tela encontram-se alinhados sete botões de macros e abaixo deles, a caixa de entrada para os dados a serem enviados pelo app.





Figura 5.1 – Ícone e tela inicial do app Serial Bluetooth Terminal.



Fonte: print screen do app no sistema Android.

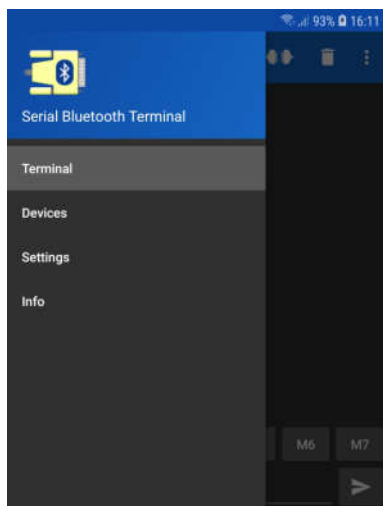
Caso seja a primeira vez de conexão é necessário fazer a configuração de pareamento. Para isso, com a placa HC-06 ativa e em busca de pareamento (com LED vermelho piscando) e com o smartphone já configurado para fazer conexão Bluetooth, deve-se acionar o botão de mudança de telas, localizado no canto superior esquerdo da tela inicial do Terminal e cujo ícone é mostrado no Quadro 5.1 (a).

Quadro 5.1 – Botões de comando do Serial Bluetooth Terminal.

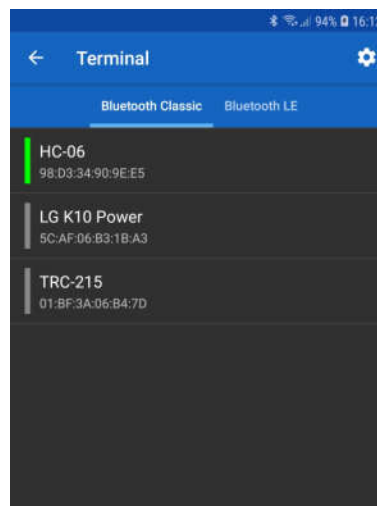
| | Ícone | Ação |
|-----|---|---------------------------|
| (a) |  | Mudança de telas |
| (b) |  | Conectar/Desconectar |
| (c) |  | Limpar lista de mensagens |
| (d) |  | Configurações |

Fonte: print screen do app no sistema Android.

Após acionar o botão de mudança de telas, aparece uma lista com quatro opções de telas disponíveis, como indica a Figura 5.2 (a).

Figura 5.2 – Lista de telas e Devices do Serial Bluetooth Terminal.

(a) Lista de Telas



(b) Tela Devices

Fonte: print screen do app no sistema Android.

Depois, acionar a opção **Devices** para ter acesso à lista com todos os dispositivos Bluetooth disponíveis, cujo aspecto é mostrado na Figura 5.2 (b). Nela existem duas abas: na primeira aba encontram-se os dispositivos disponíveis com a versão Bluetooth Classic; na segunda aba estão aqueles com a versão Bluetooth LE. A placa HC-06 deve ser selecionada na aba Bluetooth

Classic. Em seguida, será solicitada uma senha, que no caso da placa HC-06 é 1234 por padrão. Finalmente, a área de mensagens apresentará um aviso indicando o sucesso ou não do pareamento e o botão Conectar/Desconectar será representado por um dos dois ícones encontrados no Quadro 5.1 (b) que corresponda ao estado atual da conexão.

Caso a configuração de pareamento entre o smartphone e a placa HC-06 tenha sido configurada corretamente, acionar o botão Conectar/Desconectar tem como ação alternar entre os dois estados de conexão, sendo desnecessário fazer nova configuração em uso futuro.

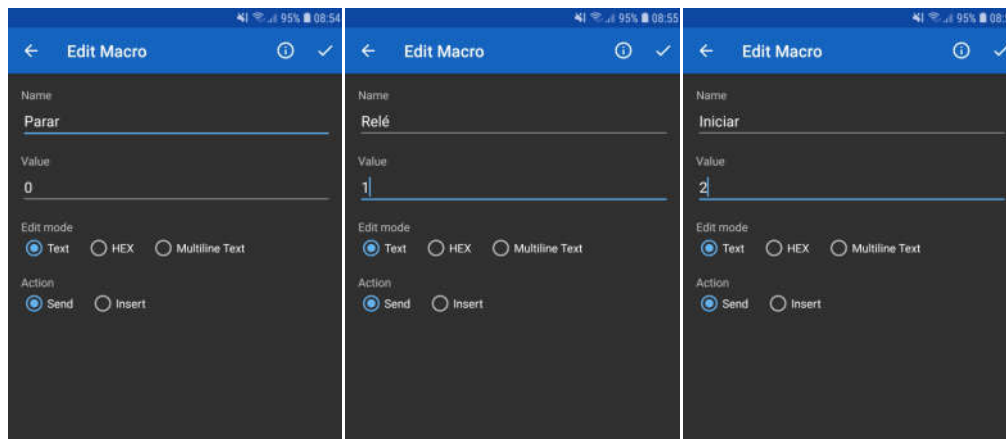
Com a conexão ativa é possível enviar diversos comandos e receber inúmeras informações. Após algum tempo a tela enche-se de dados que não são mais úteis e assim passa a ser interessante fazer a limpeza da área de mensagens, através do botão com ícone mostrado no Quadro 5.1 (c).

5.2 BOTÕES DE MACROS

É possível automatizar o envio de comandos e dados através do uso dos botões de macros. Para isso, basta pressionar um deles por dois ou três segundos até surgir a **Edit Macro** (tela de edição de macro). Com essa tela aberta, é possível inserir o nome para a macro, o valor desejado, o modo de edição (Texto, Hexadecimal ou Texto de múltiplas linhas) e a ação (Enviar ou inserir). Após preencher esses dados, deve-se acionar o botão de confirmação, encontrado no canto superior direito da tela para salvar as alterações.

Como exemplo, a Figura 5.3 (a), a Figura 5.3 (b) e a Figura 5.3 (c) mostram, respectivamente, as telas de edição de macro dos botões M1, M2 e M3 para automatizar o envio dos valores 0, 1 e 2 e trocar seus nomes para **'Parar'**, **'Relé'** e **'Iniciar'**. É importante observar que nesse caso o três botões devem enviar um valor e, portanto, no item **Action** de cada um deles deve ser escolhida a opção **Send**.

Figura 5.3 – Configuração dos botões de macros M1, M2 e M3.



(a) Botão M1

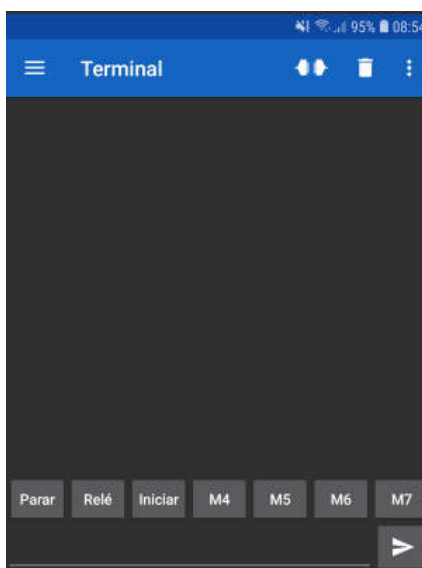
(b) Botão M2

(c) Botão M3

Fonte: print screen do app no sistema Android.

A Figura 5.4 mostra o aspecto da tela inicial após essa atualização.

Figura 5.4 – Resultado da configuração dos botões de macros M1, M2 e M3.



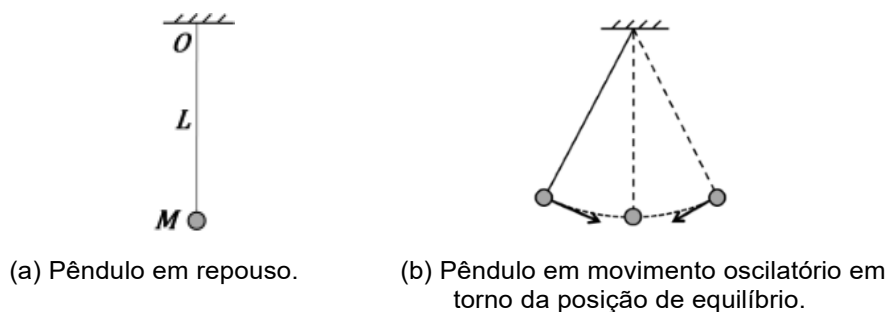
Fonte: print screen do app no sistema Android.

6 DISPOSITIVO DE MEDIDA DO PERÍODO

6.1 MONTAGEM DO PÊNDBULO SIMPLES

De forma geral, como mostra a Figura 6.1 (a), um pêndulo simples é constituído por um corpo de massa M concentrada e suspensa por um dos extremos de um fio inextensível, com massa desprezível, comprimento L e com o outro extremo fixo em um ponto O . Na condição de equilíbrio estático, quando o corpo está somente sob ação das forças gravitacional e força de tração no fio, o fio tem direção vertical. Quando o corpo é afastado de sua posição de equilíbrio e abandonado, como mostrado na Figura 6.1 (b), ele oscila em torno desta posição com um período de tempo T .

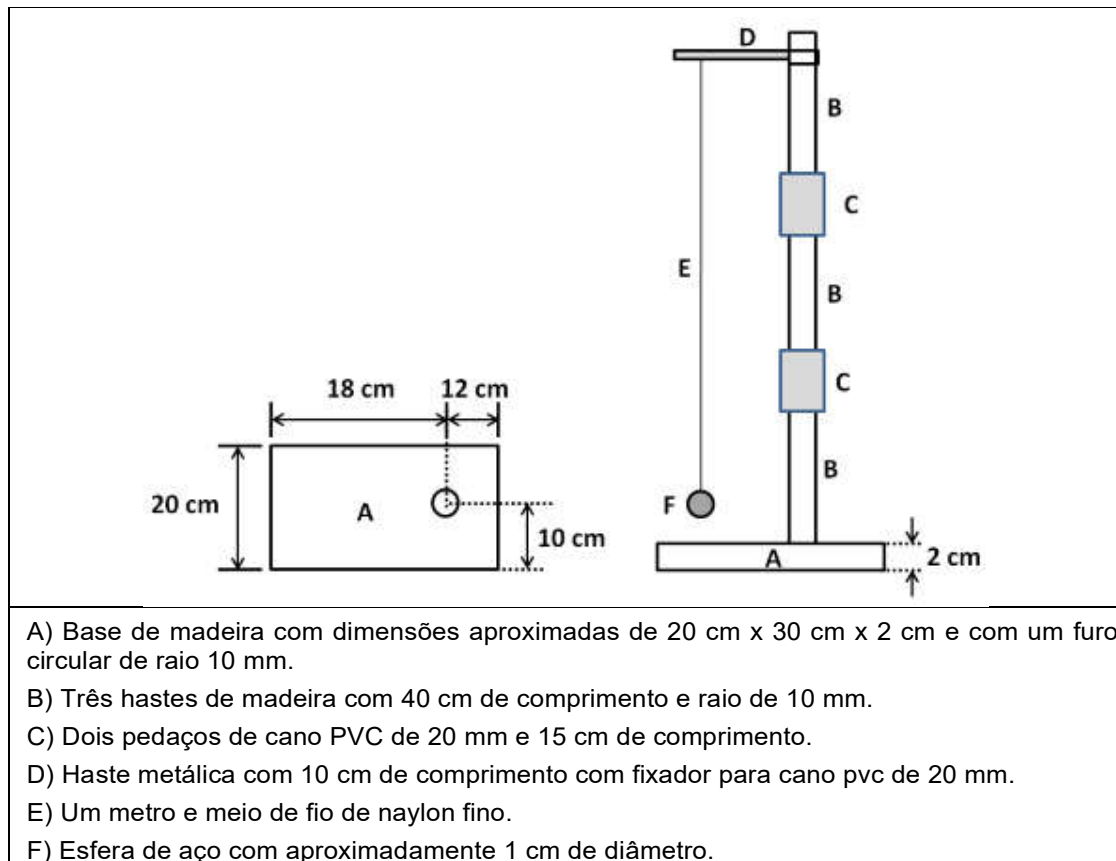
Figura 6.1 – Repouso e movimento do pêndulo simples.



Fonte: elaborado pelo autor.

A Figura 6.2 mostra o esquema usado para construção do pêndulo usado nesse trabalho. Nas medidas realizadas, L tinha valores entre 50 cm e 100 cm, por isso, foi utilizado um cabo de madeira com aproximadamente 1,2 m para ser a haste vertical. Com objetivo de facilitar o transporte, o cabo foi seccionado em três partes (item B da Figura 6.2), cada uma com aproximadamente 40 cm de comprimento e unindo-se facilmente com outra secção através de encaixes feitos com pedaços de tubos de PVC (item C da Figura 6.2). A haste vertical é então fixada na base de madeira (item A da Figura 6.2) pelo furo nela existente.

Figura 6.2 – Esquema de construção do pêndulo simples.



Fonte: elaborado pelo autor.

Como haste metálica (item D da Figura 6.2), foi usada uma haste galvanizada para telha com diâmetro de $\frac{1}{4}$ de polegada por 150 mm de comprimento, facilmente adquirida em qualquer loja de ferragens ou de material de construção. Ela foi serrada no lado rosqueado para atingir o comprimento de 10 cm. E com objetivo de facilitar a fixação do fio do pêndulo, ela também recebeu um furo de aproximadamente dois milímetros de diâmetro próximo a extremidade oposta àquela que foi cortada. Evidentemente, em seu lugar podem ser usados pedaços de madeira, pregos, parafusos ou arames, mas a haste para telha, além de ser barata e fácil de ser encontrada, tem boa resistência mecânica e tem diâmetro que se ajusta perfeitamente no furo de um fixador de tubos de PVC de 20 mm, esse também facilmente encontrado em qualquer loja de ferragens ou de material de construção.

Com objetivo de facilitar a mudança no comprimento do pêndulo, a haste para telha foi conectada ao fixador de tubos de PVC de 20 mm, como visto na Figura 6.3, e este encaixado à haste vertical de madeira. O encaixe é suficientemente rígido para suportar o movimento do pêndulo sem que a haste metálica saia de sua posição, mas permite que ela seja suspensa ou baixada quando necessário.

Finalmente, a esfera de aço com 1 cm de diâmetro foi adquirida em uma loja de aviamentos e bijuterias, mas pode também ser facilmente encontrada em sites da internet. Ela foi envolvida em um pedaço de plástico transparente comum com dimensões de 4 cm x 4 cm e amarrada ao fio do pêndulo, sendo o excesso de plástico aparado com uma tesoura.

Figura 6.3 – Haste metálica conectada ao fixador de tubos de PVC.



Fonte: elaborado pelo autor.

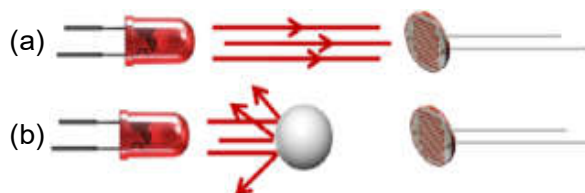
6.2 MEDIÇÃO DO PERÍODO DO PÊNDULO COM ARDUINO E LDR

Todo sistema de aquisição de dados possui pelo menos um sensor que faz a coleta de alguma grandeza física (luminosidade, temperatura, força, pressão, som, etc), converte em um sinal elétrico e envia esse sinal a um microcomputador ou computador para a análise dos dados. Atualmente existe uma enorme variedade de sensores e componentes eletrônicos de baixo custo que podem ser empregados com esse objetivo (HAAG; ARAUJO; VEIT, 2005).

No caso particular de medição de tempo pode-se citar, entre outros, o uso de sensores de som, com em Aguiar e Pereira (2012), ultrassom, como em Monk (2014, p. 79), infravermelho, como em Guaitolini Junior et al. (2016) e do par LED/LDR, como em Souza et al. (2011) ou em Soares e Borges (2010). Cada sensor apresenta vantagens e desvantagens uns em relação aos outros, mas cada um tem forma de aplicação e desempenho muito interessantes.

Para medir o período do pêndulo simples foi utilizado o par LED/LDR, seguindo a proposta de Soares e Borges (2010): Em uma situação inicial, como na Figura 6.4 (a), a luz do LED incide frontalmente sobre o LDR. Nessa situação a luminosidade que atinge o LDR é máxima; Ao ser colocado um objeto opaco entre os dois, como na Figura 6.4 (b), a luminosidade sobre o LDR diminui.

Figura 6.4 – Esquema de incidência da luz do LED sobre um LDR.



Adaptado de: Soares e Borges (2010), Mundo da Elétrica²⁰ e Sigmatechbd²¹.

Como visto no Capítulo 4, ao fazer uma conexão entre o LDR e o Arduino com um resistor em modo *pull-up*, na situação mostrada na Figura 6.4 (a) será enviado um sinal elétrico baixo para o Arduino. Já na situação mostrada na Figura 6.4 (b) será enviado ao Arduino um sinal elétrico mais intenso.

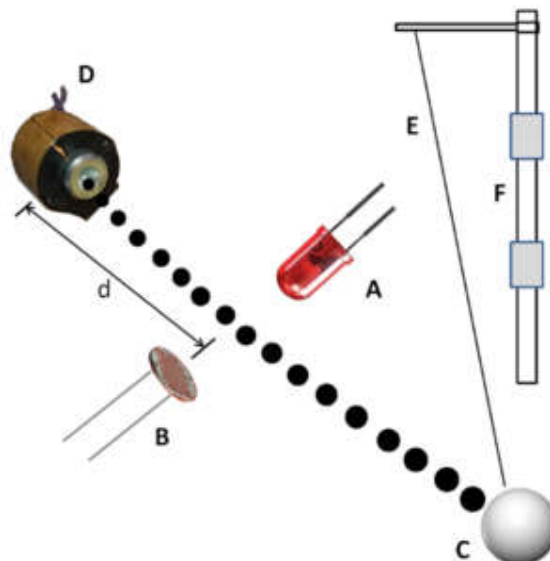
Se o objeto opaco for uma esfera de aço pendurada por um fio e representada pelo item C da Figura 6.5, ao ser liberada para mover-se, produzirá um movimento oscilatório tendo como trajetória um arco de circunferência passando entre o LED e o LDR. O período dessa oscilação será o

²⁰ Disponível em: <https://www.mundodaeletrica.com.br/i/1050/led-exemplos-1-600.webp>. Acesso em: 30 jul. 2019.

²¹ Disponível <https://sigmatechbd.com/wp-content/uploads/2018/02/LDR-Sensor.jpg>. Acesso em: 30 jul. 2019.

intervalo de tempo necessário para a esfera passar entre o LED e o LDR, atingir um extremo da sua trajetória, passar novamente entre o LED e o LDR, atingir o extremo oposto e passar mais uma vez entre o LED e o LDR.

Figura 6.5 – Componentes do pêndulo e do dispositivo de medida.



| | |
|------------------|-----------------------|
| A: LED | D: Eletroímã |
| B: LDR | E: Fio do pêndulo |
| C: Esfera de aço | F: Suporte do pêndulo |

Adaptado de: Mundo da Elétrica²² e Sigmatechbd²³.

Em resumo, com um resistor conectado em modo *pull-up* ao Arduino, no curto intervalo de tempo em que a esfera passa entre o LED e o LDR ocorre redução da luminosidade sobre o LDR, aumento da sua resistência e um sinal de valor alto é lido pelo Arduino.

Usando a notação do Quadro 6.1, o período do pêndulo pode ser medido como o tempo necessário para a esfera fazer o movimento B–C–D–C–B–A–B. O

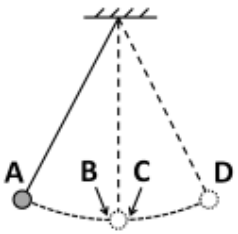
²² Disponível em: <https://www.mundodaeletrica.com.br/i/1050/led-exemplos-1-600.webp>. Acesso em: 30 jul. 2019.

²³ Disponível <https://sigmatechbd.com/wp-content/uploads/2018/02/LDR-Sensor.jpg>. Acesso em: 30 jul. 2019.

signal lido pelo Arduino é alto quando a esfera está no setor 1 e baixo quando ela está nos setores 0 e 2.

Quadro 6.1 – Setores de um ciclo de movimento da esfera.

| Posição inicial | Posição Final | signal | setor | sentido |
|-----------------|---------------|--------|-------|---------|
| A | B | baixo | 0 | 0 |
| B | C | alto | 1 | 0 |
| C | D | baixo | 2 | 0 |
| D | C | baixo | 2 | 1 |
| C | B | alto | 1 | 1 |
| B | A | baixo | 0 | 1 |

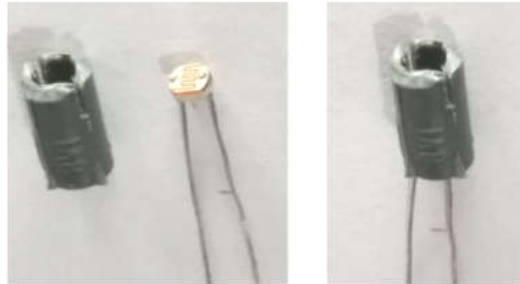


Fonte: elaborado pelo autor.

Dessa maneira, a esfera entra no setor 1 em duas situações: vinda do ponto A para o ponto B, quando está no setor 0 e tem sentido 0; ou vinda do ponto D para o ponto C, quando está no setor 2 e tem sentido 1. Assim, o movimento B–C–D–C–B–A–B sempre inicia quando o sinal recebido pelo Arduino passa de baixo para alto, instante em que a posição da esfera passa do setor 0 para o setor 1 e tem sentido de movimento obrigatoriamente igual a 0.

É importante observar que a luminosidade ambiente também atua sobre o LDR e a variação de sua intensidade pode gerar medidas que sinalizam ao Arduino uma falsa mudança no setor em que a esfera se localiza e assim produzir medidas completamente distorcidas. Uma maneira eficiente de diminuir bastante esse problema é usar um invólucro de material opaco em torno do LDR. Por exemplo, como mostrado na Figura 6.6, uma bucha para fixar parafusos em concreto, se cortada pela metade pode ser uma solução simples e muito eficiente para diminuir a ação da luz ambiente sobre o LDR.

Figura 6.6 – Invólucro opaco do LDR contra efeitos da luz ambiente.



Fonte: elaborado pelo autor.

6.3 CONTROLE DO INÍCIO DO MOVIMENTO DO PÊNDULO

A esfera do pêndulo tem diâmetro aproximado de um centímetro e como é feita de aço, um material relativamente denso, seu atrito com o ar é desprezível para a medição do período. O aço da esfera tem ainda outra vantagem: permite que seja usado um eletroímã (item D da Figura 6.5) para iniciar o movimento, garantindo que a velocidade inicial da esfera seja sempre nula e também que ela seja solta sempre da mesma posição, o que torna tanto o ângulo inicial do fio do pêndulo com a reta vertical e a direção do movimento sempre os mesmos em todas as medidas. Isso evita que a esfera seja lançada em direção ao LED ou LDR, podendo causar algum tipo de dano a esses dispositivos. Para tentar evitar que isso ocorra, o lançamento com as mãos em geral torna-se mais lento, não sendo incomum a necessidade de repetição do lançamento por acontecer o choque da esfera com um deles.

Outra vantagem do uso do eletroímã é garantir a condição harmônica do pêndulo, como visto no capítulo 3. Sendo d a metade da distância entre os pontos extremos da trajetória do pêndulo e sendo θ_0 o ângulo inicial do fio da esfera em relação ao eixo vertical, é possível obter os valores de θ_0 mostrados na Tabela 6.1.

Tabela 6.1 – Relação entre o raio e o ângulo inicial do pêndulo.

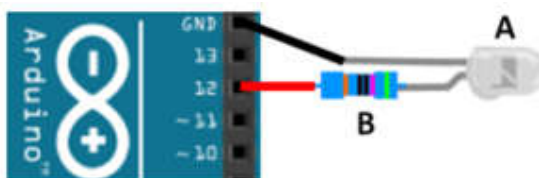
| L (cm) | d (cm) | $\text{sen}(\theta_0) = d/L$ | θ_0 (rad) | θ_0 (grau) |
|--------|--------|------------------------------|------------------|-------------------|
| 50 | 9 | 0,18000 | 0,18099 | 10,36976 |
| 60 | 9 | 0,15000 | 0,15057 | 8,62693 |
| 70 | 9 | 0,12857 | 0,12893 | 7,38705 |
| 80 | 9 | 0,11250 | 0,11274 | 6,45945 |
| 90 | 9 | 0,10000 | 0,10017 | 5,73917 |
| 100 | 9 | 0,09000 | 0,09012 | 5,16361 |

Fonte: elaborado pelo autor.

Assim, a pior situação de precisão da medida ocorre para $L = 50$ cm, ainda dentro do intervalo de aproximadamente 10 graus ou menos para a validade da condição harmônica (SERWAY e JEWETT JR; 2011, p. 3).

6.4 ESQUEMAS ELÉTRICOS DO DISPOSITIVO DE MEDIÇÃO

Existem quatro circuitos elétricos necessários ao funcionamento do projeto. O primeiro deles, mostrado na Figura 6.7 consiste no circuito do LED.

Figura 6.7 – Circuito elétrico de ligação do LED ao Arduino.

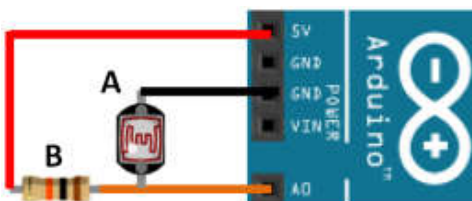
A: LED de alto brilho **B:** Resistor de $570 \Omega \pm 1\%$

Adaptado de: programa Fritzing.

Seu ânodo (+ 5 V) está conectado a um resistor e este ao pino 12 do Arduino. Apesar de poder ser usado um resistor de menor valor em série ao LED, a resistência de 570Ω permitiu ao LED ter desempenho satisfatório.

O segundo circuito é o do LDR, mostrado na Figura 6.8. O resistor é conectado em modo *pull-up*. O LDR tem seus terminais conectados ao pino analógico A0 e ao pino GND. Assim, quando o LED ilumina diretamente o LDR, o Arduino recebe um sinal elétrico baixo. Mas quando a esfera passa entre o LED e o LDR, diminui a luminosidade incidente no LDR e o Arduino passa a receber um sinal elétrico mais intenso.

Figura 6.8 – Circuito elétrico de ligação do LDR ao Arduino.

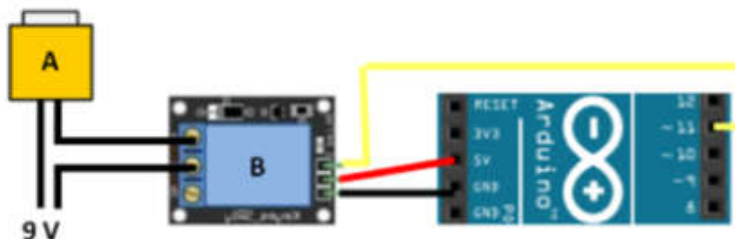


A: LDR B: Resistor de $10\text{ k}\Omega \pm 5\%$

Adaptado de: programa Fritzing.

O terceiro circuito é do eletroímã, mostrado na Figura 6.9. O pino IN do rele está conectado ao pino 11 do Arduino.

Figura 6.9 – Circuito elétrico de ligação do eletroímã ao Arduino.



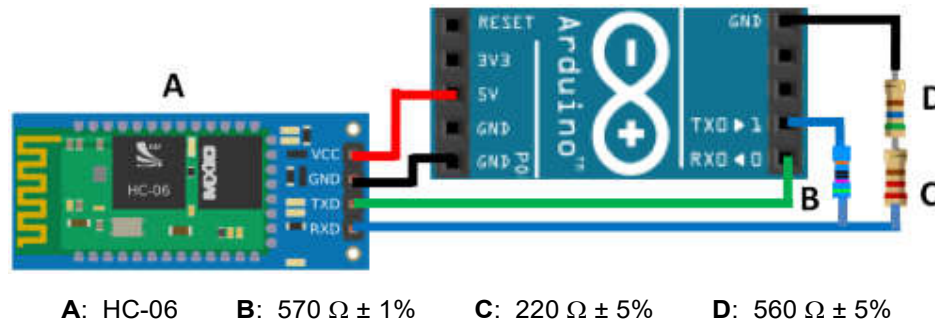
A: Eletroímã B: Módulo relé de um canal

Adaptado de: programa Fritzing.

Para evitar alguma interferência no Arduino devido ao eletroímã é aconselhável que eles tenham fontes de alimentação elétrica independentes.

O quarto e último circuito é o de comunicação Bluetooth, mostrado na Figura 6.10.

Figura 6.10 – Circuito elétrico de ligação do Bluetooth ao Arduino.



Adaptado de: programa Fritzing.

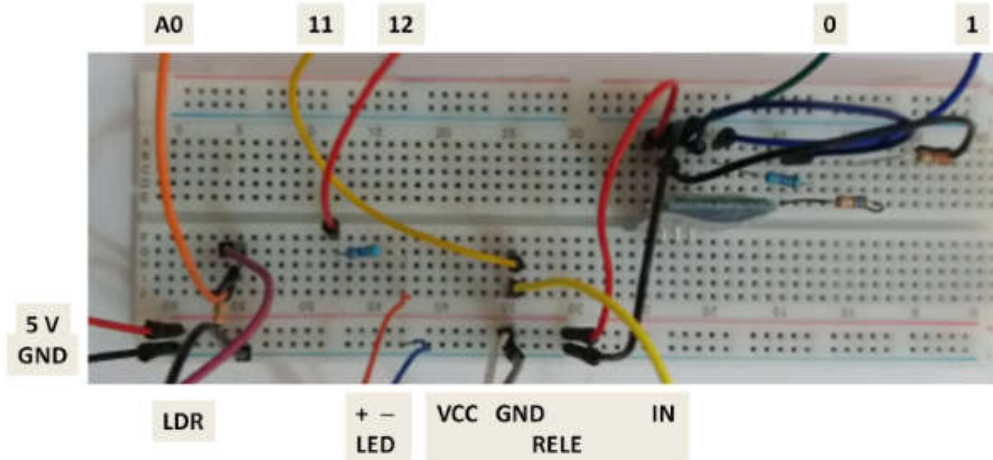
Um resistor de 570 Ω (item B da Figura 6.10) é usado em um dos lados do divisor de tensão. No outro lado do divisor são usados dois resistores ligados em série com resistências de 220 Ω (item C da Figura 6.10) e 560 Ω (item D da Figura 6.10), fazendo o HC-06 (item A da Figura 6.10) receber a ddp de:

$$\frac{560 + 220}{570 + 560 + 220} \cdot 5 \text{ V} = \frac{780}{1350} \cdot 5 \text{ V} \approx 2,89 \text{ V} \quad (6.1)$$

Essa tensão foi suficiente para o HC-06 funcionar satisfatoriamente e não ter sido verificado nenhum problema em sua operação. Inicialmente, seriam usados resistores com valores iguais a 1 kΩ e 2,2 kΩ, seguindo a proposta encontrada em Eletrogate (2019).

Na Figura 6.11 é apresentado, montado numa protoboard, o circuito elétrico do dispositivo de medição do período. No lado esquerdo encontra-se o circuito referente ao LDR. Junto a ele e mais à direita, encontra-se o circuito do LED. Ao centro, está o circuito de controle do relé do solenóide e por fim, do lado direito da protoboard encontra-se o circuito do HC-06.

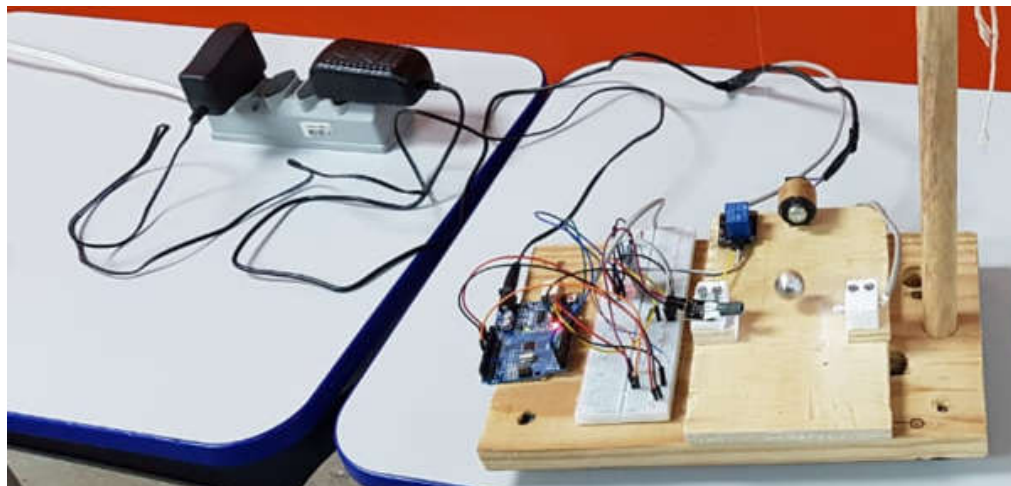
Figura 6.11 – Circuito de medição do período na protoboard.



Fonte: elaborado pelo autor.

A Figura 6.12 mostra o dispositivo de medição do período juntamente com a base de sustentação do pêndulo e as fontes de tensão elétrica usadas para alimentar o Arduino (9 V – 1 A) e o eletroímã (9 V – 2 A).

Figura 6.12 – Circuito de medição e fontes de tensão.



Fonte: elaborado pelo autor.

Finalmente, a Figura 6.13 mostra o dispositivo em uso durante a realização de um experimento.

Figura 6.13 – Dispositivo de medida do período do pêndulo simples.



Fonte: elaborado pelo autor.

6.5 DETERMINAÇÃO DA FREQUÊNCIA OPERAÇÃO DO LOOP

O loop do *sketch* deve ter sua frequência de funcionamento limitada pela máxima frequência de operação do LDR que é da ordem de alguns quilohertz (BRAGA, 2019). Esse limite é fundamental para evitar leituras incorretas do LDR.

Uma estratégia para determinar a frequência de operação de leitura do LDR é controlar o circuito elétrico de medição do período com o *sketch* de teste:

```
int tempo0

void setup() {
  tempo0 = millis()
  Serial.begin(9600);
}
```



```

void loop() {
  int valor = analogRead(0);
  int tempo1 = millis();
  Serial.println(tempo1-tempo0);
  tempo0 = tempo1;
}

```

Nesse procedimento, inicialmente é criada a variável inteira **tempo0**. A seguir, no **setup**, o tempo em milissegundos decorrido desde o início da execução do *sketch* é obtido pela função **millis()** e armazenado em **tempo0**. Depois, a porta serial é configurada para trabalhar a 9600 bps. No **loop**, primeiro a variável inteira **valor** é usada para receber a leitura do LDR. Em seguida, nova leitura de **millis()** é feita e armazenada na variável **tempo1**. O intervalo de tempo entre dois loops consecutivos é dado pela diferença **tempo1 - tempo0** e o resultado enviado para a porta serial. Finalmente, antes do loop reiniciar, a variável **tempo0** é atualizada e passa a ter o valor de **tempo1**.

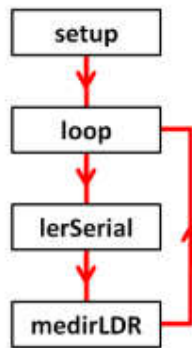
Após a realização de inúmeras medições usando esse sketch de teste, o valor médio encontrado para a frequência do loop foi da ordem de 10 000 ciclos por segundo, muito acima dos poucos quilohertz que permitem uma leitura confiável do LDR.

Como o *sketch* de teste tem uma frequência próxima à do *sketch* realmente usado na medida do período do pêndulo, será necessário usar uma estratégia para reduzir seu valor a um nível que permita uma medida confiável do LDR.

6.6 SKETCH PARA CONTROLE DO DISPOSITIVO DE MEDIÇÃO

Para controlar o sistema de medição foi desenvolvido o *sketch* **PENDULO_SIMPLES** (Apêndice A). Ele é constituído por cinco partes: a lista de variáveis globais e procedimentos, **setup**, **loop**, **lerSerial**, e **medirLDR**. O seu fluxograma é mostrado na Figura 6.14.

Figura 6.14 – Fluxograma do sketch de controle.



Fonte: elaborado pelo autor.

6.6.1 Variáveis globais do sketch

A lista de variáveis globais é:

```

int pinoRELE = 11; // pino de conexão do Rele
int pinoLED = 12; // pino de conexão do LED
int pinoLDR = 0; // pino de entrada do LDR
int ciclo; // número de ciclos medidos
int setor; // setor atual da posição da esfera
int sentido; // sentido do movimento da esfera
int valorBase; // valor mínimo da medida do LDR
float taxa = 1.1; // taxa indicativa de início de ciclo
int tempo0; // instante de início do ciclo
char buf0 = '0'; // valor anteriormente lido na porta serial
  
```

Para identificar os pinos de conexão do relé, do LED e do LDR são usadas as variáveis globais **pinoRELE**, **pinoLED** e **pinoLDR**. O número de ciclos medidos é registrado na variável global inteira **ciclo**.

De acordo com a nomenclatura presente no Quadro 6.1, os valores do setor e do sentido de movimento da esfera serão respectivamente armazenados nas variáveis globais **setor** e **sentido**.

Uma pequena mudança na luminosidade ambiente ou até mesmo uma pequena alteração na distância entre a esfera e o LDR no momento em que ela obstrui a luz vinda do LED modificam a intensidade da luminosidade sobre o

LDR e pode gerar medidas que levem a um erro na determinação da posição da esfera. Para diminuir esses inconvenientes foi tomada uma estratégia que faz uso das variáveis **valorBase**, **valor** e **taxa**.

Quando o eletroímã está energizado e a esfera conectada a ele, de acordo com o Quadro 6.1, ela encontra-se no ponto A do setor 0. Nesse instante, o Arduino recebe um sinal baixo do LDR. Quando o eletroímã é desligado e a esfera é liberada para mover-se, o Arduino registra o valor do sinal enviado pelo LDR na variável global inteira **valorBase**. Esse é o menor valor lido pelo LDR durante toda a medição, desde que a luminosidade ambiente não sofra aumento.

Durante o movimento da esfera, milhares de loops são realizados pelo *sketch* e em cada um deles o procedimento **lerLDR** recebe a leitura do sinal do LDR. O valor desse sinal é registrado na variável local inteira **valor** e em seguida comparado com **valorBase** para determinar, de acordo com o Quadro 6.1, em que setor se encontra a esfera.

A comparação $\text{valor} > \text{taxa} * \text{valorBase}$, em que **taxa** é uma variável global de ponto flutuante, determina quanto maior deve ser **valor** em relação a **valorBase** para ser considerado um sinal que corresponda à passagem da esfera entre o LED e o LDR. Testes com diversos níveis de luz ambiente e com pequenas alterações na distância entre a esfera e o LDR no ponto em que ela fica entre ele e o LED, indicaram que **taxa** igual a 1.1 era suficiente para eliminar quase por completo medidas incorretas devido a esses efeitos indesejados.

A variável **tempo0** será usada no procedimento **lerLDR** para armazenar o instante de tempo que um ciclo tem início.

Por último, é definida a variável global do tipo **char** e de nome **buf0**, cujo valor inicial é '0'. Ela será usada no procedimento **lerSerial** para armazenar a ação enviada pelo smartphone.

6.6.2 Procedimento setup

No setup encontram-se os seguintes comandos:

```
void setup() {  
  pinMode(pinoRELE, OUTPUT);  
  pinMode(pinoLED, OUTPUT);  
  digitalWrite(pinoRELE, LOW);  
  digitalWrite(pinoLED, LOW);  
  Serial.begin(9600);  
  Serial.println((String)"Sistema desligado !!!");  
}
```

As duas primeiras linhas informam ao Arduino que os pinos digitais 11 e 12 são do tipo OUTPUT. As terceira e quarta linhas informam que esses pinos estão no estado LOW, ou seja, desligados.

A penúltima linha inicia a comunicação serial com uma taxa de 9600 bps e a última linha escreve na porta serial a frase "Sistema desligado !!!" e a envia ao smartphone via Bluetooth.

6.6.3 Procedimento loop

O loop apresenta apenas três comandos:

```
void loop() {  
  lerSerial();  
  medirLDR();  
  delay(2);  
}
```

Os procedimentos **lerSerial** e **MedirLDR** serão executados repetidamente até que o Arduino seja desligado ou reiniciado. Testes realizados sem uso da função **delay** mostraram para o loop uma frequência de funcionamento da ordem de 10 kHz. Como visto anteriormente, isso causa um problema, pois o

LDR opera com frequência de operação da ordem de alguns kHz. A função **delay(2)** causa um atraso de 2 milésimos de segundo entre o fim de um loop e o início do próximo. Como o experimento tem a medida do período realizada apenas com centésimos de segundo, esse pequeno atraso é irrelevante para a sua precisão, porém é fundamental para serem obtidas leituras corretas do LDR que não causem erros de lógica na execução do *sketch*.

6.6.4 Procedimento lerSerial

O aparelho smartphone se comunica com o Arduino enviando códigos de ação de acordo com o Quadro 6.2. O procedimento **lerSerial** é responsável por receber esse código e comparar com o valor de **buf0**, que armazena o valor recebido na leitura anterior. Quando o valor recebido for diferente de **buf0** e for 0, 1 ou 2, a ação correspondente a ele é executada. Essa comparação é importante para evitar que sucessivos e desnecessários envios do mesmo código por parte do usuário sejam indevidamente executados pelo Arduino.

Quadro 6.2 – Códigos de ação enviados pelo smartphone.

| Valor lido | Ação |
|------------|--|
| 0 | Desligar o eletroímã, desligar o LED e parar a medição |
| 1 | Ligar só o eletroímã. |
| 2 | Desligar o eletroímã, ligar o LED e iniciar medição |

Fonte: elaborado pelo autor.

O procedimento **lerSerial** tem a seguinte estrutura:

```
void lerSerial(){
  while(Serial.available() > 0){
    char buf = Serial.read();
    if (buf != buf0){
      buf0 = buf;
      if (buf == '0'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, LOW);
        Serial.println((String)"Sistema desligado !!!");
      }
    }
  }
}
```

```

    }
    else if (buf == '1'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, HIGH);
        Serial.println((String)"Relé ligado !!!");
    }
    else if (buf == '2'){
        ciclo = 0;
        setor = 0;
        sentido = 0;
        digitalWrite(pinoLED, HIGH);
        delay(1000);
        digitalWrite(pinoRELE, LOW);
        Serial.println((String)"Período (segundos:");
        valorBase = map(analogRead(pinoLDR),0,1023,0,100);
    }
}
}
}
}

```

A instrução `while` se repete enquanto houver algum byte disponível para leitura da porta serial. Caso exista, seu valor é posto na variável local **buf**. Em seguida o teste **buf != buf0** (**buf** é diferente de **buf0**?) é realizado. Em caso verdadeiro, o procedimento modifica o valor de **buf0** para torna-se igual ao de **buf** e em seguida realiza a comparação do valor de **buf** com os códigos das ações contidas no Quadro 6.2.

Com **buf** igual 0, de acordo com o Quadro 6.2, uma ação de parar o experimento é executada. Assim, os pinos 12 do LED e 11 do relé são desligados e é enviada para a porta serial uma mensagem de encerramento.

Caso **buf** seja igual a 1, novamente de acordo com o Quadro 6.2, o relé do eletroímã é energizado e o LED desenergizado. Também é enviada para a porta serial a mensagem de acionamento do relé.

Finalmente, se **buf** for igual a 2, de acordo com o Quadro 6.2, a medição do experimento deve iniciar. Assim, as variáveis globais **ciclo**, **setor** e **sentido** são iniciadas com valores iguais a 0. Em seguida o LED é energizado e após um segundo o relé é desligado. O uso de **delay(1000)** é necessário para garantir

que a esfera atinja a posição entre o LED e o LDR com este já iluminado pelo LED e apto a fazer leituras corretas. Outros valores menores poderiam ser usados, mas por segurança o intervalo de um segundo foi usado. Em seguida uma mensagem indicando que a medição se inicia é enviada para a porta serial. Por fim, o valor lido pelo pino do LDR é lido e remapeado para um valor entre 0 e 100 e esse valor é armazenado na variável **valorBase**.

A função **map** remapeia um número de um intervalo para outro, truncando o resultado para um número inteiro se necessário. Dessa forma, a expressão **map(x, 0, 1023, 0, 100)** irá converter o valor de x para:

$$x \cdot \frac{100 - 0}{1023 - 0} \quad (6.2)$$

Assim, por exemplo, os valores de 502 até 511 serão convertidos para o valor 49. Essa conversão será útil para diminuir a sensibilidade do sinal enviado pelo LDR e aumentar a segurança de acerto na comparação entre **valor** e **valorBase**, realizada no procedimento **medirLDR**.

6.6.5 Procedimento medirLDR

De acordo com o Quadro 6.1, a leitura do Arduino será de nível elétrico alto somente no setor 1 e baixo nos setores 0 e 2. Assim, para terminar um ciclo e iniciar outro, a esfera deve sair do setor 0 e entrar no setor 1 com movimento no sentido 0. Somente com essas duas condições atendidas é que deve ser iniciada a medição do período de um novo ciclo. Seguindo essa estratégia, o procedimento **medirLDR** tem como código:

```
void medirLDR(){
  int valor = map(analogRead(pinoLDR),0,1023,0,100);
  if (valor > taxa * valorBase){ // Teste I
    if (setor == 0) { // Teste II
      setor = 1;
      sentido = 0;
```

```

int tempo = millis();
if (ciclo > 0){ // Teste III
  float periodo = (tempo-tempo0) / 1000.;
  Serial.println((String)periodo);
}
tempo0 = tempo;
ciclo = ciclo + 1;
}
else if (setor == 2){ // Teste IV
  setor = 1;
  sentido = 1;
}
}
else{
  if (setor == 1){ // Teste V
    if (sentido == 0){ Teste VI
      setor = 2;
    }
    else{
      setor = 0;
    }
  }
}
}
}
}
}

```

Primeiro, o valor lido pelo LDR e enviado ao Arduino pelo pino analógico A0 é remapeado entre 0 e 100 e o resultado armazenado na variável **valorBase**. Em seguida, no **Teste I** é feita a comparação **valor > taxa * valorBase**. Caso esse teste seja verdadeiro, a luz do LED que incide no LDR está sendo encoberta parcialmente ou totalmente pela esfera. Assim, pelo Quadro 6.1, a esfera se localiza no setor 1. Caso o **Teste I** seja falso, a esfera está no setor 0 ou no setor 2.

Com o **Teste I** sendo verdadeiro é necessário saber de onde a esfera veio. Por isso é realizado o **Teste II**. Nele é feita a comparação **setor == 0** (setor é igual a 0?). Caso ele seja verdadeiro, a variável **setor** é atualizada para 1 e a variável **sentido** para 0 (Quadro 6.1). Em seguida é executada a instrução **int tempo = millis()**. Ela cria a variável local inteira de nome **tempo** e nela armazena o intervalo total em milissegundos que o *sketch* está em execução. Em outras palavras, como a esfera acabou de entrar no setor 0, **tempo** armazena o instante final do ciclo anterior e que também é o instante inicial do

novo ciclo. O **Teste III** verifica se foi completado pelo menos o primeiro ciclo. Se verdadeiro, será executada a instrução **float periodo = (tempo-tempo0) / 1000**. O valor da variável **periodo** é então enviado para a porta serial e depois para o smartphone. Finalizando o bloco de instruções da condição verdadeira do **Teste II**, a variável **tempo0** é atualizada e recebe o valor de **tempo** e a variável **ciclo** é incrementada de 1.

Caso o **Teste II** seja falso, é realizado o **Teste IV**. Nele é feita a comparação **setor == 2** (setor é igual a 2?). Se ele for verdadeiro, significa que a esfera entrou no setor 1 vinda pelo setor 2 e com sentido 1, assim a variável **setor** é atualizada para 1 e a variável **sentido** para 1.

O **Teste I** sendo falso significa que o LED recebe luz direta do LDR e a esfera está no setor 0 ou no setor 2. Assim, é necessário apenas atualizar a variável **setor** se ela ainda estiver com valor 1. Por isso é realizado o **Teste V**, em que é feita a comparação **setor == 1** (setor é igual a 1?). Somente caso esse teste seja verdadeiro é feito o **Teste VI**, em que a comparação **sentido == 0** (sentido é igual a 0?) é realizada. Se ele for verdadeiro, a variável **setor** é atualizada para 2, caso contrário ela é atualizada para 0.

7 APLICAÇÃO DA PROPOSTA DIDÁTICA

7.1 PERFIL DA ESCOLA E DOS EDUCANDOS PARTICIPANTES

A aplicação da proposta ocorreu na Escola Técnica Estadual Cícero Dias, localizada na cidade de Recife, Estado de Pernambuco. Nela são oferecidas duas modalidades de ensino: No horário diurno, das 07:30 h às 17:00 h, funciona o curso técnico integrado ao ensino médio; No horário noturno, das 19:00 h às 22:00, funciona o curso técnico subsequente ao ensino médio. Na modalidade integrada ao ensino médio existem dois cursos, um de Programação de Jogos Digitais e outro de Multimídia, constituindo um total de doze turmas, duas de cada curso para cada uma das três séries do ensino médio.

O experimento foi realizado em uma das turmas da segunda série do curso de programação para jogos digitais no transcorrer da 2ª unidade do ano de 2019 e com um total de 39 educandos participantes. Aconteceu na sala de aula da turma, pois o laboratório usado pela disciplina de Física tem capacidade de comportar no máximo 24 educandos.

7.2 PERFIL DA PROPOSTA E AVALIAÇÃO DIAGNÓSTICA

Foi proposta uma atividade qualitativa do tipo verificacionista para determinar experimentalmente a lei de relacionamento do período do pêndulo com seu comprimento e compará-la com a equação teórica já apresentada a eles em aula anterior. Essa escolha estimulava nos educandos o aprimoramento de habilidades de registro e manipulação matemática de dados e exigia, entre outras coisas, realizar uma linearização simples dos dados com uso de logaritmos.

Ao final da aula teórica sobre movimentos oscilatórios foi apresentada aos educandos como atividade avaliativa uma ficha de exercícios, a qual continha questões envolvendo a relação entre o comprimento e período do pêndulo simples. E a análise dessas fichas demonstrou um rendimento satisfatório de todos os educandos da turma.

Em concordância com os pressupostos pedagógicos, essa atividade também serviu como primeira etapa da avaliação diagnóstica para o experimento, indicando que os educandos tinham um conhecimento satisfatório sobre a relação de dependência do período do pendulo simples com seu comprimento. Imediatamente após o final dessa atividade, um diálogo com os educandos foi usado como uma segunda etapa da avaliação diagnóstica, para levantar seus conhecimentos prévios sobre outros conteúdos necessários a realização do experimento. Assim, foram feitas as seguintes perguntas:

Pergunta 1: Vocês já presenciou a realização de algum experimento?

Resposta: Todos responderam que assistiram a realização de alguns.

Pergunta 2: Vocês já participaram de algum experimento que permitisse realizar alguma medida?

Resposta: Todos responderam que participaram somente de um. Na série anterior, foi realizado um experimento em grupo de medição de suas alturas e massas para cálculo do IMC e das médias dessas medidas por grupo.

Pergunta 3: Vocês já criaram algum experimento de física na escola?

Resposta: Todos responderam que não.

Pergunta 4: Vocês têm conhecimento de logaritmos?

Resposta: A maioria respondeu que sim, mas que era um conteúdo difícil e precisavam ter uma revisão.

As respostas indicaram que eles apresentavam familiaridade apenas com experimentos de demonstração, nenhum treino em experimentos não estruturados ou com ênfase investigativa. Assim, para compatibilizar as exigências do experimento aqui proposto com os conhecimentos prévios dos educandos e com o efetivo tempo disponível, optou-se pela realização de um experimento estruturado com grau de liberdade 1 (Quadro 2.1).

A avaliação diagnóstica detectou ainda que os educandos consideravam difícil o uso de logaritmos, conteúdo necessário para a análise dos resultados do experimento. Para tentar sanar esse problema, após um contato com a professora de matemática da turma, está se prontificou a fazer uma atividade de revisão sobre logaritmos, a ser realizada em uma de suas aulas.

Assim, após a aula teórica sobre movimento oscilatório e da aula de revisão de logaritmos, foi iniciada a aplicação do experimento, que em concordância com a proposta construtivista desenvolvida por ROSA (2012), foi dividida em três etapas, cada uma delas realizada em uma aula com 50 minutos de duração.

7.3 ETAPA PRÉ-EXPERIMENTAL

Realizada na sala de aula dos educandos, apresentou cinco etapas:

i) Em uma exposição dialogada, inicio-se uma rápida revisão teórica sobre o pêndulo simples. Assim, utilizando um projetor, em um primeiro slide foi apresentada a equação teórica do período pêndulo simples como função do seu comprimento.

Em um segundo slide, tomando a aceleração gravitacional por $9,8 \text{ m/s}^2$ e π por 3,14, a expressão do período pêndulo simples foi apresentada como:

$$T = 2,00 \cdot L^{0,5} \quad (7.1)$$

Foi então informado que chegar a essa equação a partir da realização do experimento proposto seria o grande objetivo da atividade.

ii) Diálogo sobre a importância de se estabelecer um roteiro bem definido para realização de um experimento físico.

Nesse momento todos os componentes do pêndulo e do aparato de medição foram colocados sobre uma banca previamente posicionada ao lado do birô do professor, de modo a ficarem visíveis a todos os alunos. Isso gerou certa agitação entre os educandos e vários chegaram a se levantar para manipular os componentes. Deixou-se que eles satisfizessem um pouco da curiosidade, mas depois de alguns instantes, foi solicitado que todos voltassem para seus lugares. Em seguida, um terceiro slide foi apresentado com o questionamento:

- É possível fazer com que todos educandos da turma de uma única vez façam a medição do período do pêndulo?

Rapidamente um dos educandos respondeu:

- Claro que não professor, deve-se definir uma sequência.

A partir dessa resposta foi lembrado aos educandos que em qualquer atividade a ser realizada, em particular nesse experimento, para se atingir os objetivos propostos seria fundamental definir uma série de detalhes sobre o que se quer fazer e de como fazer. Quanto mais complexo o objetivo, maior o detalhamento.

Foi explicado também que, no experimento em questão, essa sequência estava disponível na Ficha de Experimento do Pêndulo Simples (Apêndice B). E como era a primeira experiência mais elaborada realizada por eles, que exigiria a manipulação de uma série de componentes eletrônicos, os quais se comunicariam com um smartphone e também o uso de ferramentas matemáticas como o logaritmo, a ficha foi desenvolvida sem a participação deles, mas à medida que eles ganhassem mais experiência, com a realização de outros experimentos e passassem a desenvolver seus próprios projetos, poderiam e deveriam criar suas próprias fichas. Porém, no momento, o importante era entender bem os passos da ficha e após executá-los chegar ao objetivo planejado.

iii) Apresentação da Ficha de Experimento do Pêndulo Simples, seguida por uma discussão detalhada da Etapa 1 dessa ficha, em particular a apresentação e análise da hipótese orientadora do experimento.

Nesse momento foi distribuída uma cópia impressa da ficha para cada educando. Foi esclarecido que essa era uma cópia deles, devendo ser usada como material de estudo deles. Posteriormente outra cópia da ficha seria distribuída para cada grupo e só após seu preenchimento é que ela seria devolvida.

Em seguida, usando o projetor, a exposição dialogada prosseguiu com a apresentação da etapa pré-experimental da ficha, quando foi abordado o objetivo do experimento (tópico 1.1), a definição do que seria um pêndulo simples (tópico 1.2), o material usado na sua construção e outros materiais necessários tais como o circuito do Arduino e o smartphone (tópico 1.3).

Depois, dois educandos foram selecionados para fazer a montagem do aparato do pêndulo à medida que era apresentado o tópico referente a ele na ficha (tópico 1.4).

Logo após, foi apresentado aos educandos detalhes do dispositivo de medição do período, sendo dada uma breve explicação sobre o funcionamento do Arduino e de sua comunicação via Bluetooth com o smartphone e também do LED e do LDR.

A seguir foi apresentado o app Serial Bluetooth Terminal como o software responsável pelo gerenciamento do experimento e dada uma pequena explicação de seu uso.

Para finalizar essa etapa, a hipótese do experimento, contida no tópico 1.5 da ficha foi apresentada. Como ela era uma função envolvendo uma potência, foi informado seriam usadas as propriedades dos logaritmos para dela se obter uma função polinomial do 1º grau, mais simples de se manipular.

iv) Definição de um grupo de alunos monitores do experimento

Foi definido que um grupo de alunos monitores formado por quatro pessoas seria responsável por coordenar e orientar os demais educandos durante o experimento. Como critério de seleção para esse grupo foi usado o interesse e desempenho deles.

v) Definição dos grupos de educandos

A definição dos componentes de cada grupo foi feita a partir de uma escolha livre dos educandos e tendo como exigências que cada grupo tivesse

entre três e quatro componentes e fossem formados nove grupos sem contar o grupo dos monitores.

Finalmente, foi solicitado a todos e principalmente aos alunos monitores que, na medida do possível, antes da próxima etapa, em momento de autodidatismo, estudassem a ficha para facilitar o desenvolvimento do restante das atividades do experimento.

7.4 ETAPA EXPERIMENTAL

Inicialmente, foram distribuídas para cada grupo uma única cópia da Ficha de Experimentos a qual deveria ser devolvida ao final da atividade. Em seguida, com um projetor, foi apresentado a todos educandos o tópico 2.1 da Ficha de Experimento, referente ao procedimento de coleta de dados.

É importante observar, que a partir do instante em que o pêndulo estava com o comprimento adequado e o celular encontrava-se na mão do educando e este iniciava a contagem, o intervalo de tempo decorrido para ocorrerem quatro ciclos de movimento do pêndulo não passava de dez segundos o que resultaria em um total de menos de 7 minutos gastos para todos os 39 educandos participantes.

Porém, foi verificado que era de dois minutos o tempo médio necessário para ajustar corretamente o comprimento do pêndulo e realizar a medição para aquele comprimento. Logo, cada grupo gastaria mais de dez minutos em suas medidas. E como existia apenas um dispositivo de medida e muitos grupos para fazerem as medidas no curto intervalo de 50 minutos da aula, essa estratégia não poderia ser utilizada.

Assim, usando uma fita métrica de costureira com precisão em milímetros para medir o comprimento do pêndulo, foi estabelecido que:

i) Para o menor comprimento do pêndulo, ou seja, para $L = 0,500$ m, o grupo de monitores realizaria quatro medidas do período. Cada uma dessas medidas seria a média aritmética de quatro medições do período realizadas por cada um deles.

ii) Em seguida, outro grupo orientado pelos monitores realizaria esse procedimento para o mesmo comprimento e as médias aritméticas dos valores encontrados pelos dois grupos seria compartilhada entre todos os grupos para preenchimento da Tabela 1 da Ficha de Experimento. Assim, por exemplo, T_1 seria a média dos valores encontrados pelo primeiro educando do grupo de monitores e pelo primeiro educando do segundo grupo. Dessa forma, T_1 na verdade era resultado da média de oito medidas do período.

iii) Com orientação dos monitores dois outros grupos de educandos repetiriam o mesmo procedimento para $L = 0,600$ m. A seguir, outros dois grupos repetiriam a mesma sequência para $L = 0,700$ m e mais dois grupos para $L = 0,800$ m;

iv) Com orientação dos monitores o penúltimo grupo faria a medida para $L = 0,900$ m. Porém de forma diferente, não seriam feitas médias desses resultados como os resultados do próximo grupo, sendo esses valores diretamente compartilhados com todos para registro na Tabela 1 da Ficha de Experimento. E o último grupo repetiria exatamente esse procedimento para $L = 1,000$ m.

Com o fim das medições, a Tabela 1 da Ficha de Experimento passou a ter o aspecto mostrado na Tabela 7.1.

Desta forma, ao final da etapa experimental, cada educando em média realizou quatro medidas para o período do pêndulo. Número que pode ser considerado baixo, mas que possibilitou a todos eles uma rápida, porém

marcante, percepção do ato de medir usando uma comunicação Bluetooth entre o smartphone e o Arduino.

Tabela 7.1 – Aspecto final da Tabela 1 da Ficha de Experimento.

| L (m) | T ₁ (s) | T ₂ (s) | T ₃ (s) | T ₄ (s) | T(s) |
|-------|--------------------|--------------------|--------------------|--------------------|------|
| 0,500 | 1,42 | 1,41 | 1,42 | 1,42 | 1,42 |
| 0,600 | 1,55 | 1,55 | 1,55 | 1,55 | 1,55 |
| 0,700 | 1,69 | 1,69 | 1,68 | 1,70 | 1,69 |
| 0,800 | 1,80 | 1,80 | 1,80 | 1,81 | 1,80 |
| 0,900 | 1,90 | 1,89 | 1,89 | 1,89 | 1,89 |
| 1,000 | 2,02 | 2,01 | 2,02 | 2,01 | 2,02 |

Fonte: elaborado pelo autor.

Com a medição sendo gerenciada pelos alunos monitores, foi possível avaliar o desenrolar da atividade com mais atenção. Foi fácil perceber o grande envolvimento que todos estavam tendo com a atividade, sendo constatado o correto preenchimento da Tabela 1 por aqueles grupos que já tinham realizado a medição. E já próximo ao final da aula, foi verificado que dois desses grupos estavam tentando fazer de forma antecipada a etapa pós-experimental.

Ao final da aula, as fichas de experimentos dos grupos foram recolhidas ao mesmo tempo em que era verificado o correto preenchimento da Tabela 1. Não sendo nada encontrado fora dos valores esperados, o que era mais um indicativo do bom andamento da atividade.

Por último, foi solicitado mais uma vez a todos e principalmente aos alunos monitores que, na medida do possível, antes da próxima etapa, em momento de autodidatismo, estudassem a ficha do experimento anteriormente entregue a eles para facilitar o desenvolvimento da última etapa do experimento.

7.5 ETAPA PÓS-EXPERIMENTAL

A análise dos dados foi dividida em:

i) Com a coordenação dos monitores, os educandos realizaram o preenchimento da tabela de conversão logarítmica (Tabela 2 da Ficha de Experimento) a partir da Tabela 1 dessa mesma ficha e com uso de calculadora, ficando seu aspecto mostrado na Tabela 7.2.

Os educandos não apresentaram nenhuma dificuldade nessa etapa e uma rápida análise do preenchimento da Tabela 1 da Ficha de Experimento de cada grupo constatou que seus cálculos estavam de acordo com o esperado.

Tabela 7.2 – Aspecto final da Tabela 2 da Ficha de Experimento.

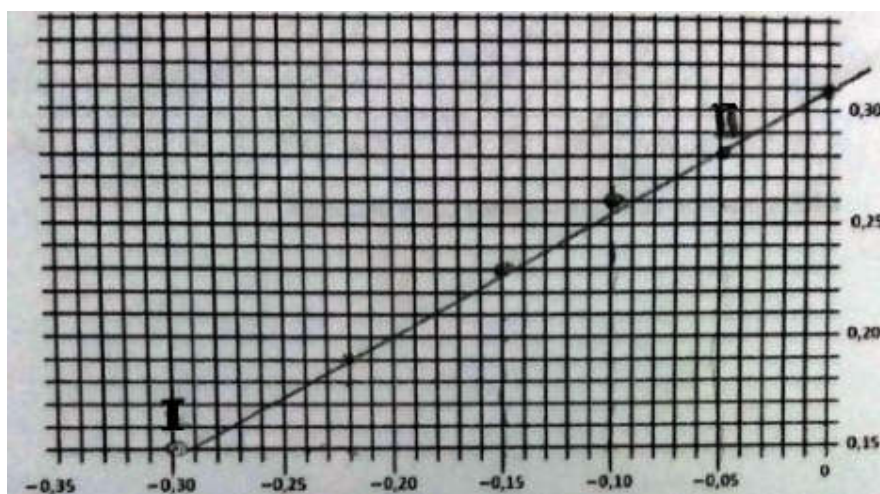
| i | L (m) | T(s) | x_i | y_i |
|-----|-------|------|-------|-------|
| 1 | 0,500 | 1,42 | -0,30 | 0,15 |
| 2 | 0,600 | 1,55 | -0,22 | 0,19 |
| 3 | 0,700 | 1,69 | -0,15 | 0,23 |
| 4 | 0,800 | 1,80 | -0,10 | 0,26 |
| 5 | 0,900 | 1,89 | -0,05 | 0,28 |
| 6 | 1,000 | 2,02 | 0,00 | 0,31 |

Fonte: elaborado pelo autor.

ii) Produção do gráfico da dependência $\log L \times \log T$ a partir do método de linearização simples com uso de lápis e régua.

Com a coordenação dos monitores, cada grupo traçou sua melhor reta. O Gráfico 7.1 apresenta a reta produzida pelo 7º grupo.

Gráfico 7.1 – Gráfico log T x log L traçado pelo 7º grupo.



Fonte: elaborado pelo autor.

Novamente os educandos não apresentaram nenhuma dificuldade nessa etapa e uma rápida observação no gráfico da ficha de cada grupo constatou que eles estavam de acordo com o esperado.

iii) Obtenção da relação experimento do período com o comprimento.

Esse foi sem dúvida o momento de maior dificuldade encontrada pelos educandos, pois muitos deles ao confrontarem seus resultados com os valores esperados perceberam discrepâncias que eles precisavam corrigir, mas como não conseguiam sozinhos fazer essa tarefa, necessitavam de orientação externa. Assim, monitores e até outros educandos que tinham terminado seus cálculos se prontificaram a ajudar.

Dois desses grupos erraram o cálculo do parâmetro b , correspondente ao coeficiente angular da reta, por simplesmente não perceberem que os valores do eixo das abscissas x serem negativos, invertendo o sinal da variação de x e gerando toda a sequência de cálculos incorretos.

É importante ressaltar aqui que esse tipo de erro é muito comum, pois o aluno sabe tratar corretamente certas situações particulares, no caso fazer a diferença entre dois valores, mas fracassa em situações próximas, a partir do momento em que certas características na aparência são modificadas, no caso, a troca de sinais dos valores a serem subtraídos.

Dois outros grupos erraram também o cálculo do parâmetro b por trocarem os valores das abscissas x pelas ordenadas y . É importante ressaltar aqui que esse tipo de erro é encontrado com muita frequência nos testes e exercícios que os educandos realizam.

iv) Comparação Teoria x Prática.

Cada grupo fez a comparação de seu resultado experimental com o resultado teórico e verificou que a hipótese $T = c.L^b$ é verdadeira e que c tem valor próximo de 2 e b próximo a 0,5.

v) Fechamento da atividade experimental.

Inicialmente, cada grupo apresentou rapidamente a todos os outros a expressão obtida para a relação período x comprimento do pêndulo, em seguida entregou sua ficha do experimento, que foi compilada na Tabela 7.3.

Para efeito de comparação e análise, na penúltima linha da Tabela 7.3, referente ao grupo RL, estão os resultados obtidos com uso de métodos matemáticos de regressão linear (RL) e calculados em uma planilha eletrônica a partir dos dados da Tabela 7.2. Já na última linha da Tabela 7.3 encontram-se os resultados referentes ao grupo Teórico e obtidos a partir da regressão linear e feitos em uma planilha eletrônica, mas com os períodos calculados teoricamente e usando o valor $9,80 \text{ m/s}^2$ para a aceleração gravitacional e o valor $3,14$ para π .

Tabela 7.3 – Relação experimental período x comprimento

| Grupo | a | b | c = 10 ^a | T = c.L ^b |
|---------|-------|-------|---------------------|------------------------------|
| 1 | 0,30 | 0,50 | 2,00 | T = 2,00.L ^{0,50} |
| 2 | 0,31 | 0,52 | 2,04 | T = 2,04.L ^{0,52} |
| 3 | 0,31 | 0,53 | 2,04 | T = 2,04.L ^{0,53} |
| 4 | 0,30 | 0,52 | 2,00 | T = 2,00.L ^{0,52} |
| 5 | 0,30 | 0,50 | 2,00 | T = 2,00.L ^{0,50} |
| 6 | 0,30 | 0,50 | 2,00 | T = 2,00.L ^{0,50} |
| 7 | 0,31 | 0,52 | 2,04 | T = 2,04.L ^{0,52} |
| 8 | 0,30 | 0,52 | 2,00 | T = 2,00.L ^{0,52} |
| 9 | 0,30 | 0,50 | 2,00 | T = 2,00.L ^{0,50} |
| 10 | 0,31 | 0,49 | 2,04 | T = 2,04.L ^{0,49} |
| RL | 0,30 | 0,50 | 2,01 | T = 2,01.L ^{0,50} |
| Teórico | 0,303 | 0,500 | 2,007 | T = 2,007.L ^{0,500} |

Fonte: elaborado pelo autor.

Em seguida os educando ficaram livres para apresentar suas impressões sobre o experimento.

Todos disseram ter gostado muito da atividade e afirmaram que seria muito bom realizar atividades similares a esta com os próximos conteúdos.

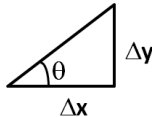
O uso de logaritmos foi apontado de forma unânime pelos educandos como a maior dificuldade encontrada durante a realização do experimento, apesar da revisão realizada em momento anterior ter sido considerada produtiva por todos eles. Informaram que o problema não estava no uso do logaritmo em si, mas em quando usar a definição e/ou suas propriedades. Isso provavelmente deve-se ao pouco tempo investido no seu estudo, uma vez que foi um conteúdo visto rapidamente no transcorrer da 1ª série do ensino médio e somente revisto no presente experimento.

Sobre a revisão de logaritmos, o grupo de monitores apesar considerá-la satisfatória, sugeriu que ela devia ter sido feita com um direcionamento maior para a física do experimento, uma vez que a professora de matemática usou exemplos gerais aplicados a matemática financeira (juros compostos), geografia (crescimento populacional) e química (decaimento radioativo). Foi então esclarecido que isso ocorreu devido a falta de tempo hábil de se fazer um planejamento integrado, tendo ele sido feito apenas pela professora de matemática. E isso seria revisto em futuras atividades.

Dois grupos que inicialmente erraram o cálculo do coeficiente angular devido a troca de sinais do eixo das abscissas x e precisaram do auxílio dos monitores para chegar ao valor correto, afirmaram que o erro foi cometido sobretudo devido ao fato deles terem focado sua atenção nos passos seguintes, que envolviam o uso de logaritmo e por isso não tiveram o devido cuidado na manipulação dos valores de x . Assim, para diminuir a chance de ocorrerem erros desse tipo, sugeriram colocar um aviso imediatamente acima da expressão do coeficiente angular b como mostra a primeira linha do Quadro 7.1.

Já os dois grupos que inverteram os valores das abscissas x e ordenas y na divisão, também afirmaram que o erro foi cometido pelo mesmo motivo dos dois grupos anteriores e auxiliados pelos monitores sugeriram que fosse realizada a substituição contida na segunda linha do Quadro 7.1.

Quadro 7.1 – Sugestões de alteração da Ficha de Experimentos.

| | |
|---|--|
| Atenção: os valores do eixo horizontal x são <u>negativos</u> | |
| $b = \frac{y_{II} - y_I}{x_{II} - x_I} = \underline{\hspace{2cm}} = \underline{\hspace{2cm}}$ | |
|  | $b = \operatorname{tg} \theta = \frac{\Delta y}{\Delta x} = \frac{y_{II} - y_I}{x_{II} - x_I} = \underline{\hspace{2cm}} = \underline{\hspace{2cm}}$ |

Fonte: elaborado pelo autor.

Essas sugestões de alterar o material instrucional, o que está dentro da fundamentação construtivista, mostram como detalhes sutis podem ser tão importantes para orientar o aprendiz e evitar que ele cometa erros tidos como simples. E com certeza essas sugestões serão consideradas na preparação do material usado em futuras atividades.

Não havendo mais nenhuma ponderação por parte dos educandos, e após uma análise de todas as fichas dos grupos em que foi verificado o correto preenchimento de todas, deu-se por concluído o fechamento da atividade.

vi) Coleta de comentários dos educandos.

Dos vários comentários particulares feitos pelos educandos vale a pena destacar:

- Um primeiro educando, que tem uma atitude distante e desinteressada em todas as aulas e apresenta baixo rendimento em quase todas as atividades fez o comentário: “Não gosto de Física, mas gostei muito dessa experiência, principalmente pelo uso da comunicação Bluetooth do Arduino com o celular. Só não gostei muito de trabalhar com logaritmos. Mesmo assim deu pra entender um pouco quando os monitores me ajudaram. Gostaria que todas as aulas fossem parecidas com a de hoje”.

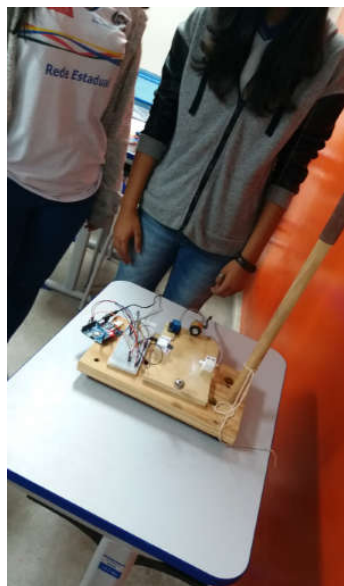
- Um segundo educando, que demonstra quase sempre uma atitude interessada nas atividades e possui bom rendimento fez o comentário: “Trabalhar com logaritmo foi meio difícil, mas no final, com a ajuda do monitor, consegui entender. O restante da atividade foi muito legal, por isso queria participar de outras atividades desse tipo”.

- Um terceiro educando, que demonstra muito interesse em fazer todas as atividades de Física, sempre muito participativo, sendo um dos monitores tanto no experimento do pêndulo quanto em outras atividades da disciplina e com rendimento excelente fez o comentário: “Achei tudo muito bom. Fiquei surpreso, pois não sabia que poderia usar logaritmos em um experimento Físico. Gostei muito de usar o Arduino com o celular. Gostei também de poder ajudar meus colegas. Gostaria de ter outras experiências usando Arduino - Bluetooth - celular”.

Esses relatos corroboram o que muitos autores afirmam sobre os benefícios dos experimentos físicos, do uso do Arduino e do smartphone, que quando usados com um ambiente dinâmico e colaborativo produzem intenso aprendizado nos educandos.

7.6 REGISTRO FOTOGRÁFICO DA APLICAÇÃO DA PROPOSTA

Figura 7.1 – Interface de medição.



Fonte: elaborado pelo autor.

Figura 7.2 – Alunas realizando a medição.



Fonte: elaborado pelo autor.

Figura 7.3 – Alunos realizando a medição.



Fonte: elaborado pelo autor.

8 CONSIDERAÇÕES FINAIS

Uma diversidade crescente de microcontroladores, sensores, atuadores e dispositivos de comunicação com custo cada vez menor e maior facilidade de aquisição no mercado especializado simplificam o desenvolvimento de experimentos de aquisição de dados e controle de dispositivos com Arduino.

Fazer uso dessa tecnologia integrada a um smartphone é uma excelente forma de dinamizar as aulas, atrair a atenção dos educandos e tornar as aulas de Física mais produtivas, pois permite acesso rápido a uma grande quantidade de dados que podem facilmente ser processados, convertidos em gráficos e analisados. Mas ela ainda é pouco utilizada, pois boa parte dos professores não tem formação adequada nessa área e não se sente preparado para fazer seu uso. Nesse sentido, como primeiro objetivo do presente trabalho, os capítulos quatro e cinco apresentaram um guia de estudo para desenvolver esse tipo de experimento. É importante observar que foi apresentado apenas o início de um estudo, um pequeno recorte da totalidade, mas que é a base para desenvolvimento de qualquer projeto que use essa tecnologia.

Outro ponto a se destacar é que pode ser interessante ou até mesmo necessário o desenvolvimento de um aplicativo customizado para substituir o Serial Bluetooth Terminal. Dependendo do tempo disponível e do interesse de aprendizagem, existem diversas possibilidades. Uma delas, o MIT App Inventor, usa uma interface gráfica de desenvolvimento, sendo mais adequado para aqueles com pouca ou nenhuma experiência em programação. Outra possibilidade, o Android Studio, exige um conhecimento maior em programação e é indicado para um desenvolvimento mais profissional.

As aulas com experimentos em geral também necessitam de um maior tempo de preparo e aplicação quando comparado com as aulas teóricas. Assim, é necessário fazer um planejamento adequado da coleta dos materiais necessários, da montagem propriamente dita e por fim da sua aplicação. Por

isso, os objetivos a serem alcançados, o tempo disponível, a disponibilidade de matérias e as habilidades e conhecimentos dos educandos são critérios fundamentais para definir o direcionamento (demonstração, verificação e investigação), a ênfase matemática (qualitativa ou quantitativa) e o grau de liberdade (mais ou menos estruturado).

Uma boa estratégia é fazer uso inicialmente de atividades mais estruturadas e mais simples, sempre se privilegiando uma metodologia que induza aos educandos a terem uma participação ativa e solidária. Com o aumento natural de seus conhecimentos e habilidades a partir de cada projeto realizado, deve-se construir com eles experimentos cada vez menos estruturados e estimular que desenvolvam projetos cada vez mais autônomos e com maior interdisciplinaridade.

Como segundo objetivo do presente trabalho, numa aplicação direta do guia de estudo visto nos capítulos 4 e 5, foi apresentado no capítulo 6 o desenvolvimento de um aparato de medição experimental para obter a relação matemática entre o comprimento do pêndulo simples e seu período de movimento.

Esse aparato foi usado em um experimento aplicado de acordo com a abordagem construtivista proposta por Rosa (2012), sendo dividido em três momentos: pré-experimental, experimental e pós-experimental. Apresentou, como destacado nos pressupostos construtivistas, atividades predominantemente realizadas em grupo, com incentivo ao diálogo professor/educando e educando/educando em um ambiente que estimulava a eles terem uma postura ativa e colaborativa.

Um indicador muito satisfatório observado durante a realização desse experimento é que foi marcante o alto grau de interesse e participação dos quase quarenta educandos presentes, inclusive daqueles que, em outras atividades, normalmente apresentavam uma atitude distante e desinteressada,

mas que durante a realização do experimento fizeram questão de entender a lógica de funcionamento do aparato, de ter maiores detalhes sobre a conexão Arduino/smartphone, de fazer a medição do período pessoalmente, apesar do curto tempo disponibilizado e finalmente de preencher corretamente a ficha do experimento. Por isso, durante o fechamento da atividade na fase pós-experimental, todos apresentaram um bom domínio de competências e habilidades relacionadas tanto ao conteúdo específico da física do pêndulo quanto da dinâmica ocorrida durante a realização do experimento.

Outro ponto a se destacar, foi como os alunos monitores conduziram as atividades, induzindo o restante dos educandos a terem também uma atitude colaborativa e de ajuda principalmente nos momentos de maior dificuldade para alguns colegas.

O uso de logaritmos foi apontado de forma unânime pelos educandos como a maior dificuldade encontrada durante a realização do experimento, apesar da revisão realizada em momento anterior ter sido considerada produtiva por todos eles. Por isso, em outra situação, talvez uma mudança na hipótese possa ser interessante. Pode-se tomar a hipótese $L = kT^2$ e então usar as transformações $x = T^2$ e $y = L$ obtendo-se $y = kx$. Assim a perda com a diminuição de generalização da hipótese seria compensada com a simplificação da análise do experimento, tornando a atividade mais acessível aos educandos uma vez que as funções polinomiais do primeiro e segundo grau são de uso muito mais comum entre eles.

De forma similar, apenas alterando os objetivos e a metodologia do experimento e sem fazer nenhuma modificação no aparato, é possível, por exemplo, calcular o valor da gravidade local ou analisar a dependência do período do pêndulo com outros parâmetros, tais como a massa da esfera, o material que a constitui, seu raio, seu volume, sua área frontal, etc.

Já com pequenas modificações no aparato é possível, por exemplo, estudar o movimento vertical de queda livre e para isso basta retirar o fio do pêndulo, colocar o eletroímã num ponto acima do LED e do LDR de modo que quando a esfera for liberada passe entre eles. Evidentemente é preciso fazer um ajuste na programação do Arduino. Também é possível estudar o movimento de um corpo no plano inclinado. Para tanto, basta fixar um prego em um pequeno bloco de madeira de modo que ele possa ser atraído pelo eletroímã e permaneça em repouso no topo de uma tábua plana e inclinada. Ao liberar o bloco para mover-se, ele deve passar entre o LED e o LDR.

A utilização de sensores como LDR, infravermelho IR, pressão, temperatura, som, ultra-som, temperatura, vazão, corrente elétrica, campo magnético, velocidade e aceleração permitem fazer aquisição de dados em todas as áreas de estudo da física do ensino médio. E se combinados com atuadores como motores, aquecedores elétricos, emissores infravermelho, entre outros e conectados por USB, ethernet, Bluetooth, wifi ou GSM permitem o desenvolvimento de uma gigantesca quantidade de projetos com alto potencial pedagógico.

Essa integração entre tecnologia, teoria e experimento é capaz de contribuir substancialmente na aprendizagem dos educandos, preparando-os para lidar com as mudanças e avanços das tecnologias e que geram continuamente novas exigências para a sociedade moderna. Dentre esses avanços tecnológicos está o smartphone, que adquiriu um papel muito importante na atualidade e por isso, não faz mais sentido que fique fora do processo de ensino e aprendizagem.

Os jovens estudantes se adaptam às novas tecnologias de forma muito rápida, às vezes com boas práticas e outras vezes nem tanto. E o uso adequado dessas inovações precisa da orientação apropriada feita pelo professor, mediador que deve ser. Não é necessário que ele seja um especialista em

tecnologia, mas o educador tem que se preparar para fazer o melhor uso pedagógico possível dessas tecnologias.

É necessário ressaltar que as tecnologias são apenas recursos e seu uso adequado pode facilitar e estimular em muito a aprendizagem dos educandos, mas é muito importante lembrar que a ação do professor é fundamental nesse processo, sendo ele quem deve definir quais tecnologias usar e quando e como usá-las.

Por fim, esse trabalho mostra que o uso de uma metodologia que crie um ambiente em que o educando seja induzido a ter uma atitude ativa e colaborativa, em conjunto com tecnologias como o smartphone e a placa robótica Arduino, pode se tornar uma ferramenta muito poderosa para o desenvolvimento de experimentos físicos com grande potencial de aprendizagem.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, C. E.; PEREIRA, M. M. O computador como cronômetro, Revista Brasileira de Ensino de Física, v. 34, n. 3, art. 3303, 2012. Disponível em: <https://www.scielo.br/pdf/rbef/v34n3/a03v34n3.pdf>. Acesso 20 jul. 2019.

ANASTASIOU, Léa das Graças Camargos; ALVES, Leonir Pessate. Estratégias de ensinagem. In: ANASTASIOU, Léa das Graças Camargos; ALVES, Leonir Pessate. (Orgs.). Processos de ensinagem na universidade. Pressupostos para as estratégias de trabalho em aula. 3. ed. Joinville: Univille, 2007.

ARAÚJO, M. S. T. de; ABIB, M. L. V. dos S. Atividades Experimentais no Ensino de Física: Diferentes Enfoques, Diferentes Finalidades. Revista Brasileira de Ensino de Física, São Paulo, v. 25, n. 2, p. 176-194, jun. 2003. Trimestral. Disponível em <https://www.scielo.br/pdf/rbef/v25n2/a07v25n2.pdf>. Acesso 20 jul. 2019.

ARDUINO BLINK. Blink. Disponível em: <https://www.arduino.cc/en/tutorial/blink>. Acesso em: 01 abr. 2019.

ARDUINO BUTTON. ARDUINO TUTORIAL. Button. 2015. Disponível em: <https://www.arduino.cc/en/tutorial/button>. Acesso em: 20 jul. 2019.

ARDUINO CLONES. Send in the clones. Disponível em: <https://blog.arduino.cc/2013/07/10/send-in-the-clones/>. Acesso em: 05 mar. 2019.

ARDUINO CONSTANTES. Arduino Constantes. Disponível em: <https://www.arduino.cc/reference/en/language/variables/constants/constants/>. Acesso em: 06 abr. 2019.

ARDUINO DIGITAL READ. digitalRead(). Disponível em: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>. Acesso em: 10 jun. 2019.

ARDUINO ELSE. Else. Disponível em: <https://www.arduino.cc/reference/en/language/structure/control-structure/else/>. Acesso em: 16 jun. 2019.

ARDUINO HOME. Arduino Home. Disponível em: <https://www.arduino.cc>. Acesso em: 03 fev. 2019.

ARDUINO IF. IF. Disponível em: <https://www.arduino.cc/reference/en/language/structure/control-structure/if/>. Acesso em: 16 jun. 2019.

ARDUINO INTRODUÇÃO. Arduino Introduction. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 10 fev. 2019.

ARDUINO LDR & LED. LDR & LED Light. Disponível em:
https://create.arduino.cc/projecthub/Kenpoca_Dias/ldr-led-light-1147c3. Acesso em: 20 jul. 2019.

ARDUINO LINGUAGEM. Arduino Language Reference. Disponível em:
<https://www.arduino.cc/reference/en/>. Acesso em: 06 abr. 2019.

ARDUINO PINOS ANALÓGICOS. Analog Input Pins. Disponível em:
<https://www.arduino.cc/en/Tutorial/AnalogInputPins>. Acesso em: 09 ago. 2019.

ARDUINO PINOS DIGITAIS. Digital Pins. Disponível em:
<https://www.arduino.cc/en/Tutorial/DigitalPins>. Acesso em: 10 ago. 2019.

ARDUINO PRODUTOS. Arduino Products. Disponível em:
<https://www.arduino.cc/en/Main/Products>. Acesso em: 04 abr. 2019.

ARDUINO PUSHBUTTON. Pushbutton. Disponível em:
<https://www.arduino.cc/en/Tutorial/Pushbutton/>. Acesso em: 20 ago. 2019.

ARDUINO READ SERIAL. Digital Read Serial. Disponível em:
<https://www.arduino.cc/en/Tutorial/DigitalReadSerial>. Acesso em: 25 ago. 2019.

ARDUINO SERIAL. Serial. Disponível em:
<https://www.arduino.cc/reference/en/language/functions/communication/serial/>. Acesso em: 20 ago. 2019.

ARDUINO SOFTWARE. Arduino Software. Disponível em:
<https://www.arduino.cc/en/main/software>. Acesso em: 20 jan. 2019.

ARDUINO START. Getting Started with Arduino and Genuino products. Disponível em: <https://www.arduino.cc/en/Guide/HomePage>. Acesso em: 04 abr. 2019.

ARDUINO UNO. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>. Acesso em: 04 abr. 2019.

ARGENTO, Heloisa. Teoria Construtivista. Disponível em:
<http://penta3.ufrgs.br/midiasedu/modulo11/etapa2/construtivismo.pdf>. Acesso em: 25 mar. 2020.

ATMEGA. ATmega808/1608/3208/4808 – 32-Pin. Disponível em:
<http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega808-1608-3208-4808-32-Pin-40002017C.pdf>. Acesso em: 13 ago. 2019.

AUTOCOREROBOTICA. Funcionamento dos resistores de pull-up e pull-down. Disponível em: <https://autocorerobotica.blog.br/resistores-pull-up-e-pull-down-como-funcionam/>. Acesso em: 04 abr. 2019

AZEVEDO, Hernani Luiz; MONTEIRO JÚNIOR, Francisco Nairon; SANTOS, Thiago Pereira dos; CARLOS, Jairo Gonçalves; TANCREDO, Bruno Nogueira. O

uso do experimento no ensino da Física: tendências a partir do levantamento dos artigos em periódicos da área no Brasil. In: VII Encontro Nacional de Pesquisa em Educação em Ciências, 7., 2009, Florianópolis. Anais... Florianópolis, 2009. Disponível em:

<http://posgrad.fae.ufmg.br/posgrad/viienpec/pdfs/1067.pdf>. Acesso em: 24 mar. 2020.

BANZI, Massimo. Getting Started with Arduino. Second Edition. U.S.A: O'Reilly Media, 2011.

BRAGA, C. Newton. Eletrônica básica para mecatrônica. São Paulo: Saber, 2005.

BECKER, F. O que é construtivismo. Idéias. São Paulo: FDE, n.20, p.87-93, 1993. Disponível em: http://www.crmariocovas.sp.gov.br/pdf/ideias_20_p087-093_c.pdf. Acesso em: 07 nov. 2020.

BRAGA, C. Newton. LDR (ALM332) Disponível em:

<https://www.newtoncbraga.com.br/index.php/almanaque-tecnologico/201-l/7547-ldr-alm332>. Acesso em: 20 mar. 2019.

BRASIL. Lei de Diretrizes e Bases da Educação Nacional. Senado Federal, 2005. Disponível em: <https://www2.senado.leg.br/bdsf/bitstream/handle/id/70320/65.pdf>. Acesso em: 20 nov. 2020.

BRASIL. PCN+: Orientações Curriculares para o Ensino Médio: Ciências da Natureza, Matemática e suas Tecnologias. Brasília: MEC, 2000. Disponível em: <http://portal.mec.gov.br/seb/arquivos/pdf/CienciasNatureza.pdf>. Acesso em: 23 mar. 2019.

CAMARGO, A. C. V. C.; FARIA, M. A. Avaliação: concepções e reflexão. Revista Eletrônica Saberes da Educação, São Roque, v. 2, n. 1, p. 1-13, 2011. <http://docs.uninove.br/arte/fac/publicacoes/pdf/v2-n1-2011/Aline.pdf>. Acesso em: 25 nov. 2020.

CARVALHAES, Claudio G. I; SUPPES, Patrick I. O cálculo de alta precisão do período do pêndulo simples. Rev. Bras. Ensino Física. vol. 31 n°. 2. pp.2701.1-2701.6. São Paulo: Abril/Junho 2009. Disponível em: <https://www.scielo.br/pdf/rbef/v31n2/16.pdf>. Acesso em: 20 mar. 2019.

CARVALHO, A. M. P.; VANUCCHI, A. I; BARROS, M. A.; GONÇALVES, M. E. R.; REY, R. C. Ciências no Ensino Fundamental – O Conhecimento Físico. São Paulo: Scipione, 1998. - (Pensamento e ação no magistério). Disponível em: https://edisciplinas.usp.br/pluginfile.php/167463/mod_resource/content/1/Texto%203%20-%20Cap.%202%20e%203%20do%20livro%20Ci%20C3%AAncias%20no%20Ensino%20Fundamental.pdf. Acesso em: 24 mar. 2020.

CARVALHO, Anna Maria Pessoa. As práticas experimentais no ensino de Física. In: CARVALHO, Anna Maria Pessoa de et AL. (Orgs.) “Ensino de Física” Coleção Ideias em Ação. São Paulo: Cengage Learning, 2010, página 53 até 78.

CAVALCANTE, M. A.; TAVOLARO, C. R. C. e MOLISANI, E. Física com Arduino para iniciantes. Revista Brasileira de ensino de Física, v. 33, n. 4, 4503 (2011). Disponível em: <http://www.sbfisica.org.br/rbef/pdf/334503.Pdf>. Acesso em: 23 mar. 2020.

CHAKUR, Cilene Ribeiro de Sá Leite. A desconstrução do Construtivismo na educação: crenças e equívocos de professores, autores e críticos. São Paulo: Editora Unesp Digital, 2014. Disponível em: <https://static.scielo.org/scielobooks/hf4w9/pdf/chakur-9788568334485.pdf>. Acesso em: 25 nov. 2020.

CUSTÓDIO, J. F.; ALVES FILHO, J. de P.; CLEMENT, L.; RICHETTI, G. P.; FERREIRA, G. K. (2013). Práticas didáticas construtivistas: critérios de análise e caracterização. Tecné, Episteme y Didaxis, v. 1, n. 33, p.11-35, jan. 2013. Disponível em: <http://www.scielo.org.co/pdf/ted/n33/n33a01.pdf>. Acesso em: 5 nov. 2020

ENGINEERS. Stepper Motor Speed and Direction Control Using Arduino and Bluetooth HC-06 Module through an Android App. 2019. Disponível em: <https://www.engineersgarage.com/arduino/stepper-motor-controlled-with-hc06-bluetooth-module/>. Acesso em: 10 fev. 2020.

ELECTRICARENA. Arduino Onboard embedded LED's Their Function and Color. Disponível em <http://electricarena.blogspot.com/2015/01/arduino-onboard-embedded-leds-function.html>. Acesso em: 08 maio 2019.

ELETROGATE. Módulos Bluetooth HC05 e HC06 para comunicação com dispositivos móveis com Arduino. Disponível em: <https://blog.eletrogate.com/modulos-bluetooth-hc05-e-hc06-para-comunicacao-com-dispositivos-moveis-com-arduino/>. Acesso em: 17 ago. 2019.

EL-HANI, Charbel Nino; BIZZO, Nelio Marco Vincenzo. Formas de Construtivismo: Mudança Conceitual e Construtivismo Contextual. Ens. Pesqui. Educ. Ciênc. (Belo Horizonte) [online]. 2002, vol.4, n.1, pp.40-64. ISSN 1983-2117. Disponível em: <https://doi.org/10.1590/1983-21172002040104>. Acesso em: 01 maio 2020.

EMBARCADOS LÂMPADA. Acionamento de uma lâmpada com Arduino. Disponível em: <https://www.embarcados.com.br/acionamento-de-uma-lampada-com-arduino/>. Acesso em: 15 maio 2019.

EMBARCADOS SERIAL. Arduino - Comunicação Serial. Disponível em: <https://www.embarcados.com.br/arduino-comunicacao-serial/>. Acesso em: 10 set. 2019.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. Arduino em ação. São Paulo: Novatec, 2013. Tradução de Camila Paduan.

GOMES, Luciano Carvalhais; BELLINI, Luzia Marta. Uma revisão sobre aspectos fundamentais da teoria de Piaget: possíveis implicações para o ensino de física. Rev. Bras. Ensino Fís., São Paulo, v. 31, n. 2, p. 2301.1-2301.10, June 2009. Disponível em: <https://www.scielo.br/pdf/rbef/v31n2/02.pdf>. Acesso em 15 nov. 2020.

GUAITOLINI JUNIOR, Judismar Tadeu; RAMOS, Gabryel Silva; SILVA, Samir Lacerda da; GAMA, Aline Costalonga. Avaliação do módulo da aceleração da gravidade com Arduino. Caderno Brasileiro de Ensino de Física, v. 33, n. 2, p. 619-640, ago. 2016. Disponível em: <https://periodicos.ufsc.br/index.php/fisica/article/download/2175-7941.2016v33n2p619/32333>. Acesso em: 05 abr. 2020.

GUEDDES, Mark. Manual de projetos Arduino. São Paulo: Novatec, 2017.

FILIFELOP. Como usar o Arduino Bluetooth HC-05 em modo mestre. Disponível em: <https://www.filipeflop.com/blog/tutorial-arduino-bluetooth-hc-05-mestre/>. Acesso em: 10 fev. 2019.

HAAG, Rafael; ARAUJO, Ives Solano; VEIT, Eliane Angela. Por Que e Como Introduzir a Aquisição Automática de Dados no Laboratório Didático de Física? Física na Escola, v. 6, n. 1, p.69-74, maio 2005. Disponível em: <https://lume.ufrgs.br/handle/10183/116432>. Acesso em: 23 mar. 2020.

KAMEI, Camila A. N.; Barbosa, Lucas F. L. Introdução a Programação Programando com Intel Galileo. Experimento 1. Recife: UFPE - Centro de Informática-Grupo de Engenharia da Computação. Disponível em: https://www.cin.ufpe.br/~lflb/Tutorial%20%20-%20conceitos_basicos_galileo.pdf. Acesso em: 14 set. 2019.

LABURÚ, C. E., CARVALHO, M. de. (2011). Controvérsias construtivistas e pluralismo metodológico no ensino de ciências naturais. Revista Brasileira De Pesquisa Em Educação Em Ciências, 1(1). Disponível em: <https://periodicos.ufmg.br/index.php/rbpec/article/view/4184>. Acesso em 15 nov. 2020.

LEÃO, Denise Maria Maciel. Paradigmas Contemporâneos de Educação: Escola Tradicional e Escola Construtivista. Cadernos de Pesquisa, nº 107, p. 187-206, julho/1999. Disponível em: <https://www.scielo.br/pdf/cp/n107/n107a08.pdf>. Acesso em: 05 nov. 2020.

LED. PL16 Series 5mm (T1¾) Ultra Bright White LED. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/292012/P-TEC/PL16C-WCW13Z.html>. Acesso em: 20 fev. 2019.

LOPES, P. A., PIMENTA, C. C. C., O uso do celular em sala de aula como ferramenta pedagógica: Benefícios e desafios. Revista Cadernos de Estudos e Pesquisas na Educação básica, Recife, v. 3, n. 1, p. 52 - 66, 2017 . CAP UFPE. Disponível em: <https://periodicos.ufpe.br/revistas/cadernoscap/article/view/229430>. Acesso em: 23 mar. 2020.

MARQUES, Gil da Costa. Mecânica Clássica para Professores. São Paulo: Editora Universidade de São Paulo, 2014.

MARTINAZZO, C. A.; TRENTIN, D. S.; FERRARI, D.; PIAIA, M. M.; Arduino: Uma tecnologia no ensino de física. Perspectiva, Erechim. v. 38, n.143, p. 21-30, setembro/2014. Disponível em: http://www.uricer.edu.br/site/pdfs/perspectiva/143_430.pdf. Acesso em: 23 mar. 2020.

MCROBERTS, Michael. Arduino Básico. São Paulo: Novatec, 2011.

MIRANDA, Maria Irene. Conceitos centrais da teoria de vygotsky e a prática pedagógica. Ensino em Re-Vista, 13(1): 7-28, jul.04/jul.05. Disponível em https://www.researchgate.net/publication/268329076_Conceitos_centrais_da_teor%C3%ADa_de_Vygotsky_e_a_pr%C3%A1tica_pedag%C3%B3gica/fulltext/546d9bef0cf26e95bc3cbaf3/Conceitos-centrais-da-teoria-de-Vygotsky-e-a-pratica-pedagogica.pdf. Acesso em: 15 nov. 2020.

MONK, Simon. Projetos com Arduino e Android: use seu smartphone ou tablet para controlar o arduino. Porto Alegre: Bookman, 2014.

MOREIRA, Marco Antonio. Teorias da aprendizagem. São Paulo: EPU, 1999.

MOREIRA, M.A. & GONÇALVES, E.S. Laboratório estruturado versus não estruturado: um estudo comparativo em um curso individualizado. Rev. Bras. Fis., 10(2): 367-81, 1980. Disponível em: <http://www.sbfisica.org.br/bjp/download/v10/v10a27.pdf>. Acesso em: 24 mai. 2019.

MORICH, Kai. Disponível em: <https://github.com/kai-morich/SimpleBluetoothTerminal>. Acesso em: 20 fev. 2019.

NETTO, Luiz Ferraz. Circuitos Eletrônicos. Disponível em: <http://netsaber.com.br/apostilas/apostilas/38.doc>. Acesso em: 14 ago. 2019.

OLIVEIRA, Cláudio Luís Vieira. Aprenda Arduino - Uma abordagem prática. – Duque de Caixas: Katzen Editora, 2018.

OLIVEIRA, J. R. S. Contribuições e abordagens das atividades experimentais no ensino de ciências: reunindo elementos para a prática docente. Acta Scientiae. v.12, n.1, p. 139-156, Jan./Jun. 2010. Disponível em: <http://www.periodicos.ulbra.br/index.php/acta/article/view/31/28>. Acesso em: 23 mar. 2020.

PALMA, Daniel A.; OLIVEIRA, Maria Stella N.; Jesus, Vitor L. B. de. As representações integrais e o ensino de matemática nas licenciaturas em física: Um exemplo simples baseado em observações experimentais. Revista Perspectivas da Ciência e Tecnologia vol. 1, nº. 1. Rio de Janeiro: jan-jun 2009.

PAMBOUKIAN, Sergio Vicente Denser; PEDROTTI, Jairo José. Desenvolvimento de Instrumentação Analítica e Software para Controle, Aquisição e Tratamento de Dados Analíticos. Revista Mackenzie de Engenharia e Computação, v. 12, n. 1, p. 32-43, 2012. Disponível em: <http://editorarevistas.mackenzie.br/index.php/rmec/article/view/3783/3802>. Acesso em: 29 maio 2020.

PERUZZO, Jucimar. A Física Através de Experimentos: Mecânica. V.I / Jucimar Peruzzo. Irani (SC): 2013. 354p. Disponível em: <http://files.superleomatematica.webnode.com/200000032-0b4d70d420/A%20f%C3%ADsica%20atrav%C3%A9s%20de%20experimentos%20-%20Vol.%20I%20-%20Mec%C3%A2nica.pdf>. Acesso em: 29 mar. 2020.

RAGUGNETTI, T.T.A.; LABURÚ, C.E.; ARRIGO, V. Considerações Sobre o Construtivismo, a Educação Científica e o Ensino Crítico. XI Encontro Nacional de Pesquisa em Educação em Ciências – XI ENPEC, 2017. Disponível em: <http://www.abrapecnet.org.br/enpec/xi-enpec/anais/resumos/R1523-1.pdf>. Acesso em: 05 nov. 2020.

ROGGIA, Leandro. Automação industrial / Leandro Roggia, Rodrigo Cardozo Fuentes. – Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2016.

ROSA, Cleci T. Werner da; ROSA, Álvaro Becker da. Aulas experimentais na perspectiva construtivista. Física na Escola, v. 13, n. 1, 2012. Disponível em: <http://www1.fisica.org.br/fne/phocadownload/Vol13-Num1/a021.pdf>. Acesso em: 23 mar. 2020.

SALSA, Ivone da Silva. A importância do erro do aluno em processos de ensino e de aprendizagem. REMATEC/Ano 12/n. 26/set.-dez. 2017, p. 86 – 99. Disponível em: <file:///C:/Users/jbosc/Downloads/112-Texto%20do%20artigo-299-3-10-20180324.pdf>. Acesso em: 15 dez. 2020.

SANCHIS, Isabelle de Paiva e MAHFOUD, Miguel – Construtivismo: desdobramentos teóricos e no campo da educação. Revista Eletrônica de Educação. São Carlos, SP: UFSCar, v.4, no. 1, p. 18-33, mai. 2010. Disponível em <http://www.reveduc.ufscar.br>.

SERWAY, Raymond A.; JEWETT JR, John W. Física para cientistas e engenheiros, volume 2: oscilações, ondas e termodinâmica. – São Paulo: Cengage Learning, 2011.

SILVA, Eleonora Maria Diniz da. A Virtude do Erro: uma visão construtiva

da avaliação. Estudos em Avaliação Educacional, v. 19, n. 39, jan./abr. 2008. Disponível em: <http://www.fcc.org.br/pesquisa/publicacoes/eae/arquivos/1420/1420.pdf>. Acesso em: 05 nov. 2020.

SILVA, Ione de Cássia Soares da et al. As Novas Tecnologias e aprendizagem: desafios enfrentados pelo professor na sala de aula. Revista Em Debate (UFSC), Florianópolis, v. 16, p. 107-123, 2016. ISSN 1980-3532. Disponível em: <https://periodicos.ufsc.br/index.php/emdebate/article/view/1980-3532.2016n15p107/33788>. Acesso em: 20 maio 2019.

SOARES, Reginaldo R.; BORGES, Paulo de F. O plano inclinado de Galileu: uma medida manual e uma medida com aquisição automática de dados. Revista Brasileira de Ensino de Física, São Paulo, v. 32, n. 2, p.2501.1-2501.11, abr. 2010. disponível em: <https://www.scielo.br/pdf/rbef/v32n2/v32n2a12.pdf>. Acesso em: 25 mar. 2020.

SOUZA, Anderson R. de; PAIXÃO, Alexsander C.; Uzêda, Diego D.; M.A. DIAS, Marco A; DUARTE, Sérgio e Amorim Helio S. de. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. Revista Brasileira de Ensino de Física v. 33, n. 1, 1702 (2011). Disponível em: <https://www.scielo.br/pdf/rbef/v33n1/26.pdf>. Acesso em: 23 mar. 2020

SOUZA, Solange Jobim; KRAMER, Sonia. O debate Piaget/Vygotsky e as políticas educacionais. Cadernos de Pesquisas (77), p 69-80, mai. 1991. Disponível em: <http://publicacoes.fcc.org.br/ojs/index.php/cp/article/download/1044/1052>. Acesso em: 05 nov. 2020.

TELECO. Bluetooth: O que é? Disponível em: https://www.teleco.com.br/tutoriais/tutorialblue/pagina_1.asp. Acesso em: 1 set. 2019.

TUTORIALSPPOINT. Arduino – Board Description. Disponível em: https://www.tutorialspoint.com/arduino/arduino_board_description.htm. Acesso em: 04 abr. 2019.

UNESCO. O Futuro da Aprendizagem Móvel: Implicações para planejadores e gestores de políticas. 2014. Disponível em: <https://unesdoc.unesco.org/ark:/48223/pf0000228074>. Acesso em: 23 mar. 2020.

UNESP. Eletroímã. Disponível em <http://www2.fc.unesp.br/experimentosdefisica/ele11.htm>. Acesso em: 14 set. 2019.

USP. Aplicações do 1º fenômeno eletromagnético. Disponível em: http://efisica.if.usp.br/eletricidade/basico/campo_corrente/aplic_prim_fenom_eletromag/. Acesso em: 14 set. 2019.

VALADARES, J. (2001). Estratégias Construtivistas e Investigativas no Ensino das Ciências. Conferência proferida no Encontro “O Ensino das Ciências no Âmbito dos Novos Programas”, na Faculdade de Engenharia da Universidade do Porto. Disponível em

https://www.dge.mec.pt/sites/default/files/Secundario/Documentos/Programas/CE_Programa/publicacoes_estrat_const.pdf. Acesso em: 05 nov. 2020.

VIDADESILÍCIO LDR. Sensor de Luz – Aprendendo a usar o LDR com Arduino. Disponível em: <https://portal.vidadesilicio.com.br/sensor-de-luz-com-ldr/>. Acesso em 15 set. 2019.

VIDADESILÍCIO HC06. Módulo Bluetooth HC-05 e HC-06 – Acionando um módulo relé via Bluetooth. Disponível em: <https://portal.vidadesilicio.com.br/modulo-bluetooth-hc-05-e-hc-06/>. Acesso em: 27 fev. 2020.

VIDADESILÍCIO RELÉ. Módulo relé – Acionando cargas com Arduino. Disponível em: <https://portal.vidadesilicio.com.br/modulo-rele-com-arduino/>. Acesso em: 15 maio 2019.

WESENDONK, Fernanda Sauzem; PRADO, Letícia do. Atividade didática baseada em experimento: Discutindo a implementação de uma proposta investigativa para o Ensino de Física. *Experiências em Ensino de Ciências*, Bauru, v. 10, n. 1, p. 54-80, abr. 2015. Disponível em: http://if.ufmt.br/eenci/artigos/Artigo_ID265/v10_n1_a2015.pdf. Acesso em: 23 mar. 2020.

WIRED. Build It. Share It. Profit. Can Open Source Hardware Work? . Disponível em: <https://www.wired.com/2008/10/ff-openmanufacturing/>. Acesso em: 09 fev. 2019.

WIRING. Disponível em: <http://wiring.org.co/>. Acesso em: 06 maio 2019.

WUSTL. Bluetooth Module (HC-06) + Arduino. Disponível em: [https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_\(HC-06\)_%2B_Arduino&oldid=12261](https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_(HC-06)_%2B_Arduino&oldid=12261). Acesso em: 10 mar. 2019.

APÊNDICE A – Sketch PENDULO_SIMPLES

```

int pinoRELE = 11; // pino de conexão do Rele
int pinoLED = 12; // pino de conexão do LED
int pinoLDR = 0; // pino de entrada do LDR
int valorBase; // valor minimo da medida do LDR
float taxa = 1.1; // taxa sobre o valor medido no LDR indicativa de início de ciclo
int ciclo; // número de ciclos medidos
int trecho = 0;
int sentido = 0;
int tempo0; // tempo início do ciclo
char buf0 = ' '; // valor anteriormente lido na porta serial

void setup(){
  pinMode(pinoRELE, OUTPUT);
  pinMode(pinoLED, OUTPUT);
  digitalWrite(pinoRELE, LOW);
  digitalWrite(pinoLED, LOW);
  Serial.begin(9600);
  Serial.println("Medindo o período");
}

void loop(){
  lerSerial();
  medirLDR();
  delay(2);
}

void lerSerial(){
  while(Serial.available() > 0){
    char buf = Serial.read();
    if (buf != buf0){
      buf0 = buf;
      if (buf == '0'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, LOW);
        Serial.println((String)"Sistema desligado !!!");
      }
      else if (buf == '1'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, HIGH);
        Serial.println((String)"Relé ligado !!!");
      }
      else if (buf == '2'){
        ciclo = 0;
        trecho = 0;
        sentido = 0;
        digitalWrite(pinoLED, HIGH);
      }
    }
  }
}

```

```

    delay(1000);
    digitalWrite(pinoRELE, LOW);
    Serial.println((String)"Período (segundos):");
    valorBase = map(analogRead(pinoLDR),0,1023,0,100);
  }
}
}
}

```

```

void medirLDR(){
  int valor = map(analogRead(pinoLDR),0,1023,0,100);
  if (valor > taxa * valorBase){
    if (trecho == 0){
      trecho = 1;
      sentido = 0;
      int tempo = millis();
      if (ciclo > 0){
        float periodo = (tempo-tempo0) / 1000.;
        Serial.println((String)periodo);
      }
      tempo0 = tempo;
      ciclo = ciclo + 1;
    }
    else if (trecho == 2){
      trecho = 1;
      sentido = 1;
    }
  }
  else{
    if (trecho == 1){
      if (sentido == 0){
        trecho = 2;
      }
      else{
        trecho = 0;
      }
    }
  }
}
}

```

APÊNDICE B – Ficha de Experimento do Pêndulo Simples

Escola Técnica Estadual Cícero Dias

Educandos:

Disciplina: Física

Turma: _____

Data: ___/___/_____

FICHA DE EXPERIMENTO

Atividade:

A expressão de dependência entre o período e o comprimento de um pêndulo simples.

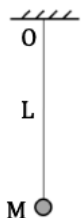
1. ETAPA PRE-EXPERIMENTAL

Organização

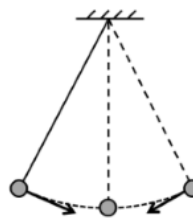
1.1. Objetivo

Determinar a expressão que relaciona o período com o comprimento de um pêndulo simples a partir da medição experimental dos períodos de pêndulos simples com diversos comprimentos.

1.2. Introdução



Pêndulo em repouso.

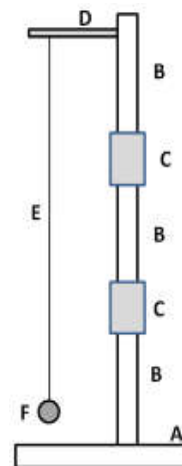


Pêndulo em movimento oscilatório em torno da posição de equilíbrio.

Um pêndulo simples é constituído por um objeto de massa M concentrada e suspenso em um ponto fixo O por um fio inextensível e de comprimento L . Quando afastado de sua posição de equilíbrio e abandonado, o corpo oscila em torno desta posição com um período de tempo T .

1.3. Material utilizado no dispositivo do pêndulo

- A. Uma base de madeira tendo dimensões de 20 cm x 30 cm x 2 cm aproximadamente e com um furo circular de raio 10 cm.
- B. Três hastes de madeira com 40 cm de comprimento e raio de 10 mm.
- C. Dois pedaços de cano PVC de 20 mm e 15 cm de comprimento.
- D. Uma haste metálica com 15 cm de comprimento com fixador para tubos de PVC de 20 mm.
- E. 1,5 m de fio de nylon fino.
- F. Uma esfera de aço com aproximadamente 1 cm de diâmetro.



Outros materiais

- a) Aparato de medição de tempo composto por placa arduino, sensor de luz, solenóide, LED e placa Bluetooth.
- b) Smartfone (ou tablet, ou computador) com apps de comunicação serial por Bluetooth e planilha eletrônica.
- c) Fita métrica de costureira com 150 cm.
- d) Lápis.
- e) Borracha.
- f) Régua.

1.4. Montagem

Preparação da parte mecânica:

- a) Para facilitar o transporte, a haste de madeira do pêndulo é constituída por três pedaços que são facilmente encaixados um ao outro usando os tubos de PVC. Deve-se primeiro fixar um deles na base de madeira e só depois fazer o encaixe dos outros dois.
- b) Amarrar num extremo da linha de nylon a esfera de aço. Uma sugestão é envolver a esfera em um pedaço de pano ou plástico e amarrá-lo com o nylon.
- c) Amarrar o outro extremo da linha de nylon na haste metálica.
- d) Encaixar a haste metálica na haste de madeira a uma altura de aproximadamente 50 cm.
- e) Colocar o aparato de medição sobre a base de madeira e diretamente abaixo da esfera metálica.

1.5. Hipótese

O período do pêndulo é proporcional a uma potência de seu comprimento:

$$T = c \cdot L^b$$

Então

$$\log T = \log(c \cdot L^b)$$

$$\log T = \log c + \log L^b$$

$$\log T = \log c + b \cdot \log L$$

Fazendo

$$y = \log T$$

$$a = \log c$$

$$x = \log L$$

Encontramos

$$y = a + b \cdot x$$

Ou seja, o logaritmo do período é uma função do primeiro grau do logaritmo do comprimento do pêndulo.

2. ETAPA EXPERIMENTAL

Coleta de dados

2.1. Procedimento de coleta de dados

- a) Ajuste a distância L entre a haste metálica e o centro da esfera para ter 50 cm.
- b) Afastar a esfera da vertical aproximadamente 15 cm, tendo o cuidado de deixar a linha de nylon esticada;
- c) Acionar o smartphone para liberar a esfera e medir o valor do período T_I da oscilação;
- d) Repetir outras três vezes para fazer as medidas de T_{II} , T_{III} e T_{IV} , registrando na Tabela 1;
- e) Repetir a medida do período para todos tamanhos de L na Tabela 1;
- f) Calcular o período médio da oscilação para cada comprimento na Tabela 1.

2.2. Tabela de registro de dados

$$T = \frac{T_I + T_{II} + T_{III} + T_{IV}}{4}$$

| Tabela 1 | | | | | |
|----------|-----------|--------------|---------------|--------------|-------|
| L (m) | T_I (s) | T_{II} (s) | T_{III} (s) | T_{IV} (s) | T (s) |
| 0,500 | | | | | |
| 0,600 | | | | | |
| 0,700 | | | | | |
| 0,800 | | | | | |
| 0,900 | | | | | |
| 1,000 | | | | | |

OBS: Trabalhar com apenas duas casas decimais

3. ETAPA PÓS-EXPERIMENTAL

Análise de dados

3.1. Conversão das medidas para seus logaritmos

Como visto em 1.5:

Hipótese: $T = c \cdot L^b$

Então $\log T = \log(c \cdot L^b)$
 $\log T = \log c + \log L^b$
 $\log T = \log c + b \cdot \log L$

Fazendo $y = \log T$
 $a = \log c$
 $x = \log L$

Encontra-se $y = a + b \cdot x$

Para achar a e b, preencha a tabela 2 seguindo as orientações a seguir:

Transcreva para a coluna 3 todos os períodos para cada comprimento do pêndulo da tabela 1;

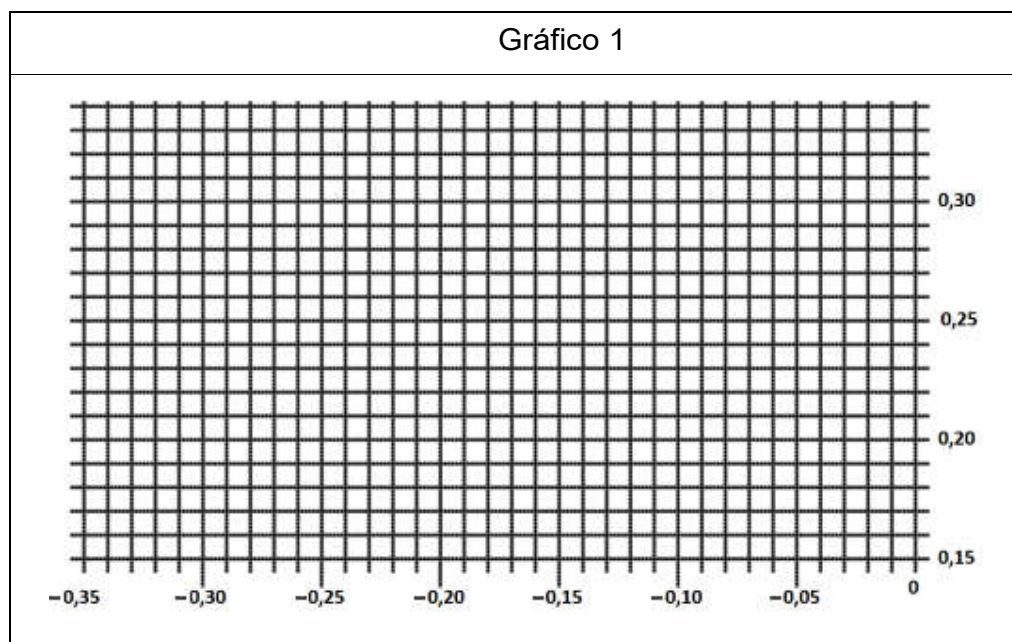
A. Calcule $x_i = \log L_i$ para cada comprimento e registre o resultado na coluna 4;

B. Calcule $y_i = \log T_i$ para cada período e registre o resultado na coluna 5;

| Tabela 2 | | | | |
|----------|-----------|-----------|-------|-------|
| i | L_i (m) | T_i (s) | x_i | y_i |
| 1 | 0,500 | | | |
| 2 | 0,600 | | | |
| 3 | 0,700 | | | |
| 4 | 0,800 | | | |
| 5 | 0,900 | | | |
| 6 | 1,000 | | | |

3.2. Fazendo o gráfico x por y

- a) Transcreva os valores de x e y para o Gráfico 1;
 b) Trace a melhor reta para os pontos no Gráfico 1;



3.3. Obtendo a expressão de dependência a partir do gráfico

- a) Como $y = a + b \cdot x$, o valor de a é igual ao valor de y para $x = 0$, assim

$$a = \underline{\hspace{2cm}}$$

- b) Selecione dois pontos (x_I, y_I) e (x_{II}, y_{II}) sobre a reta e calcule

$$b = \frac{y_{II} - y_I}{x_{II} - x_I} = \underline{\hspace{2cm}} = \underline{\hspace{2cm}}$$

- c) Calcule o valor de c usando

$$a = \log_{10} c \quad \rightarrow \quad c = 10^a = \underline{\hspace{2cm}}$$

- d) Finalmente escreva a relação

$$T = c \cdot L^b \quad \rightarrow \quad \mathbf{T = \underline{\hspace{1cm}} \cdot L}$$

3.4. Comparação Teoria x Prática

A expressão teórica é:

$$T = 2\pi \sqrt{\frac{L}{g}}$$

$$T = \frac{2\pi}{\sqrt{g}} \sqrt{L}$$

$$T \approx \frac{2 \cdot 3,14}{\sqrt{9,8}} \sqrt{L}$$

$$T \approx 2,007 \cdot L^{0,5}$$

A expressão experimental é:

$$T = \underline{\hspace{2cm}} \cdot L$$

3.5 Observações e Conclusões

APÊNDICE C – Produto Educacional



AQUISIÇÃO DE DADOS COM ARDUINO E SMARTPHONE:
EXPERIMENTO DO PÊNDBULO SIMPLES.

João Bosco Araújo Fernandes

Produto Educacional

Orientador:
Prof. Dr. Jairo Ricardo Rocha de Oliveira

Recife – PE
2020

LISTA DE FIGURAS

| | |
|---|-----|
| Figura 1 – Sistema interativo. | 169 |
| Figura 2 – Circuito elétrico simples..... | 170 |
| Figura 3 – Circuito elétrico automatizado. | 171 |
| Figura 4 – Dispositivos de entrada e saída em um projeto Arduino..... | 172 |
| Figura 5 – Arduino Uno..... | 176 |
| Figura 6 – Ícone da IDE do Arduino | 177 |
| Figura 7 – IDE do Arduino quando aberto pela primeira vez | 178 |
| Figura 8 – Menu Ferramentas do Arduino..... | 178 |
| Figura 9 – Item de menu Placas do Arduino | 179 |
| Figura 10 – Item de menu Porta do Arduino | 179 |
| Figura 11 – Detalhe de um LED verde com 5mm de tamanho | 188 |
| Figura 12 – Protoboard..... | 191 |
| Figura 13 – Esquema de ligação de um LED externo ao Arduino | 192 |
| Figura 14 – Esquema de um relé EMR simples | 195 |
| Figura 15 – Módulo relé de 1 canal | 196 |
| Figura 16 – Acionamento de uma lâmpada por um módulo relé. | 197 |
| Figura 17 – Estados de um comutador elétrico..... | 198 |
| Figura 18 – Aspecto e símbolo do botão de pressão de quatro pinos..... | 198 |
| Figura 19 – Controle do nível lógico com botão de pressão. | 199 |
| Figura 20 – Resistor de pull-up e resistor de pull-down. | 200 |
| Figura 21 – Botão pull-down para controlar acionamento do <i>LED built-in</i> | 201 |
| Figura 22 – Aspecto, elementos e símbolos de um LDR. | 204 |
| Figura 23 – Circuito do divisor de tensão. | 206 |
| Figura 24 – Divisor de tensão com LDR..... | 207 |
| Figura 25 – Controle do acionamento de um LED com um LDR. | 209 |
| Figura 26 – Monitor serial integrado a IDE do Arduino. | 211 |
| Figura 27 – Circuito para monitorar o estado de um botão momentâneo..... | 213 |
| Figura 28 – Módulo HC-06. | 215 |
| Figura 29 – Circuito para acionamento da placa HC-06. | 216 |
| Figura 31 – Lista de telas e Devices do Serial Bluetooth Terminal..... | 221 |

| | |
|---|-----|
| Figura 32 – Configuração dos botões de macros M1, M2 e M3. | 223 |
| Figura 33 – Resultado da configuração dos botões de macros M1, M2 e M3. . | 223 |
| Figura 34 – Repouso e movimento do pêndulo simples. | 224 |
| Figura 35 – Esquema de construção do pêndulo simples. | 225 |
| Figura 36 – Haste metálica conectada ao fixador de tubos de PVC..... | 226 |
| Figura 37 – Esquema de incidência da luz do LED sobre um LDR. | 227 |
| Figura 38 – Componentes do pêndulo e do dispositivo de medida. | 228 |
| Figura 39 – Invólucro opaco do LDR contra efeitos da luz ambiente..... | 229 |
| Figura 40 – Circuito elétrico de ligação do LED ao Arduino..... | 231 |
| Figura 41 – Circuito elétrico de ligação do LDR ao Arduino. | 232 |
| Figura 42 – Circuito elétrico de ligação do eletroímã ao Arduino..... | 232 |
| Figura 43 – Circuito elétrico de ligação do Bluetooth ao Arduino. | 232 |
| Figura 44 – Circuito de medição do período na protoboard..... | 233 |
| Figura 45 – Circuito de medição e fontes de tensão. | 234 |
| Figura 46 – Dispositivo de medida do período do pêndulo simples..... | 234 |
| Figura 47 – Fluxograma do sketch de controle | 236 |

SUMÁRIO

| | | |
|-------|--|-----|
| 1 | Apresentação..... | 167 |
| 2 | A plataforma Arduino | 169 |
| 2.1 | Aquisição de dados, monitoramento e controle | 169 |
| 2.2 | O microcontrolador Arduino..... | 172 |
| 2.3 | A IDE Arduino..... | 177 |
| 2.4 | Configurando a IDE off-line do Arduino..... | 178 |
| 2.5 | A linguagem de programação do Arduino..... | 180 |
| 2.5.1 | Controle de acionamento do LED built-in | 185 |
| 2.5.2 | Controle de acionamento do LED externo..... | 188 |
| 2.5.3 | Eletroímã | 193 |
| 2.5.4 | Controle de acionamento de um módulo relé | 195 |
| 2.5.5 | Controle da recepção de sinal do sensor de toque | 197 |
| 2.5.6 | Controle da recepção de sinal do sensor de luz..... | 203 |
| 2.5.7 | Transmissão via porta serial | 209 |
| 2.5.8 | Transmissão via Bluetooth..... | 214 |
| 3 | Serial Bluetooth Terminal | 219 |
| 3.1 | Usando o app serial bluetooth terminal..... | 219 |
| 3.2 | Botões de macros..... | 222 |
| 4 | Dispositivo de medida do período | 224 |
| 4.1 | Montagem do pêndulo simples | 224 |
| 4.2 | Medição do período do pêndulo com Arduino e LDR | 226 |
| 4.3 | Controle do início do movimento do pêndulo | 230 |
| 4.4 | Esquemas elétricos do dispositivo de medição..... | 231 |
| 4.5 | Determinação da frequência operação do loop | 235 |
| 4.6 | Sketch para controle do dispositivo de medição..... | 236 |
| 4.6.1 | Variáveis globais do sketch..... | 236 |
| 4.6.2 | Procedimento setup | 238 |
| 4.6.3 | Procedimento loop | 239 |

| | | |
|-------|---|-----|
| 4.6.4 | Procedimento lerSerial | 239 |
| 4.6.5 | Procedimento medirLDR | 242 |
| 5 | Proposta didática de aplicação..... | 245 |
| 6 | Considerações finais | 247 |
| | REFERÊNCIAS BIBLIOGRÁFICAS | 250 |
| | APÊNDICE I – Sketch PENDULO_SIMPLES | 255 |
| | APÊNDICE II – Ficha de Experimento do Pêndulo Simples | 257 |

1 APRESENTAÇÃO

Esse Produto Educacional foi elaborado para o Mestrado Nacional Profissional em Ensino de Física, ofertado pela Sociedade Brasileira de Física em parceria com a Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Mestre em Ensino de Física, no ano de 2020.

Seu primeiro objetivo é apresentar um guia de estudo para permitir a professores criar projetos com seus educandos do ensino médio em que um smartphone possa comunicar-se via Bluetooth com uma placa Arduino e assim controlar o acionamento de dispositivos elétricos e fazer aquisição de dados.

Seu segundo objetivo é desenvolver uma aplicação direta desse guia em que experimentalmente seja determinada a expressão harmônica do período de um pêndulo simples.

Com o propósito de reduzir a complexidade inerente a esses objetivos, foi elaborada a seguinte estratégia:

- a) Uso de uma sequência didática tentando suavizar a curva de aprendizado do Arduino e de outros dispositivos eletrônicos. Ela é fruto da experiência de vários anos orientando educandos do ensino médio no desenvolvimento de projetos usando o Arduino e foi adaptada para que professores do ensino médio e outros interessados desenvolvam projetos com smartphone e Arduino.
- b) Para evitar a nada simples tarefa de desenvolver um aplicativo de comunicação entre o smartphone e o Arduino foi usado o app gratuito Serial Bluetooth Terminal. Simples de ser obtido e instalado no smartphone, esse aplicativo Android utiliza uma interface muito fácil e amigável para enviar e receber dados por conexão Bluetooth.

- c) O experimento para determinar a expressão do período do pêndulo simples foi selecionado devido à simplicidade de sua construção e estudo experimental. É importante observar que pequenas modificações apenas no software e na disposição dos componentes desse experimento permitem fazer a medição de tempo em uma série de outros experimentos. A inclusão de uns poucos novos dispositivos simples de usar permite elevar em muito essa quantidade de experimentos.

Assim, no Capítulo 2 é apresentado o Arduino e visto em detalhes como usá-lo para acionar dispositivos elétricos tais como LEDs e reles. A seguir é analisado como esse microcontrolador manipula sensores, em particular, o sensor de luz. E no final do capítulo é descrito como é feita a comunicação via Bluetooth entre o Arduino e um smartphone Android.

No Capítulo 3 é apresentado o Serial Bluetooth Terminal, um programa que instalado em um smartphone Android permite de forma muito simples enviar e receber dados para qualquer dispositivo que use uma conexão Bluetooth.

Em seguida, no Capítulo 4 é discutido em detalhes um dispositivo de medição do período do pêndulo simples por aquisição de dados com Arduino e smartphone.

Posteriormente, no Capítulo 5 é apresentada uma proposta didática para aplicação do experimento de obtenção da expressão período do pêndulo simples em uma turma de educandos do ensino médio.

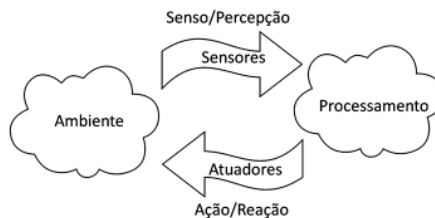
Por último, no Capítulo 6, são apresentadas as considerações finais.

2 A PLATAFORMA ARDUINO

2.1 AQUISIÇÃO DE DADOS, MONITORAMENTO E CONTROLE

A aquisição de dados pode ser definida, de maneira simples, como a medição de informações do mundo real. Um sistema de aquisição de dados realiza a coleta de parâmetros de um fenômeno físico real, faz sua conversão num sinal elétrico e o formata de modo conveniente para seu armazenamento, processamento, visualização e análise. Os sistemas de monitoramento e controle verificam continuamente os dados recebidos e enviam sinais em resposta aos valores recebidos e executam ações que dependem desses valores (PAMBOUKIAN; PEDROTTI, 2012), ao que Banzi (2011, p. 27) chama de processo interativo e está esquematizado na Figura 1.

Figura 1 – Sistema interativo.



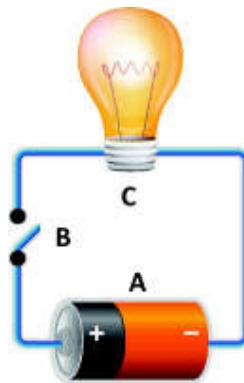
Adaptado de Banzi (2011, p. 27).

As quantidades físicas a serem medidas podem ser as mais variadas (luminosidade, temperatura, pressão, força, deslocamento, etc.), mas todas relacionadas com algum tipo de energia. Os dispositivos que convertem uma forma de energia em outra são denominados transdutores. Assim, um transdutor elétrico converte alguma grandeza não elétrica em sinais elétricos e vice-versa e são classificados em sensores e atuadores. Os sensores convertem alguma grandeza medida no mundo real em sinais elétricos. Já os atuadores são transdutores que podem converter um sinal elétrico em uma ação física de reação (BANZI, 2011, p. 27).

Para realizar o processamento e a análise dos dados é necessário um sistema eletrônico, que não necessariamente precisa usar a tecnologia da informática. Mas, com o barateamento e aumento de capacidade e velocidade dos microcontroladores, é cada vez mais comum o seu uso juntamente com sensores e atuadores no desenvolvimento de dispositivos interativos eletrônicos capazes de realizar aquisição de dados, monitoramento e controle de sistemas (BANZI, 2011, p. 27).

De um ponto de vista da física elétrica, qualquer dispositivo interativo eletrônico se baseia no funcionamento de um circuito elétrico simples, como o mostrado na Figura 2. Quando a chave interruptora manual (item B da Figura 2) está aberta não há corrente elétrica circulando. Porém, quando ela é fechada, passa a fluir uma corrente elétrica que leva ao funcionamento do receptor elétrico (item C da Figura 2).

Figura 2 – Circuito elétrico simples



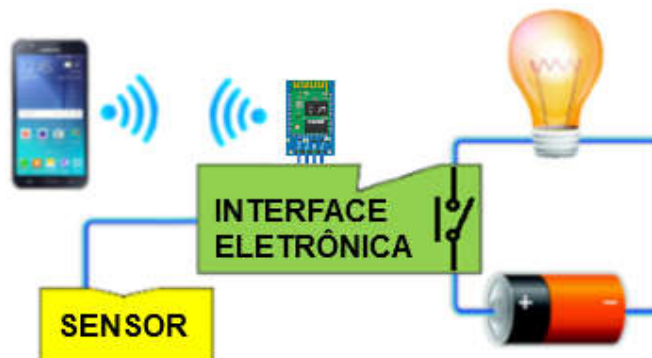
A = Gerador elétrico B = Chave interruptora C =Receptor elétrico

Adaptado de: <https://vestibular.brasilecola.uol.com.br/enem/abordagem-dos-circuitos-eletricos-no-enem.htm>. Acesso em: 25 jun 2020.

A substituição da chave interruptora manual da Figura 2 por uma interface eletrônica controlada por um computador ou microcontrolador, como mostra a Figura 3, leva a automatização do funcionamento circuito, com o seu controle

podendo ser realizado com uso de sensores e também com equipamentos de comunicação sem fio, como por exemplo, via Bluetooth.

Figura 3 – Circuito elétrico automatizado.



Adaptado de Brasil Escola¹, Samsung² e Embarcados³.

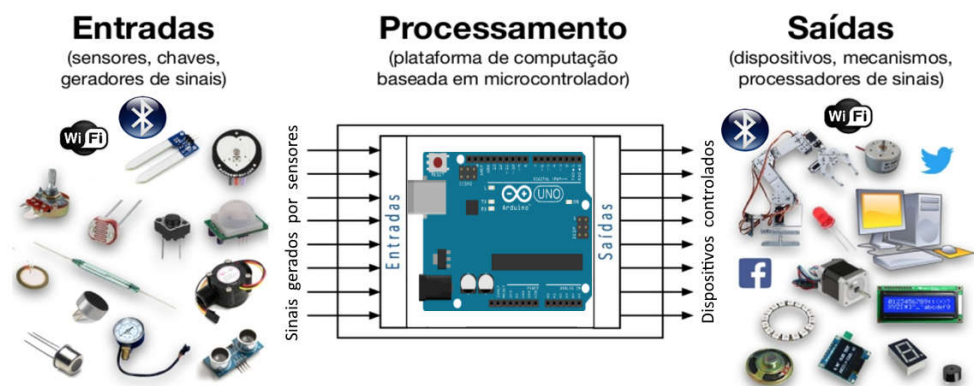
É importante observar que, como mostra a Figura 4, existe uma diversidade de sensores, chaves e geradores de sinais que podem ser usados como entrada de dados e imensa quantidade de dispositivos, mecanismos e processadores de sinais usados como saída de dados, além de ser possível usar comunicação por USB, ethernet, Bluetooth, wifi, GSM, etc. Além disso, todos esses dispositivos e tecnologias têm custo cada vez mais baixo e são cada vez mais fáceis de serem adquiridos no mercado especializado. Tudo isso permite o desenvolvimento de uma infinidade de projetos.

¹ Disponível em: <https://vestibular.brasilecola.uol.com.br/enem/abordagem-dos-circuitos-eletricos-no-enem.htm>. Acesso em: 25 jun. 2019.

² Disponível em: <https://www.samsung.com/br/smartphones/galaxy-j5-j500m/SM-J500MZKZZTO/>. Acesso em: 25 jun. 2019.

³ Disponível em: <https://www.embarcados.com.br/modulos-bluetooth-hc-05-e-hc-06/>. Acesso em: 25 jun. 2019.

Figura 4 – Dispositivos de entrada e saída em um projeto Arduino.



Adaptado de: <https://pt.slideshare.net/helderdarocha/introduo-ao-arduino-81351703>.
Acesso em: 30 jul. 2019.

2.2 O MICROCONTROLADOR ARDUINO

O Arduino foi desenvolvido em 2005 no *Interaction Design Institute*, na cidade italiana de Ivrea e o nome Arduino foi dado à nova placa em referência a um bar frequentado por professores e estudantes desse instituto. Foi idealizado por Massimo Banzi, um professor preocupado em encontrar um meio barato e fácil de seus estudantes de arte e design trabalharem com tecnologia, em contrapartida aos produtos existentes até então, caros e relativamente difíceis de usar. As duas exigências básicas eram, primeiro que fosse barato e com preço não podendo ser maior que o custo da pizza que um estudante fosse comprar e em segundo lugar que fosse um sistema que qualquer um pudesse utilizar. Assim Banzi solicitou a David Cuartielles, pesquisador visitante da Universidade sueca de Malmö, desenhar os circuitos da placa, ao mesmo tempo em que David Mellis, aluno de Banzi, desenvolvia o software de controle da placa. Em seguida, um engenheiro local chamado Gianluca Martino foi contratado para orientar os alunos do *Design Institute* no desenvolvimento dos projetos (EVANS; NOBLE; HOCHENBAUM, 2013, p. 25-26)

Como era trabalhoso para os estudantes encontrar todos os componentes e dispositivos necessários, foram produzidos kits contendo as placas e os itens eletrônicos mais comuns dos seus projetos. Foi um grande sucesso e o estoque

inicial de duzentos kits foi rapidamente vendido, sendo necessária a fabricação de mais unidades para suprir a demanda. Designers, artistas e profissionais de outras áreas se interessaram pela novidade e passaram a usá-lo em seus projetos e a sua popularidade cresceu ainda mais. O sucesso foi confirmado com a obtenção de uma menção honrosa na categoria Comunidades Digitais em 2006, pela *Prix Ars Electronica*, além da marca de mais de 50.000 placas vendidas até outubro de 2008. O grande público percebeu que além de sua relativa facilidade de uso e custo baixo ele era uma excelente introdução para programação de microcontroladores e dessa forma poderia ser usado por qualquer pessoa interessada (WIRED, 2019).

O Arduino é uma plataforma eletrônica baseada em hardware livre e software de código aberto. Assim, pode-se comprar uma placa Arduino pronta ou adquirir seus componentes separadamente e conectá-los de acordo com as necessidades do projeto a ser desenvolvido. É possível encontrar, principalmente em sites da internet, todas as rotinas necessárias ao projeto já prontas para uso ou somente parte delas e desenvolver só o restante que falta ou ainda desenvolver totalmente os algoritmos necessários ao projeto. Tudo dependendo da necessidade, tempo e interesse do usuário (ARDUINO INTRODUÇÃO, 2019).

Ele pode ser usado para desenvolver objetos interativos independentes ou ser conectado a um computador, a uma rede ou até mesmo à internet para recuperar e enviar dados do Arduino e trabalhar com eles. Assim, pode ser usado em soluções IoT (*Internet of Things*, internet das coisas), *wearable* (vestível), impressão 3D e ambientes incorporados. Como exemplo simples, pode enviar um conjunto de dados recebidos de sensores para um site e exibi-los na forma de tabela ou de um gráfico (ARDUINO INTRODUÇÃO, 2019).

Com o passar dos anos, o Arduino tem permitido o desenvolvimento de milhares de projetos, desde objetos do cotidiano até instrumentos científicos complexos. Uma crescente comunidade mundial de entusiastas (estudantes,

amadores, artistas, programadores e profissionais de diversas áreas) utilizando de forma intensa essa plataforma contribuiu para o desenvolvimento de uma incrível quantidade de literatura livre, de fácil acesso e de grande ajuda para iniciantes e especialistas (ARDUINO INTRODUÇÃO, 2019).

Uma pesquisa na Internet por “Arduino” mostrará um vasto número de sites dedicados ao Arduino e que apresentam projetos interessantes em que ele foi utilizado para ler dados e controlar uma enorme quantidade de dispositivos. Por isso ele é um produto incrível e com um pouco de vontade para aprender como usá-lo será possível a criação de quase tudo, desde obras de arte interativas até robôs (MCROBERTS 2011).

O Arduino original tem sua programação feita diretamente pelo IDE (*Integrated Development Environment*, ambiente de desenvolvimento integrado) oficial do Arduino, o layout da placa é padronizada, está documentado no site oficial (ARDUINO HOME, 2019), está devidamente licenciado para exibir o nome e o logotipo do Arduino e é produzido por fabricantes autorizados e oficiais que atualmente são SmartProjects na Itália, Sparkfun nos EUA e DogHunter em Taiwan / China (ARDUINO CLONES, 2019).

Como ele tem hardware aberto, há muitas placas não oficiais disponíveis para compra com base no Arduino original, ou que podem ser criadas a partir de um diagrama de fácil compreensão. Assim é possível comprar os componentes apropriados e montar uma placa Arduino completamente funcional em uma matriz de pontos ou em uma PCB (*Printed Circuit Board*, placa de circuito impresso) feita em casa. A equipe do Arduino impõe apenas que não seja utilizado o nome “Arduino” nem o seu logotipo, pois eles são de uso exclusivo da placa oficial (MCROBERTS, 2011).

O Arduino original ou oficial usa um microcontrolador da família ATmega (Oliveira 2018). Apenas no site do Arduino estão disponíveis para compra dezenas de diferentes tipos de placas, cada uma com uma finalidade específica,

além de módulos para expansão (*shields*) e kits que estão agrupados em *Entry Level* (Nível de Entrada), *Enhanced Features* (Recursos Aprimorados), *Internet of Things* (*IoT - Internet das Coisas*), *Education* (Educação), *Wearable* (Vestível), além do grupo *Retired* (Aposentado) que mostra a história de diversos modelos de Arduino e placas auxiliares que foram descontinuados por ficarem obsoletos (ARDUINO PRODUTOS, 2019).

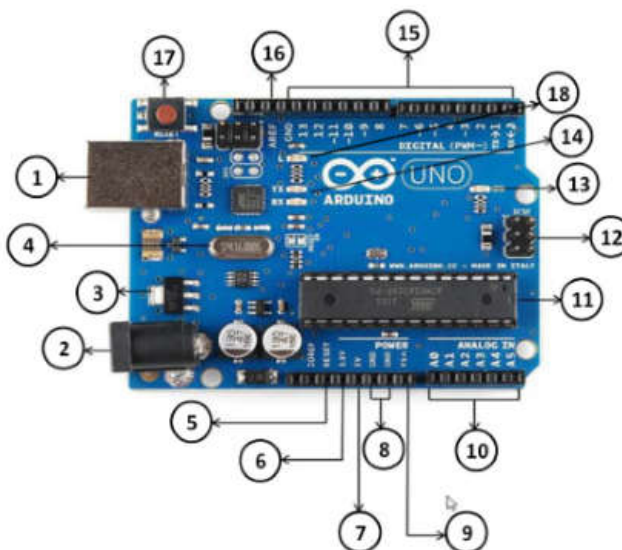
A placa Arduino Uno (Figura 5) é a primeira de uma série de placas USB Arduino com microcontrolador ATmega328P (componente 11 da Figura 5). Pertencente ao grupo *Entry Level*, é atualmente o modelo de referência para a plataforma Arduino, sendo a placa mais utilizada nos projetos em geral e a mais fácil de fazer a iniciação em Arduino (ARDUINO UNO, 2019).

Através do conector USB (componente 1 da Figura 5) o Arduino pode ser alimentado e controlado diretamente pelo computador. Para tornar-se autônomo, um programa deve ser desenvolvido no computador e transferido para a memória do Arduino também através da conexão USB. Depois de desconectado, é necessário que através do conector de alimentação (componente 2 da Figura 5) a placa receba energia de alguma fonte com tensão contínua entre 6 V e 20 V. (MONK, 2014, p. 1).

É importante observar que os pinos digitais de 0 a 13 podem ser configurados como pinos de entrada para ler valores lógicos (0 ou 1) ou como pinos de saída para acionar diferentes dispositivos como LEDs, relés, etc. Os pinos 3, 5, 6, 9, 10 e 11 são rotulados com “~” pois podem ser usados também para fornecer saída do tipo PWM⁴ (TUTORIALSPPOINT, 2019).

⁴ Acrônimo de *Pulse Width Modulation* e significando Modulação de Largura de Pulso.

Figura 5 – Arduino Uno



| Componente | Descrição |
|------------|--|
| 1 | Conector USB – Permite comunicação com um PC via cabo USB |
| 2 | Conector de alimentação – Entrada de tensão (6 V a 20 V DC). |
| 3 | Regulador de voltagem – Estabiliza as tensões DC em toda a placa. |
| 4 | Oscilador de cristal de 16 MHz – Temporizador do sistema. |
| 5 | Pino de reset da placa – Reinicializa a placa com botão externo. |
| 6 | Pino de 3,3 V – Fornece 3,3 V de saída. |
| 7 | Pino de 5 V – Fornece 5 V de saída. |
| 8 | Pino GND (Ground) – Pino terra e de retorno de corrente. |
| 9 | Pino Vin – Entrada de tensão por fonte de energia externa AC. |
| 10 | Pinos analógicos de A0 a A5. |
| 11 | Microcontrolador da placa – Atualmente no Uno é o ATmega328P. |
| 12 | Pino ICSP – Usado para transferir programas/firmwares e também para executar tarefas administrativas. |
| 13 | LED indicador – Indica que a placa está ligada corretamente. |
| 14 | LEDs TX e RX – O LED TX pisca enquanto envia os dados seriais. A O LED RX pisca durante o processo de recebimento. |
| 15 | Pinos digitais – A placa Arduino UNO possui 14 pinos de E/S digitais |
| 16 | Pino AREF – Define uma tensão de referência externa (entre 0 e 5 Volts) como o limite superior para os pinos de entrada analógica. |
| 17 | Botão de reset da placa – Reinicializa a placa. |
| 18 | LED built-in – conectado geralmente ao pino digital 13. |

Adaptado de: TUTORIALSPPOINT (2019).

2.3 A IDE ARDUINO

Para controlar a placa do Arduino é necessário desenvolver um *sketch* (código) que é um algoritmo de controle e fazer sua gravação na placa através de uma conexão USB. Esse desenvolvimento é feito no Arduino Software (IDE) e atualmente está disponível em duas opções (ARDUINO START, 2019):

a) IDE *on-line* (*Arduino Web Editor*) – Está disponível para execução na seção Software do site oficial do Arduino (ARDUINO SOFTWARE, 2019) Deve-se usar essa opção somente quando houver disponível uma conexão de Internet confiável. As placas originais e algumas placas clones licenciadas funcionam prontamente sem necessidade de nenhuma instalação adicional. Essa interface permite que os *sketchs* criados sejam salvos na internet, disponibilizando-os a partir de qualquer dispositivo online e fazendo backup automático. Sempre será executada a versão mais atualizada do IDE sem a necessidade de instalar atualizações ou bibliotecas geradas pela comunidade.

b) IDE *off-line* – Também está disponível na seção Software do site oficial do Arduino (ARDUINO SOFTWARE, 2019). Essa opção é indicada quando não houver disponível uma conexão confiável com a internet ou forem utilizadas placas clones não compatíveis com a opção *on-line*. É necessário baixar a IDE de acordo com o sistema operacional usado (Windows, Linux ou OS) e o próprio site sugere o uso da versão mais recente. Após o download deve-se fazer a instalação no computador que pode ser feita através do hiperlink **Getting Started**, disponível na própria página de download da IDE. Após a instalação aparece o ícone da IDE do Arduino indicado pela Figura 6 (ARDUINO START, 2019).

Figura 6 – Ícone da IDE do Arduino

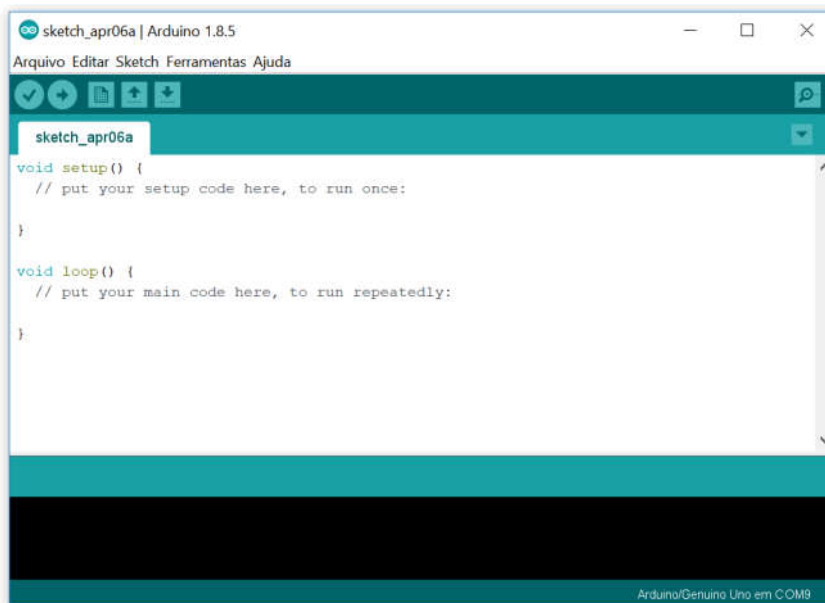


Fonte: print screen da IDE no sistema operacional Windows 10.

2.4 CONFIGURANDO A IDE OFF-LINE DO ARDUINO

Assim que a IDE for aberta aparece algo similar a Figura 7.

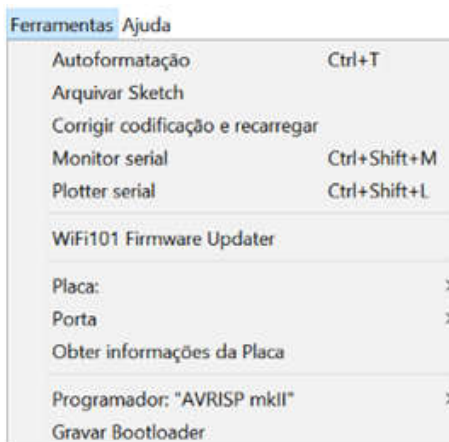
Figura 7 – IDE do Arduino quando aberto pela primeira vez



Fonte: print screen da IDE no sistema operacional Windows 10.

Acionando o item de menu **Ferramentas** clicar em **Placa** (Figura 8).

Figura 8 – Menu Ferramentas do Arduino



Fonte: print screen da IDE no sistema operacional Windows 10.

A seguir será apresentada a uma lista de placas (Figura 9). Deve-se selecionar a opção apropriada a partir dessa lista.

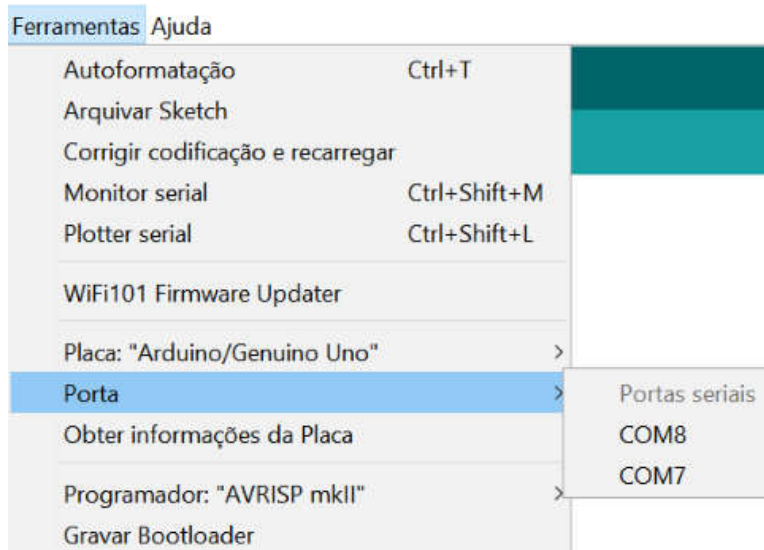
Figura 9 – Item de menu Placas do Arduino



Fonte: print screen da IDE no sistema operacional Windows 10.

Como último passo da configuração, deve-se acionar novamente o menu **Ferramentas**, clicar em **Porta** e escolher na lista a porta apropriada para a placa do Arduino (Figura 10).

Figura 10 – Item de menu Porta do Arduino



Fonte: print screen da IDE no sistema operacional Windows 10.

O IDE do Arduino é bem simples, e permite rápido aprendizado à medida que é utilizado. Conforme o usuário se torna mais hábil no uso da IDE padrão, pode ser considerado experimentar alguma IDE profissional, como Eclipse, ArduIDE, GNU/Emacs, AVR-GCC, AVR Studio e até mesmo o XCode da Apple (MCROBERTS, 2011).

2.5 A LINGUAGEM DE PROGRAMAÇÃO DO ARDUINO

A linguagem de programação do Arduino é baseada em Wiring (ARDUINO INTRODUÇÃO, 2019). Wiring é uma linguagem de programação de código aberto para microcontroladores e se baseia no Processing, um projeto aberto desenvolvido a partir das idéias surgidas no *MIT Media Lab* (WIRING, 2019).

Um programa em Arduino é chamado *sketch* e é composto por uma sequência de instruções que apenas podem ser escritas com palavras-chave e que seguem um conjunto bem definido de regras da linguagem (MCROBERTS, 2011).

O termo função na programação do Arduino é extremamente importante e tem vários significados. Primeiro, ele pode ser usado com o mesmo sentido de uma função matemática, pois recebe parâmetros como entrada (o domínio da função) e retorna um valor de saída (a imagem da função). Como exemplo simples, a execução da instrução com sintaxe **sqrt(25)** calcula a raiz quadrada de 25. Ela tem como parâmetro de entrada o valor 25 e como parâmetro de retorno ou de saída o valor 5. Aqui o termo função está associado a apenas uma única instrução pertencente ao grupo de palavras-chave do Arduino. Podem existir funções sem parâmetros de entrada e nesse caso não há nada entre seus parênteses e podem existir funções sem parâmetros de saída em que é obrigatório colocar a palavra-chave **void** antes de serem declaradas. Assim, a instrução **void setup()** indica que `setup` é uma função sem parâmetros de entrada e também sem parâmetros de saída (MCROBERTS, 2011).

Em um sentido diferente, uma função pode ser também um conjunto de comandos agrupados em um bloco de código que se inicia com o símbolo { e termina com o símbolo }. Tudo entre esses dois símbolos fará parte da função. Neste sentido, uma função pode ter muitas linhas de código, cada linha contendo uma ou várias instruções e pode-se executar esse código inúmeras vezes simplesmente chamando o nome da função, em vez de ter de reescrever todo o código cada vez que for usá-lo. É obrigatório para um *sketch* Arduino ter uma função **setup()** e uma função **loop()**, do contrário, ele não funcionará. Por não terem nenhum parâmetro de entrada não há nada entre seus parênteses. Ambas devem ser precedidas da palavra-chave **void** para indicar que são funções que não retornam nenhum parâmetro. A Figura 7 mostra uma possível imagem da IDE logo após sua abertura e nela as duas funções podem ser vistas (MCROBERTS, 2011).

Assim que o Arduino é energizado ou imediatamente após o término da transferência do *sketch* entre o computador e a memória do Arduino, a função **setup()** é a primeira parte do *sketch* a ser executada e sempre uma única vez. Nela estão contidas as instruções de inicialização do programa, como a definição dos modos dos pinos, das taxas de transmissão serial etc. Ela inicia com **void setup()** para informar ao compilador que é uma função chamada **setup**, que não retorna nenhum dado (**void**) e que não recebe nenhum parâmetro (parênteses vazios) (MCROBERTS, 2011).

A função **loop()** é a principal função do *sketch* e inicia sua execução logo após o **setup**. As declarações dentro dessa função são executadas uma por vez até a sua última linha, quando é reiniciando o loop num processo repetitivo até que o Arduino seja desligado ou o botão Reset pressionado. De modo similar ao **setup**, ela inicia com **void loop()** para informar ao compilador que não retorna nenhum dado e que não recebe nenhum parâmetro (MCROBERTS, 2011).

Por exemplo, se num programa for desenvolvida uma função (ou um bloco de código) de nome **myFunc**, em que a sintaxe da escrita seja algo como:

```
int myFunc (int x, int y) {  
    // instruções da função  
}
```

Então a função **myFunc** recebe como entrada dois números inteiros, x e y. E quando seu processamento for concluído, será retornado um valor inteiro no ponto de execução imediatamente após onde ela tiver sido chamada no programa. Para indicar que ela retorna um valor inteiro é obrigatório preceder seu nome com a palavra-chave `int`. Pelo mesmo motivo deve-se fazer isso com cada parâmetro de entrada inteiro (MCROBERTS, 2011). É sempre necessário preceder o nome da função e cada um de seus parâmetros de entrada com a correspondente palavra-chave indicativa tipo de dado (ARDUINO LINGUAGEM, 2019).

À medida que um *sketch* se torna mais complexo, e passar a ter dezenas, centenas ou talvez milhares de linhas, comentários serão vitais para facilitar a compreensão de como cada seção funciona. Pode-se desenvolver um trecho incrível de código, mas provavelmente não será possível lembrar-se de como ele funciona quando for tentar entendê-lo dias, semanas, meses ou até anos após a sua criação. Por isso, inserir comentários é essencial para a compreensão do código. Além disso, faz parte do espírito do Arduino, e de toda a comunidade de fonte aberta, o compartilhamento de código e projetos. E o uso dos comentários pode ajudar muito a alguém interessado na compreensão de um determinado algoritmo (MCROBERTS, 2011).

Um comentário que ocupa toda uma linha ou parte dela deve iniciar com `//`, e qualquer texto a direita será desconsiderado no momento da execução dessa linha do programa. Caso o comentário possua várias linhas é melhor usar outro formato: um bloco com linhas de texto entre os sinais `/*` e `*/`. Como exemplo, as duas formas são usadas para fazer comentários no *sketch* Blink (ARDUINO BLINK, 2019).

As instruções da linguagem de programação do Arduino estão agrupadas em três partes principais: **functions** (funções), **values** (valores variáveis e constantes) e **structure** (estrutura). Na IDE do Arduino as instruções do grupo **functions** são escritas em vermelho, as do grupo **values** então em azul e aquelas do grupo **structure** em marrom. Já os comentários ficam em cinza e os demais termos numéricos e alfanuméricos não pertencentes ao grupo de palavras-reservadas ficam na cor preta (ARDUINO LINGUAGEM, 2019).

O grupo **functions** reúne as instruções que servem para controlar a placa do Arduino e realizar cálculos matemáticos. Como exemplo, no trecho de código:

```
void setup() {  
  pinMode(13, OUTPUT); // configura o pino digital 13 como saída  
}
```

A função **setup** é usada para configurar o pino digital 13 para o modo de saída de dados através do uso da função **pinMode**, que pertence ao grupo **functions** da linguagem do Arduino. É importante observar que por exigência da linguagem, é obrigatório finalizar cada instrução com um ponto e vírgula. Se desejado, após uma instrução pode-se colocar outra instrução ou um comentário (ARDUINO LINGUAGEM, 2019).

O grupo **values** contém as constantes e variáveis que serão usadas ao longo do programa. Constantes são expressões predefinidas na linguagem Arduino. Elas são usadas para tornar os programas mais fáceis de ler. É importante observar que a linguagem diferencia caracteres maiúsculos e minúsculos. Como no exemplo, na instrução **pinMode** é necessário especificar como primeiro parâmetro o número do pino e como segundo parâmetro o comportamento elétrico do pino, ou seja, se é um pino de entrada (INPUT) ou de saída (OUTPUT). Pinos configurados como INPUT adquirem um estado de alta impedância e consomem correntes extremamente pequenas no circuito que eles estão conectados, equivalente a um resistor em série de 100 MΩ conectado ao

pino. Isso os torna úteis para ler um sensor. Pinos configurados como OUTPUT passam a ter um estado de baixa impedância, podendo fornecer até 40 mA de corrente para outros circuitos. Isso os torna úteis para alimentar LEDs e dispositivos que usam menos de 40 mA de intensidade de corrente. Cargas maiores que 40 mA, como motores, exigirão um transistor ou outro circuito de interface (ARDUINO LINGUAGEM, 2019).

No grupo **values**, além das constantes, pode-se trabalhar com variáveis. Uma variável é um local na memória do computador em que é possível armazenar dados de diversos tipos, tais como texto (**string**), inteiro (**int**), decimal (**float**) etc. Assim, uma variável de tipo **int** é um número inteiro entre -32 768 e 32 767. Logo, segundo Mcroberts 2011, a instrução:

```
int ledPin = 10;
```

Atribui à variável de tipo inteiro o nome **ledPin** e dá a ela um valor de 10.

Pelas regras da linguagem Arduino, todas as variáveis devem iniciar com uma letra maiúscula ou minúscula e o restante do nome pode ser formado por letras, números e underscores. É proibido utilizar palavras-chave da linguagem como nomes de variáveis. Para ajudar a evitar isso, todas as palavras-chave no *sketch* serão mostradas em vermelho, azul ou verde (MCROBERTS, 2011).

Toda variável pode ter um escopo local ou global em relação ao programa. Se ela for global deverá ser declarada no início do algoritmo e antes da função setup e existirá enquanto o Arduino estiver em funcionamento, podendo ser acessada e alterada por qualquer função do algoritmo. Já uma variável local deverá ser declarada dentro do bloco de instruções de uma função. Poderá ser vista e modificada apenas pelo código dentro do seu próprio bloco e será destruída ao final da execução dessa função. Assim, no trecho de código:


```
int ledPin = 10;

void setup() {
  int x = 0;
  // instruções da função
}
```

São criadas a variável global inteira **ledPin** e a variável local inteira **x**. A variável **x** é uma variável local e só poderá ser usada dentro da função `setup`. Já a variável **ledPin** é uma variável global e poderá ser utilizada em qualquer outra função do algoritmo (MCROBERTS, 2011).

Structure é o terceiro grupo da linguagem Arduino e contém os elementos do *sketch* (**setup** e **loop**), de controle de estrutura (**if**, **for**, **while**, etc.), dos operadores (aritméticos, de comparação, booleanos, etc) e elementos de sintaxe adicional (**#define**, **#include**, etc.) (ARDUINO LINGUAGEM, 2019).

2.5.1 Controle de acionamento do LED built-in

Um LED é um *Light Emitting Diode* (Diodo Emissor de Luz). Um diodo é um dispositivo que permite a corrente elétrica passar em um único sentido por ele. Caso a corrente tente passar no sentido oposto, o diodo impede que ela o faça. Diodos podem ser úteis para prevenir que se faça uma conexão errada que reverta o sentido da corrente e danifique algum componente. Os LEDs impedem essa reversão e também emitem luz. Eles são fabricados com diferentes tamanhos, níveis de luminosidade e cores, incluindo o espectro ultravioleta e infravermelho como os LEDs dos controles remotos do televisor, condicionador de ar e outros aparelhos eletrônicos (MCROBERTS, 2011).

Na placa Arduino existem quatro LEDs. O primeiro é o *Power LED*, na cor verde, que permanece aceso enquanto o Arduino estiver em funcionamento. Outros dois LEDs, para comunicação serial, na cor laranja, só ficam acesos quando a placa envia ou recebe dados do computador através do cabo USB. O

TX LED informa quando o Arduino envia dados para o PC e o *RX LED* informa quando o Arduino recebe dados do PC. Finalmente, o *Status LED* ou *LED built-in* (embutido na placa) ou ainda *LED on-board*, também de cor laranja, está conectado eletricamente a um dos pinos digitais (ELECTRICARENA, 2019).

Para referir-se ao *LED built-in* em um comando é usada a constante predefinida `LED_BUILTIN`. Ela tem valor inteiro e igual ao do pino digital a que o LED está conectado e na grande maioria das placas é o pino 13. Se houver certeza do número do pino, esse valor pode ser usado no lugar de `LED_BUILTIN` (ARDUINO CONSTANTES, 2019).

A IDE Arduino apresenta, no menu Arquivo, item Exemplo, uma grande quantidade de exemplos de *sketchs* que podem ser abertos, analisados estudados e modificados. No subitem Exemplos embutidos 01.Basics, encontra-se o Blink (pisca), o exemplo ideal para início de estudo do Arduino. Sua codificação precisa de poucos comandos para fazer o *LED built-in* de forma repetitiva acender por um segundo e em seguida apagar por um segundo. Sem as linhas de comentário o seu algoritmo é:

```
void setup() {  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

A primeira e única linha do `setup` simplesmente informa através do comando **`pinMode`** que o pino `LED_BUILTIN` será de saída e por isso recebe o parâmetro `OUTPUT`. Essa indicação deverá ser sempre usada quando se desejar acionar um dispositivo conectado a qualquer porta digital. Caso o pino fosse usado para fazer leitura de um sensor deveria ser usado o parâmetro `INPUT` (ARDUINO CONSTANTES, 2019).

Dentro da função loop, **digitalWrite** é a primeira instrução e permite definir um valor HIGH ou LOW para o pino. Quando se define um pino como HIGH, ele passa a receber entre 4,5 a 5,0 volts do Arduino. Quando se define como LOW, o pino recebe entre 0 a 0,8 a volts, tornando-se um terra. Assim, a primeira instrução envia 5 V para o pino LED_BUILTIN e acende o *LED built-in* (ARDUINO CONSTANTES, 2019).

Quando um pino é configurado com saída, ele é posto em estado de baixa impedância e pode fornecer (drenar) até 40 mA de corrente para um circuito externo. Essa configuração é perigosa para a placa Arduino. Por isso, quando um pino for usado como saída de dados, deve sempre haver a preocupação em não ultrapassar o limite de corrente e danificar o pino ou mesmo queimar o microcontrolador (ARDUINO PINOS DIGITAIS, 2019).

É muito importante também observar que pelo *datasheet* do microcontrolador do Arduino Uno, o ATmega 328, a soma das correntes drenadas por todos os pinos não pode ultrapassar 200 mA com temperatura de 25 °C e apenas 100 mA quando a 125 °C (ATMEGA, 2019, p. 15).

A segunda instrução do loop usa o comando **delay** e simplesmente informa ao Arduino para esperar 1000 milissegundos (um segundo) antes de executar a próxima instrução. A terceira instrução do loop anula a tensão que vai para o pino LED_BUILTIN apagando o LED. Por fim a quarta instrução, um **delay**, manda que se espere por mais 1000 milissegundos para ir para o final do loop e reiniciar o algoritmo num processo infinito (MCROBERTS, 2011).

Uma possível melhoria desse *sketch* é:

```
int pinoLED = LED_BUILTIN;

void setup() {
  pinMode(pinoLED, OUTPUT);
}
```

```

void loop() {
  digitalWrite(pinoLED, HIGH);
  delay(1000);
  digitalWrite(pinoLED, LOW);
  delay(1000);
}

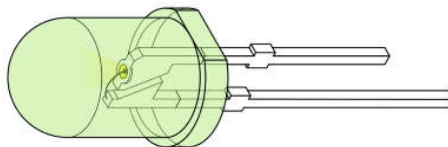
```

No trecho de código, através da instrução **int**, é criada uma variável global inteira para armazenar o número do pino digital usado como parâmetro nos comandos **pinMode** e **digitalWrite**. A grande vantagem dessa modificação é que se por qualquer motivo for necessário usar outro pino, bastar alterar uma única linha, a primeira, simplesmente inserindo o número do pino desejado.

2.5.2 Controle de acionamento do LED externo

Em um examine cuidadoso de um LED percebem-se dois detalhes: os terminais têm comprimentos diferentes, e um lado do LED é chanfrado, em vez de cilíndrico (Figura 11). Esses detalhes servem para indicar qual terminal é o ânodo (positivo) e qual é o cátodo (negativo): o terminal mais comprido (ânodo) é conectado à tensão positiva e o terminal mais curto e com o lado chanfrado (cátodo) vai para o terra ou negativo. Se o LED for conectado ao contrário, isso não o danificará se ele não tiver que conduzir correntes muito elevadas.

Figura 11 – Detalhe de um LED verde com 5mm de tamanho



Adaptado de: Wikipédia⁵

De acordo com o *datasheet* da Atmega, um pino digital emite 5 V com corrente máxima de 40 mA (Mcroberts 2011). Já o *datasheet* de um LED de alto

⁵ Disponível em: [https://commons.wikimedia.org/wiki/File:LED,_5mm,_green_\(en\).svg#/media/File:LED,_5mm,_green_\(unlabelled,_full\).svg](https://commons.wikimedia.org/wiki/File:LED,_5mm,_green_(en).svg#/media/File:LED,_5mm,_green_(unlabelled,_full).svg). Acesso em: 05 abr. 2019.

brilho branco de 5 mm (LED, 2019) indica que ele suporta uma corrente máxima de 20 mA, situação em que recebe entre seus terminais uma tensão da ordem de 3 V. Assim, ligar diretamente um LED a uma porta digital irá sobrecarregá-lo, sendo por isso necessário utilizar alguma estratégia para limitar a corrente que atravessa o LED. A mais simples é a utilização de um resistor em série a ele.

Um resistor é um dispositivo projetado para provocar resistência a uma corrente elétrica e quando conectado em série a um dispositivo, tem-se como resultado uma queda da corrente que atravessa a ambos e também a diminuição da tensão nos terminais do dispositivo. Assim, para limitar a corrente que atravessa o LED e evitar a sua queima, é necessário reduzir os 5 V do pino digital do Arduino para os 3 V requeridos pelo LED. Logo, o terminal curto do LED (cátodo) deve ser conectado ao terra (GND), o terminal longo (ânodo) deve ser ligado em série com um resistor e este deve ser conectado a um dos pinos digitais. Para determinar o valor R da resistência adequada é necessário usar a primeira lei de Ohm:

$$R = \frac{V}{I} \quad (1)$$

Em que V é a tensão nos terminais do resistor e I a corrente que o percorre. Mas a tensão no resistor deve ser a diferença entre a tensão fornecida pelo pino digital (V_S) e a tensão requerida pelo LED (V_L), então:

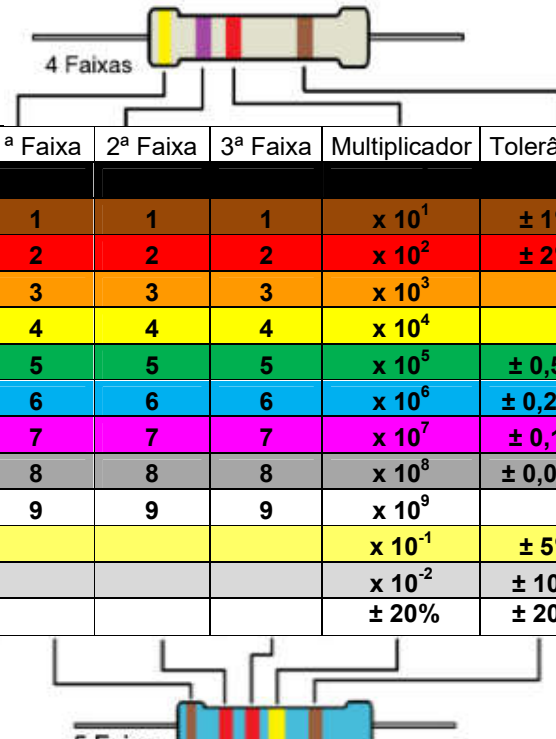
$$R = \frac{V_S - V_L}{I} \quad (2)$$

Como $V_S = 5$ V, $V_L = 3$ V e $I = 20$ mA, o valor necessário da resistência seria de $R = (5 - 3) / 0,02$, ou seja, um valor de 100 Ω . É prudente sempre escolher um resistor com valor mais próximo e maior ao valor necessário, caso contrário uma corrente muito intensa pode atravessar o resistor e danificar o LED (MCROBERTS, 2011). Mas, é comum o uso de resistores com valores bem

maiores, com 220Ω até $1 \text{ k}\Omega$ (ARDUINO BLINK, 2019), tendo como consequência apenas a diminuição da luminosidade emitida pelo LED.

Em geral os resistores são fabricados com valores-padrão e como são muito pequenos para conter informações de fácil leitura é utilizado um código de cores. Ao redor do resistor encontram tipicamente quatro anéis de cores e é possível descobrir o valor de sua resistência utilizando o código de cores contido no Quadro 1. Resistores com três, cinco ou seis faixas também são fabricados, mas são de uso muito mais restrito que aqueles com quatro faixas.

Quadro 1 – Código de cores dos resistores



| Cor | 1ª Faixa | 2ª Faixa | 3ª Faixa | Multiplicador | Tolerância |
|----------|----------|----------|----------|------------------|--------------|
| Marrom | 1 | 1 | 1 | $\times 10^1$ | $\pm 1\%$ |
| Vermelho | 2 | 2 | 2 | $\times 10^2$ | $\pm 2\%$ |
| Laranja | 3 | 3 | 3 | $\times 10^3$ | |
| Amarelo | 4 | 4 | 4 | $\times 10^4$ | |
| Verde | 5 | 5 | 5 | $\times 10^5$ | $\pm 0,5\%$ |
| Azul | 6 | 6 | 6 | $\times 10^6$ | $\pm 0,25\%$ |
| Violeta | 7 | 7 | 7 | $\times 10^7$ | $\pm 0,1\%$ |
| Cinza | 8 | 8 | 8 | $\times 10^8$ | $\pm 0,05\%$ |
| Branco | 9 | 9 | 9 | $\times 10^9$ | |
| Dourado | | | | $\times 10^{-1}$ | $\pm 5\%$ |
| Prata | | | | $\times 10^{-2}$ | $\pm 10\%$ |
| Nenhuma | | | | $\pm 20\%$ | $\pm 20\%$ |

Adaptado de: Mcroberts (2011, p. 48) e de Pakéquis⁶

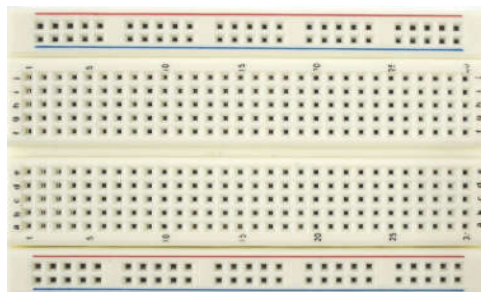
Assim, para um resistor de 220Ω é necessário um 2 na primeira faixa, que é vermelho, seguido por um 2 na faixa seguinte, também vermelho. A

⁶ Disponível em: <https://www.pakequis.com.br/p/calculadora-de-codigo-de-cores-de.html>. Acesso em: 05 abr. 2019.

terceira faixa se refere ao expoente da potência de 10 que deve ser multiplicada ao valor já computado nas duas primeiras faixas e que no caso é 10^1 . Ou seja, o expoente é 1, resultando em marrom na terceira faixa. A faixa final indica a tolerância do resistor. Caso o resistor tenha uma faixa dourada, ele tem uma tolerância de $\pm 5\%$ e isso significa que o valor, de fato, do resistor varia entre 209Ω e 231Ω . Dessa forma, um LED que requer 3 V de tensão e 20 mA de corrente, necessitará de um resistor com uma combinação de faixas vermelho, vermelho e marrom (MCROBERTS, 2011).

Para conectar um LED a um dos pinos digitais, será necessária também uma protoboard (Figura 12). Ela é uma placa muito usada para montagem de circuitos experimentais, pois é reutilizável e não necessita de realização de solda elétrica. O tipo mais comum e usado na maioria dos projetos é a que mede 16,5 cm por 5,5 cm e apresenta 840 furos (ou pontos) na placa. Os furos são conectados eletricamente por tiras de metal condutor alojados no interior da placa. Os furos das duas linhas de tiras paralelas ao topo e a base estão conectados eletricamente somente aos furos da mesma linha, e são projetados para carregar o barramento de alimentação e o barramento do terra. As tiras no centro correm a 90 graus dos barramentos de alimentação e do terra, e há um espaço vazio no meio para que se possa colocar circuitos integrados, de modo que cada pino do chip vá para um conjunto diferente de furos e, portanto, para um barramento diferente (MCROBERTS, 2011).

Figura 12 – Protoboard



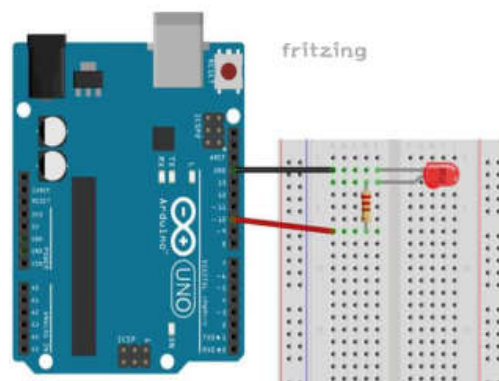
Fonte: Wikipédia⁷.

⁷ Disponível em: https://upload.wikimedia.org/wikipedia/commons/b/be/Protoboard_Unitec.jpg. Acesso em: 20 abr. 2019.

As conexões elétricas entre os dispositivos em geral são feitas com fios jumper. Eles são encontrados comercialmente com pontas moldadas para facilitar sua inserção na protoboard. Porém, é possível usar também pedaços de fios rígidos de núcleo único em que se retira cerca de 6 mm do material isolante de cada ponta (MCROBERTS, 2011).

A Figura 13 mostra o diagrama elétrico para acionar o LED. O jumper preto conecta o cátodo do LED ao pino GND. O ânodo do LED é conectado com o resistor de 220 Ω e este com o pino 10 do Arduino pelo jumper vermelho.

Figura 13 – Esquema de ligação de um LED externo ao Arduino



Adaptado de: Como Fazer as Coisas⁸.

Com as conexões elétricas concluídas, é necessário desenvolver o algoritmo no Arduino. Como exemplo, um código para fazer o LED conectado ao pino digital 10, repetidamente, ficar aceso por um segundo e em seguida ficar apagado por um segundo é apresentado abaixo:

```
int pinoLED 10

void setup() {
  pinMode(pinoLED, OUTPUT);
}
```

⁸ Disponível em: <http://www.comofazerascosas.com.br/projeto-arduino-pisca-led.html>. Acesso em: 21 abr. 2019.


```
void loop() {  
  digitalWrite(pinoLED, HIGH);  
  delay(1000);  
  digitalWrite(pinoLED, LOW);  
  delay(1000);  
}
```

Esse código tem uma única diferença em relação ao código para controle do *LED built-in*: a numeração da porta. E poderia até ser igual se o circuito fosse conectado ao pino do *LED built-in*. A reutilização do código ou de parte dele é uma das coisas mais úteis na programação do Arduino.

2.5.3 Eletroímã

Em abril de 1820, o cientista dinamarquês Hans Christian Orsted descobriu que se um fio condutor percorrido por uma corrente elétrica fosse colocado próximo a uma bússola, a sua agulha era defletida da posição de equilíbrio. Verificou assim que uma corrente elétrica ao atravessar um fio condutor cria ao seu redor um campo magnético (UNESP, 2019).

Enrolando-se um fio condutor de tal modo que forme uma sequência de voltas (espiras) muito próximas em forma de um tubo retilíneo produz-se uma bobina de fios chamado solenóide. Se por ele passar uma corrente elétrica, é gerado um campo magnético no sentido perpendicular à seção reta do solenóide. O resultado final é que o solenóide possui pólos norte e sul, tal como um ímã natural (UNESP, 2019)

O ferro doce (ferro puro) tem a propriedade de só se imantar enquanto estiver próximo de um campo magnético. Assim, usando um núcleo de ferro doce em um solenóide é criado um ímã temporário relativamente forte chamado eletroímã. Os eletroímãs apresentam inúmeras aplicações práticas tais como telégrafos, fones, alto falantes, telefones, campainhas e relés entre outras (USP,

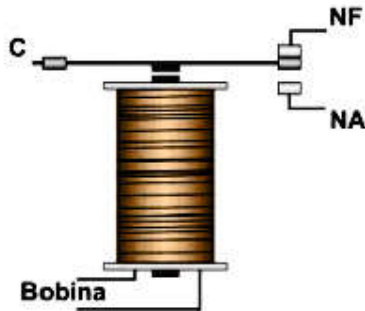
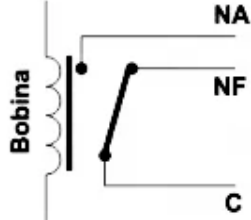
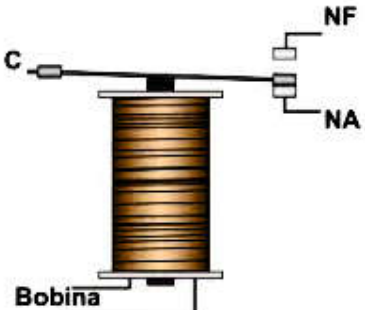
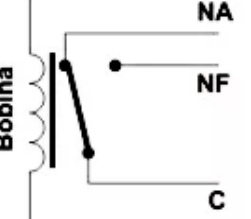
2019), mais aqui será dada atenção a apenas duas aplicações: o eletroímã solenóide de tracionamento e o relé eletromecânico.

O Eletroímã Solenóide de tracionamento ou de levantamento é um dispositivo capaz de utilizar corrente elétrica para gerar um campo magnético e atrair materiais magnéticos próximos a ele e controlar a movimentação e transporte do material. Possui uma larga faixa de capacidade de tração, existindo desde aqueles com capacidade de levantar uns poucos newtons até os usados para erguer cargas com várias toneladas. É possível construir um para controlar cargas pequenas, mas é muito mais simples, por exemplo, entrar com “Eletroímã Solenoide 2,5kg” ou “Imã Elétrico 12v Dc” em um buscador de páginas da internet e se obter uma lista de sites que possibilitam a aquisição de um deles.

Já um relé do tipo eletromecânico (EMR – *Electromechanical Relay*) é simplesmente um interruptor eletromecânico. Como mostra a Figura 14, ele é composto por um eletroímã, uma armadura (haste de material ferromagnético) e pelos contatos: comum (C), normalmente aberto (NA) e normalmente fechado (NF).

No caso da bobina estar desligada, C e NF estão conectados eletricamente. Quando uma tensão é aplicada na bobina, ela se magnetiza e atrai a armadura, levando C a se desconectar de NF e a se conectar com NA. Quando a bobina é desligada, ocorre o contrário, C se desconecta de NA e volta a ter contato com NF (EMBARCADOS LÂMPADA, 2019).

Figura 14 – Esquema de um relé EMR simples

| Situação do relé | Imagem | Esquema elétrico |
|------------------|---|---|
| Desligado |  |  |
| Acionado |  |  |

Fonte: <http://autocorerobotica.blog.br/controlando-lampadas-por-controle-remoto-com-arduino/>. Acesso em: 10 maio 2019.

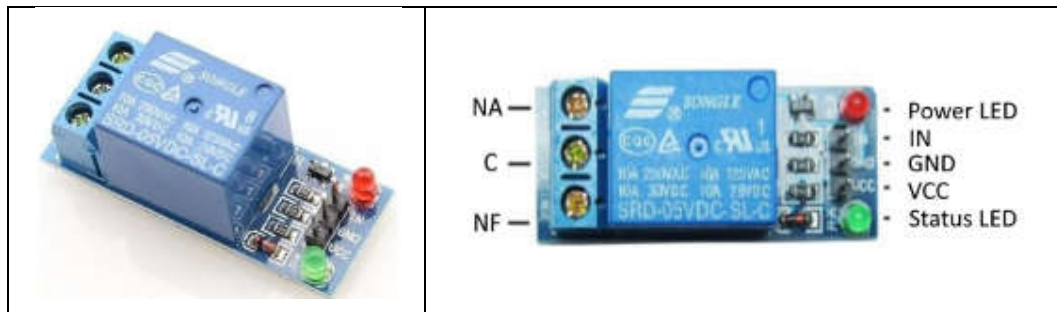
2.5.4 Controle de acionamento de um módulo relé

Quando for desejado acionar um dispositivo que utilize um valor de tensão superior aos 5 V ou correntes maiores que 40 mA fornecidos pela placa do Arduino Uno, é necessário utilizar alguma interface que suporte tal carga. Um relé seria uma opção, mas quando se liga e desliga uma bobina, ela induz correntes de valores bem superiores aos limites suportados pelo Arduino e para evitar isso é necessário o uso de circuito de proteção. Esse circuito, não é trivial de ser construído, mas pode ser encontrado pronto num módulo relé, que é um relé montado em uma placa junto com o circuito auxiliar para seu acionamento, e por isso é frequentemente usado com o Arduino (EMBARCADOS LÂMPADA, 2019).

Num módulo relé, além dos pinos C, NA e NF, existem os pinos VCC, GND e IN. Os pinos VCC e GND devem ser conectados aos pinos +5 V e GND do Arduino, respectivamente. Já o pino IN deve ser conectado à porta digital do Arduino responsável pelo controle de acionamento (VIDADESILÍCIO RELÉ, 2019).

Atualmente, são fabricados vários modelos de módulos relés com diversas quantidades de relés, cada um deles podendo controlar um circuito diferente dos outros. Cada circuito controlado por um relé é chamado canal. Comercialmente, encontram-se placas com 1 (Figura 15), 2, 4, 8 e até 16 canais (VIDADESILÍCIO RELÉ, 2019).

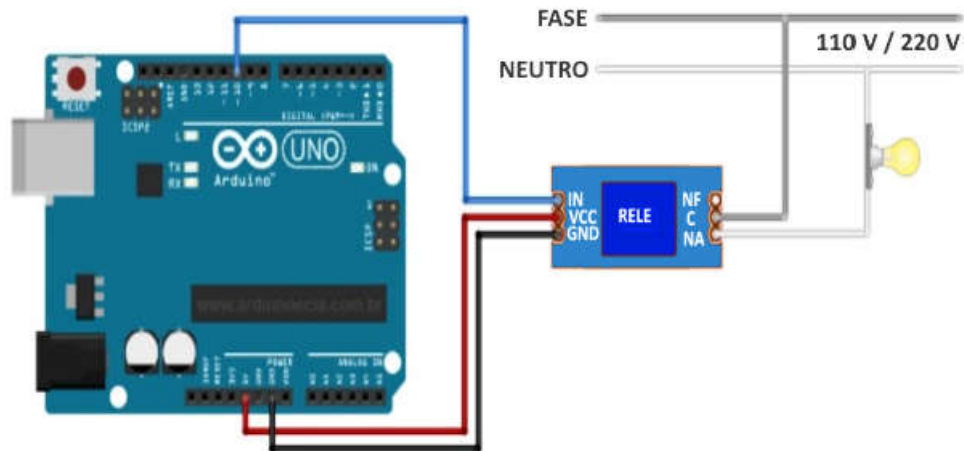
Figura 15 – Módulo relé de 1 canal



Fonte: <http://mekhos.com.br/mekhos/produto/modulo-relé-de-1-canal/>. Acesso em: 10 maio 2019.

Como exemplo, a Figura 16 mostra um esquema elétrico em que o pino 10 do Arduino é conectado ao pino IN do módulo relé para controlá-lo. A lâmpada tem um de seus terminais conectado ao contato NA do relé e o outro ao neutro da rede. Finalmente, o contato C do relé é conectado ao fase da rede. Desta forma, se for usado o algoritmo desenvolvido para o LED externo piscar, a lâmpada ficará acendendo e apagando com intervalo de um segundo.

Figura 16 – Acionamento de uma lâmpada por um módulo relé.



Fonte: Arduino e Cia.⁹

2.5.5 Controle da recepção de sinal do sensor de toque

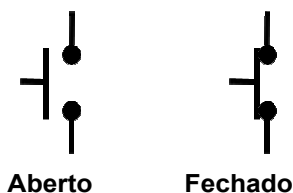
Roggia (2016, p. 23) define um sensor como um dispositivo sensível a uma forma de energia do ambiente (energia luminosa, cinética, sonora, térmica, entre outras), relacionando informações sobre uma grandeza física que precisa ser medida como luminosidade, temperatura, pressão, vazão, posição e corrente. E de acordo com a natureza do sinal que geram, os sensores podem ser classificados em digitais (discretos) e analógicos (contínuos).

Ainda segundo Roggia (2016, p. 23), os sensores digitais são utilizados para monitorar a ocorrência ou não de um determinado evento. Apresentam como sinal apenas dois estados: O primeiro, quando a corrente pode fluir por ele, é chamado de fechado ou ligado (ON) ou alto (HIGH) ou nível 1; O segundo, quando não há corrente passando por ele, é chamado de aberto ou desligado (OFF) ou nível baixo (LOW) ou nível 0.

⁹ Disponível em: <https://www.arduinoecia.com.br/ligando-uma-lampada-com-modulo-rele-arduino/>. Acesso em: 20 maio 2019.

Os sensores digitais são de ativação/desativação (ON/OFF) e por isso são chamados comutadores, pois comutam entre os estados aberto (desligado) e fechado (ligado), como mostrado na Figura 17.

Figura 17 – Estados de um comutador elétrico.

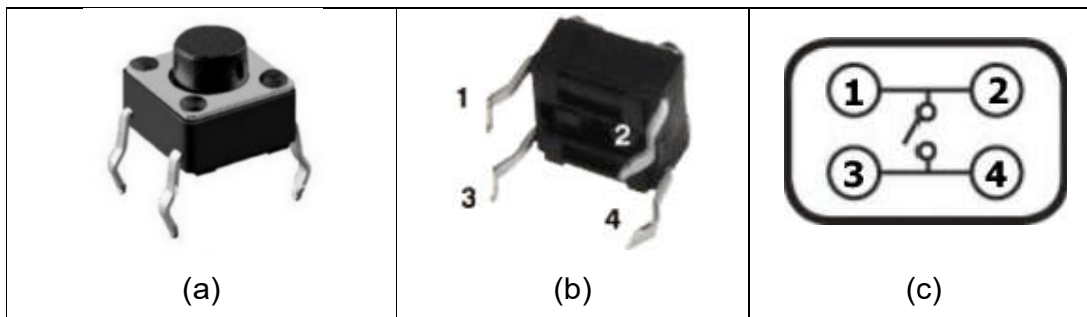


Adaptado de: Eletrônica para Artistas¹⁰

O comutador alternado é o tipo mais usado, mudando de um estado para outro apenas quando acionado. O Interruptor elétrico e a chave liga/desliga são seus exemplos mais comuns. Já os comutadores do tipo momentâneo, quando pressionados passam do estado aberto para o fechado e retornam ao estado anterior quando a pressão termina. Seus exemplos mais comuns são o botão da campainha elétrica e o botão de pressão ou *push button* (BANZI, 2011, p. 40).

O botão de pressão também é conhecido como botão momentâneo, botão tátil ou botão de apertar. O tipo mais comum tem quatro pinos e seu aspecto é mostrado nas Figura 18 (a) e Figura 18 (b) e seu símbolo elétrico é visto na Figura 18 (c).

Figura 18 – Aspecto e símbolo do botão de pressão de quatro pinos.



¹⁰ Disponível em: <http://eletronicaparaartistas.com.br/arduino-5-resistores-de-pull-up/>. Acesso em: 11 jul. 2019.

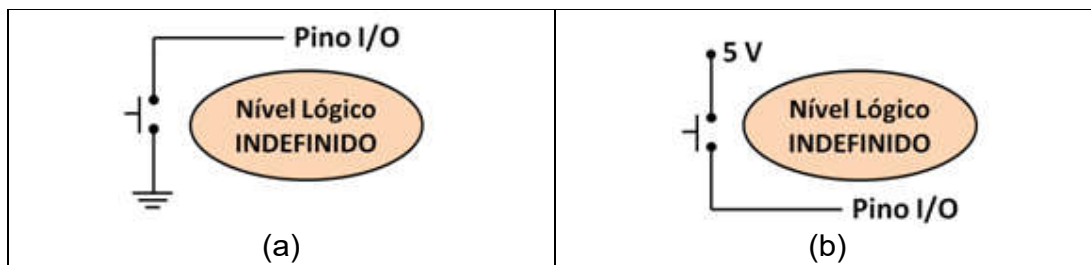
Adaptado de: Gueddes (2017, p. 44) e Kamei (2019, p. 7).

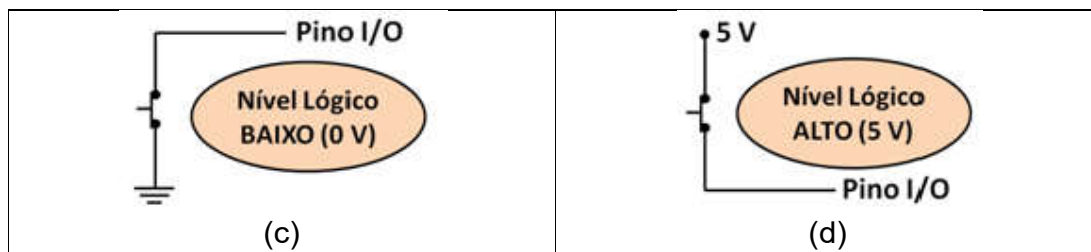
Com o botão não pressionado, só os pinos 1 e 2 estão em contato elétrico entre si, assim como os pinos 3 e 4. Os quatro pinos ficarão conectados eletricamente enquanto o botão for pressionado, (Kamei 2019, p. 8).

Um sensor digital é comumente conectado ao Arduino em um pino digital configurado como de entrada de dados. Nesse modo o pino é posto em um estado de alta impedância, equivalente a um resistor de 100 M Ω em série com o circuito a ser monitorado e consumindo corrente extremamente pequena. Isso significa que é necessária uma corrente muito baixa para mudar seu estado lógico de um valor para outro. E significa também que um pino configurado como entrada sem nada conectado a ele, ou com fios conectados a ele que não estão conectados a outros circuitos, estará em um estado lógico indefinido e seu nível de tensão apresentará alterações aparentemente aleatórias, captando ruído elétrico do ambiente ou acoplamento capacitivo do estado de um pino próximo (ARDUINO PINOS DIGITAIS, 2019).

Uma tentativa não eficiente de evitar que o estado do pino seja indefinido é mostrada na Figura 19 (a) e na Figura 19 (b), em que o botão não está pressionado e o pino encontra-se com estado indefinido. Com o botão pressionado, o estado do pino passa a ser LOW ou HIGH como é visto, respectivamente, na Figura 19 (c) e Figura 19 (d).

Figura 19 – Controle do nível lógico com botão de pressão.



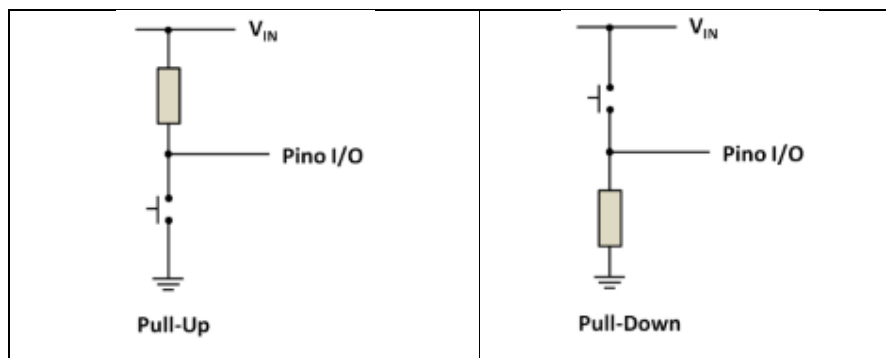


Adaptado de: <http://eletronicaparaartistas.com.br/arduino-5-resistores-de-pull-up/>.
Acesso em: 11 jul. 2019.

As conexões com resistores *pull-up* e *pull-down* mostradas na Figura 20 são as soluções mais simples para o pino nunca ficar em um estado indefinido.

Em ambas é usado um resistor geralmente de 10 K Ω conectado a um botão de pressão. Na ligação do tipo *pull-up*, enquanto o botão está livre o estado lógico do pino é HIGH e enquanto o botão está pressionado o pino está no estado LOW. Já na conexão *pull-down* ocorre o contrário, com o pino digital no estado LOW com o botão livre e HIGH com o botão pressionado (AUTOCOREROBOTICA, 2019).

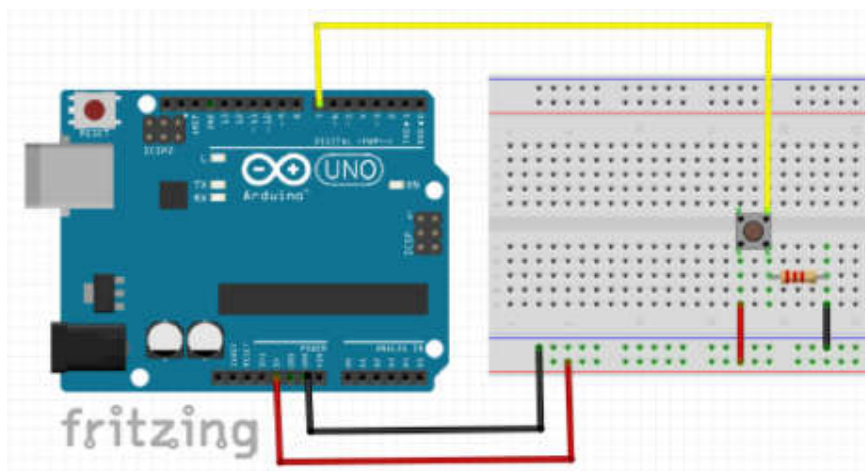
Figura 20 – Resistor de pull-up e resistor de pull-down.



Adaptado de: <https://www.embarcados.com.br/pinos-digitais-do-arduino/>.
Acesso em: 15 jul. 2019.

Em Arduino Button (2019) é apresentado um projeto que usa uma ligação *pull-down* conectada ao pino digital 7 para controlar o *LED built-in* e a Figura 21 mostra o esquema elétrico do circuito.

Figura 21 – Botão pull-down para controlar acionamento do *LED built-in*.



Adaptado de: ARDUINO BUTTON (2019).

Num *sketch* do Arduino, se for desejado que o pino digital de número `val_pino` seja usado para fazer a entrada de dados, ele deve ser previamente configurado no `setup` com a instrução `pinMode(val_pino, INPUT)`, e para ler seu valor lógico deve ser usada a instrução `digitalRead(val_pino)`. O retorno dessa instrução será **HIGH** ou **LOW**. Em Arduino Digital Read (2019) é encontrado um exemplo de código para controle do circuito mostrado na Figura 21:

```
int ledPin = 13; // Atribui a variável ledPin o valor do pino do LED built-in.
int inPin = 7;   // Atribui a variável inteira inPin o valor do pino digital 7.
int val = 0;    // Atribui a variável inteira val o valor 0.

void setup() {
  pinMode(ledPin, OUTPUT); // Configura o pino 13 como saída de dados
  pinMode(inPin, INPUT);  // Configura o pino 7 como entrada de dados
}

void loop() {
  val = digitalRead(inPin); // Lê o pino de entrada
  digitalWrite(ledPin, val); // Aciona o LED com o valor lido do botão
}
```

Este programa é bem simples: inicialmente são criadas as variáveis inteiras **ledPin** com valor 13, **inPin** com valor 7 e **val** com valor 0. Dentro do `setup` do *sketch*, **ledPin** é configurada para saída de dados e **inPin** para entrada

de dados. Finalmente, na primeira instrução do loop, **val** recebe constantemente o valor do estado de **inPin** e na segunda instrução este valor é colocado em **ledPin**.

Pode-se escrever uma versão simplificada para esse algoritmo, sem uso de variáveis. Nessa versão, o valor do pino 7 é colocado diretamente como valor do pino 13.

```
void setup() {  
  pinMode(13, OUTPUT);  
  pinMode(7, INPUT);  
}  
  
void loop() {  
  digitalWrite(13, digitalRead(7));  
}
```

É importante observar que em um *sketch* mais extenso, em que varias portas são usadas em muitos comandos, é muito comum a necessidade de troca de uma porta ou de varias delas e o uso de variáveis torna muito mais simples a realização dessas mudanças.

Em Arduino Button (2019) e Arduino Pushbutton (2019) encontram-se outras versões para este algoritmo que usam um teste lógico para saber se o botão está pressionado e acender o LED caso o resultado seja verdadeiro com nível lógico HIGH, ou apagar o LED em caso contrário, com nível lógico LOW. A instrução **if** serve para verificar se uma condição é verdadeira ou falsa e, segundo Arduino IF (2019), sua sintaxe é:

```
if (condição) [bloco de instruções];
```

A condição deve sempre estar entre parêntesis. Ela é um teste lógico que só possui dois resultados possíveis: **true** (verdadeiro) ou **false** (falso). O bloco de instruções dentro dos colchetes somente será executado se o resultado da condição for **true**.

Caso seja necessário executar alguma instrução somente se o resultado do teste for falso, deve-se, segundo Arduino Else (2019), usar:

```
if (condição) [bloco de instruções]
else [bloco de instruções];
```

O uso de **if..else** traz maior controle sobre o fluxo do código quando comparado com a instrução **if** sozinha. Permite também que vários testes sejam agrupados. O **else** pode realizar outro teste **if**, criando a possibilidade de executar simultaneamente vários testes mutuamente exclusivos.

Assim, uma versão reduzida do *sketch* com uso de teste lógico, encontrado em Arduino Button (2019), pode ser

```
void setup() {
  pinMode(13, OUTPUT;
  pinMode(7, INPUT);
}

void loop() {
  if (digitalRead(7) == HIGH) [digitalWrite(13, HIGH)]
  else [digitalWrite(13, LOW)];
}
```

Em um teste lógico de igualdade é necessário usar dois sinais seguidos de igual, pois na linguagem do Arduino um único sinal de igual é usado para realizar atribuição de variáveis (ARDUINO IF, 2019).

É fácil perceber que no projeto que usa uma ligação *pull-down* para controlar o *LED built-in* o teste condicional além de ser opcional, traz uma complexidade desnecessária. Porém, existem incontáveis situações em que seu uso é indispensável.

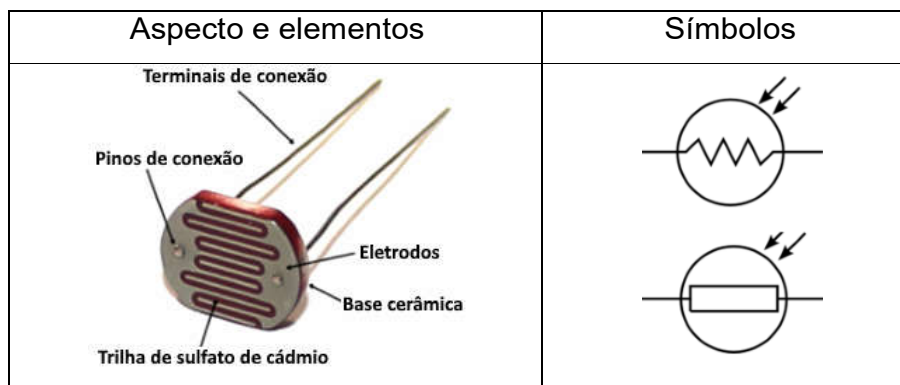
2.5.6 Controle da recepção de sinal do sensor de luz

De acordo com Roggia (2016, p. 23), além do sensor digital, existe o sensor analógico que serve para medir uma grandeza analógica, ou seja, a grandeza física com valor compreendido entre um mínimo e um máximo de uma faixa contínua de valores. O sensor digital apresenta em sua saída um sinal de tensão, corrente ou resistência proporcional ao valor da grandeza física monitorada.

Existem diversos tipos de sensores analógicos e entre eles, Braga (2005, p. 106) define sensores resistivos como transdutores que apresentam a característica de mudar sua resistência elétrica na presença de um sinal de energia qualquer, normalmente luz, temperatura ou pressão e convertendo-o em um sinal elétrico analógico.

Segundo Braga (2005, p.107), o tipo mais comum de sensor resistivo de luz é o LDR (Ligh Dependent Resistor), também conhecido como fotorresistor ou ainda célula de sulfeto de cádmio (CdS). Os LDRs têm diferentes formas, tamanhos e valores de mínima e máxima resistência, com aspecto mais comum e símbolos mostrados na Figura 22.

Figura 22 – Aspecto, elementos e símbolos de um LDR.



Adaptado de: Bóson Treinamentos¹¹ e Proesi¹².

¹¹ Disponível em: <http://www.bosontreinamentos.com.br/eletronica/curso-de-eletronica/curso-de-eletronica-o-que-e-um-ldr-light-dependent-resistor/>. Acesso em: 21 jul. 2019.

¹² Disponível em: <https://proesi.com.br/sensor-ldr-10mm.html>. Acesso em: 21 jul. 2019.

Um LDR é constituído por uma base de cerâmica isolante sobre a superfície da qual se encontram duas placas condutoras separadas por uma fina trilha ondulada de sulfeto de cádmio. Cada placa está conectada a um dos dois eletrodos que constituem os terminais de conexão do LDR.

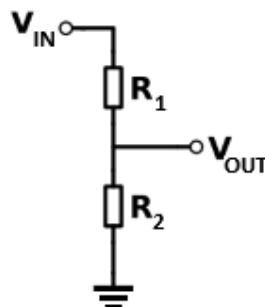
O sulfeto de cádmio apresenta resistividade extremamente alta na escuridão total, levando a resistência do LDR a ter de milhões de ohms, mas como essa substância é sensível a todo espectro de luz visível, o aumento da luminosidade incidente sobre ela produz a liberação cada vez maior de grandes quantidades de portadores de cargas elétricas, diminuindo drasticamente sua resistividade, o que faz a resistência do LDR atingir algumas dezenas de ohms. Assim, ele permite detectar se o ambiente está claro, escuro ou com qualquer valor de luminosidade. É importante observar que o LDR, ao contrário do LED, não tem polaridade (BRAGA, 2005, p. 107).

Como a resistência do LDR é uma grandeza analógica, para medi-la com o Arduino é necessário usar uma porta analógica. Diferentemente de um pino digital, não é necessário definir um pino analógico como de entrada ou saída e para ler o seu valor é utilizado o comando ***analogRead(num_pino)*** em que *num_pino* é um dos seis pinos analógicos, ou seja, A0, A1, A2, A3, A4 ou A5. Cada pino analógico tem um conversor analógico-digital de 10 bits, o que corresponde a 2^{10} valores inteiros. Assim, um pino analógico ao receber voltagens entre 0 e 5 volts, converte-as em valores inteiros entre 0 (0 V) e 1023 (5 V) (MCROBERTS, 2011, p. 76).

Embora a principal função dos pinos analógicos nas aplicações em geral seja a leitura de sensores analógicos, eles também possuem toda a funcionalidade de pinos de entrada/saída de uso geral (GPIO), ou dos pinos digitais de 0 a 13. Assim, caso seja preciso de mais pinos digitais para uso geral, e nenhum pino analógico estiver em uso, os pinos analógicos poderão ser usados como se fossem GPIO (ARDUINO PINOS ANALOGICOS, 2019).

Praticamente todos os circuitos sensores usam de algum modo, um divisor de tensão ou divisor de potencial. Um divisor de tensão padrão utiliza dois resistores ligados em série, como mostra a Figura 23 (NETTO, 2019, p. 52).

Figura 23 – Circuito do divisor de tensão.



Adaptado de: Nerd Elétrico¹³.

Sendo V_1 e V_2 as tensões em R_1 e R_2 , a voltagem de entrada V_{IN} será reduzida (dividida) para uma tensão de saída V_{OUT} , assim:

$$V_{IN} = V_1 + V_2 \quad (3)$$

Se i_1 e i_2 são as correntes em R_1 e R_2 , é fácil perceber que $V_1 = i_1 \cdot R_1$ e $V_{OUT} = V_2 = i_2 \cdot R_2$ e admitindo que $i_{IN} = i_1 \approx i_2$, ou seja, que pouca corrente é drenada entre R_1 e R_2 , ou que $i_{OUT} \ll i_{IN}$, encontra-se:

$$V_{IN} = (R_1 + R_2) \cdot i_{IN} \quad (4)$$

Mas como:

$$i_{IN} \approx i_2 = \frac{V_2}{R_2} = \frac{V_{OUT}}{R_2} \quad (5)$$

Então:

¹³ Disponível em: <http://nerdeletrico.blogspot.com/2011/04/circuito-divisor-de-tensao.html>. Acesso em: 24 jul. 2019.

$$V_{IN} = (R_1 + R_2) \cdot \frac{V_{OUT}}{R_2} \quad (6)$$

Assim:

$$V_{OUT} = \frac{R_2}{R_1 + R_2} V_{IN} \quad (7)$$

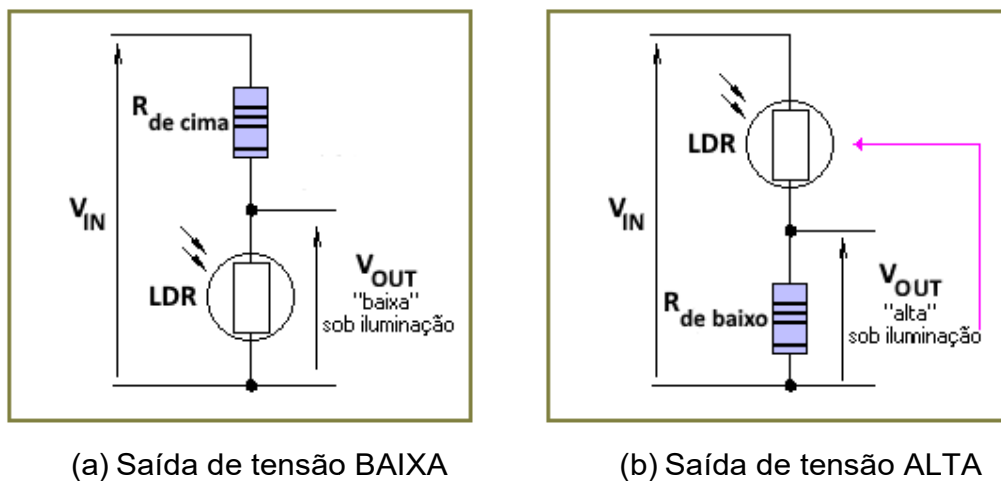
Logo:

$$V_{OUT} = \frac{V_{IN}}{1 + R_1/R_2} \quad (8)$$

Mostrando que é a razão entre os valores dos resistores e não os seus valores em si que importam para determinar V_{OUT} (MCROBERTS, 2011, p. 118).

Substituindo R_1 ou R_2 por um LDR no divisor de tensão, pode-se obter duas situações, como mostra a Figura 24. Na situação (a), obtém-se na saída uma tensão BAIXA quando o LDR está intensamente iluminado e uma tensão ALTA quando o LDR é posto no escuro. Já na situação (b), obtém-se o contrário, ou seja, uma tensão ALTA quando o LDR está intensamente iluminado e uma tensão baixa quando o LDR é posto no escuro. A escolha de uma delas só depende da leitura desejada no sensor (NETTO, 2019, p. 53-54).

Figura 24 – Divisor de tensão com LDR.



Adaptado de: Netto 2019, p. 53-54.

Como exemplo, a Tabela 1 mostra as voltagens obtidas conforme a luminosidade sobre o LDR se altera no modo de tensão de saída baixa, usando $V_{IN} = 5\text{ V}$ e um LDR com resistência de $100\text{ k}\Omega$ no escuro e $10\text{ k}\Omega$ quando bem iluminado.

Tabela 1 – Valores de V_{OUT} para um LDR com $V_{in} = 5\text{ V}$.

| R_1 | R_2 (LDR) | V_{OUT} | Luminosidade |
|---------------------|----------------------|-----------|--------------|
| $10\text{ k}\Omega$ | $100\text{ k}\Omega$ | 4,54 V | Mais escuro |
| $10\text{ k}\Omega$ | $73\text{ k}\Omega$ | 4,39 V | 25% |
| $10\text{ k}\Omega$ | $45\text{ k}\Omega$ | 4,09 V | 50% |
| $10\text{ k}\Omega$ | $28\text{ k}\Omega$ | 3,68 V | 75% |
| $10\text{ k}\Omega$ | $10\text{ k}\Omega$ | 2,5 V | Mais claro |

Fonte: Mcroberts, 2011, p. 119.

Em Vidadesilício LDR (2019) é apresentado um projeto simples capaz de detectar a luminosidade e acender o LED em um ambiente escuro, e o apagar em um ambiente claro. Como mostra a Figura 25, o LED está conectado com a porta digital 7 e em série com o um resistor de $300\ \Omega$. O divisor está conectado com a porta analógica A0, tem saída de tensão baixa sob iluminação e possui além do LDR, um resistor de $10\text{ k}\Omega$. Também foi definido que, uma leitura da porta analógica entre 0 e 500, indica muita luminosidade ambiente e o LED deve ser apagado. Já uma leitura entre 501 e 1023, significa pouca luminosidade e o LED deve ser aceso. O *sketch* do projeto é:

```
int ldrValor = 0; //Valor lido do LDR

void setup() {
  pinMode(7,OUTPUT); //define a porta 7 como saída
}

void loop() {
  ldrValor = analogRead(A5); //Lê valor do LDR: valor entre 0 e 1023
  if (ldrValor > 500) {
    digitalWrite(ledPin,HIGH); //Liga o LED se o valor > 500
  }
  else {
    digitalWrite(ledPin,LOW); // senão, apaga o LED
  }
}
```

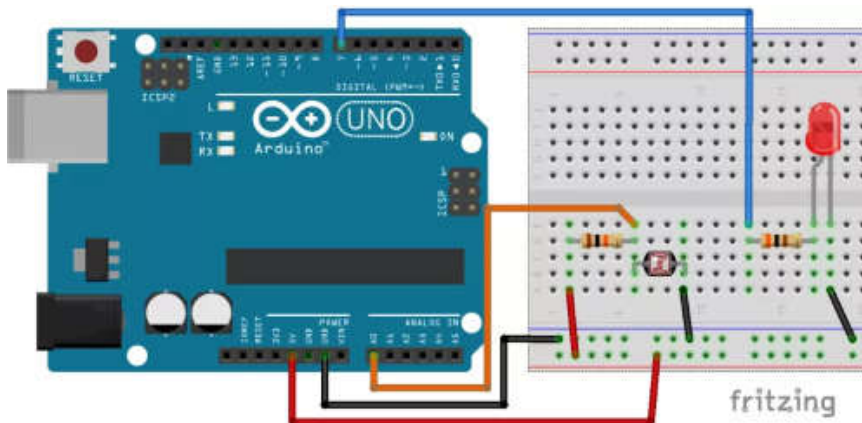


```

}
delay(100);
}

```

Figura 25 – Controle do acionamento de um LED com um LDR.



Adaptado de: <https://portal.vidadesilicio.com.br/sensor-de-luz-com-ldr/>.
Acesso em: 25 jul 2019.

Nesse algoritmo, primeiro é criada a variável inteira **ldrValor** para armazenar o valor da leitura do LDR e com valor inicial nulo. A seguir, dentro do setup, a porta digital 7 é configurada para saída de dados. Depois, dentro do loop, a primeira instrução faz a leitura da porta analógica e armazena seu valor em **ldrValor**. A instrução seguinte realiza o teste lógico: Se o valor de **ldrValor** for maior que 500 o LED será aceso, caso contrário será apagado. Por último, ocorre uma pequena espera de 100 ms antes do reinício do loop.

2.5.7 Transmissão via porta serial

Toda placa Arduino possui pelo menos uma porta serial, também chamada UART¹⁴ ou USART¹⁵. Essa porta serve para fazer a comunicação entre o Arduino e um computador através da conexão USB. Além disso, como está conectada aos pinos RX (de recepção) e TX (de transmissão), usa níveis

¹⁴ Acrônimo de Universal Asynchronous Receiver/Transmitter e significando Receptor/Transmissor Universal Assíncrono.

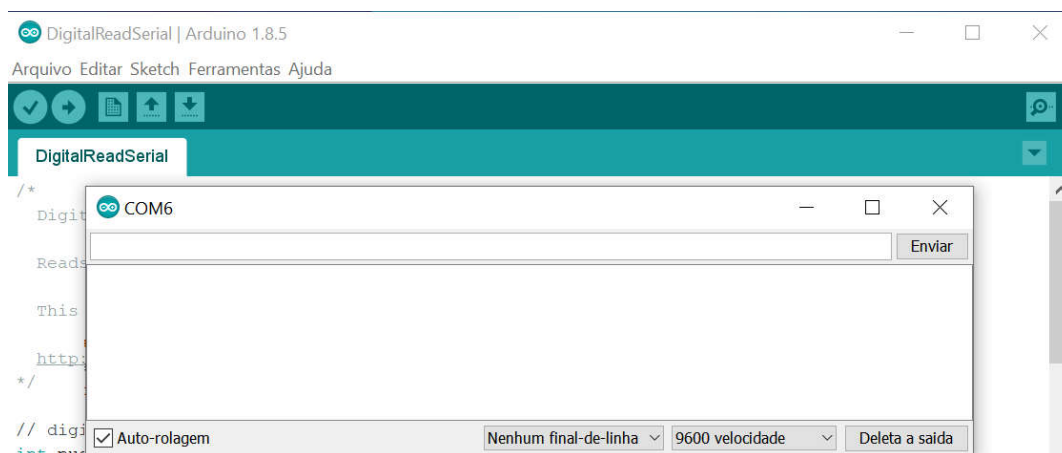
¹⁵ Acrônimo de Universal Synchronous Asynchronous Receiver Transmitter e significando Transmissor/Receptor Universal Síncrono e Assíncrono.

lógicos TTL¹⁶ para fazer comunicação com outros dispositivos. Na placa Uno, o RX corresponde ao pino 0 e o TX ao pino 1. Como o upload do computador para o Arduino também é realizado pela conexão USB, durante sua realização nenhum outro dispositivo deve estar conectado ao RX ou ao TX, pois pode ocorrer interferência nessa comunicação, causando falhas no upload (ARDUINO SERIAL, 2019).

A plataforma Arduino possui diversas funções para manipulação da comunicação serial. Segundo Embarcados Serial (2019) as principais são: **Serial.begin(taxa)**, usada para iniciar a porta serial e definir a taxa de velocidade para transmissão de dados seriais em bits por segundo; **Serial.available()**, usada para obter o número de bytes (caracteres) disponíveis para leitura na porta serial; **Serial.read()**, que permite ler o byte mais recente apontado no buffer de entrada da serial; **Serial.print(texto)**, que permite escrever um texto na serial; e **Serial.println(texto)**, que além de escrever um texto na serial, acrescenta ao seu final o caractere de fim de linha e o caractere de nova linha.

Ainda, segundo Embarcados Serial (2019), o monitor serial Integrado a IDE do Arduino é uma das interfaces mais simples para fazer comunicação serial. Seu uso exige que o Arduino esteja devidamente conectado ao computador e para iniciá-lo, basta clicar no botão com ícone de lupa localizado na parte superior da janela e à direita da barra de ferramentas da IDE (Figura 26).

¹⁶ Acrônimo de Transistor-Transistor Logic e significando Lógica Transistor-Transistor.

Figura 26 – Monitor serial integrado a IDE do Arduino.

Fonte: print screen da IDE no sistema operacional Windows 10.

Após a abertura do monitor, deve-se usar o botão de ajuste da velocidade de troca de dados, localizado na parte inferior direita da interface, para configurar a velocidade com o mesmo valor definido no *sketch* do Arduino. Para enviar dados é preciso fazer sua inserção na caixa de envio, localizada na parte superior da janela do monitor e acionar o botão enviar ou a tecla enter. Os dados de retorno da placa para o computador, caso existam, serão exibidos na caixa localizada no centro da janela do monitor.

Em *Arduino Read Serial* (2019) é encontrado um exemplo de aplicação para monitorar o estado de um botão momentâneo usando a comunicação serial. O circuito é praticamente igual ao mostrado na Figura 25Figura 20, tendo como única diferença a substituição do pino 7 pelo pino 2.

O *sketch* sem os comentários para realizar a comunicação serial é:

```
int pushButton = 2;

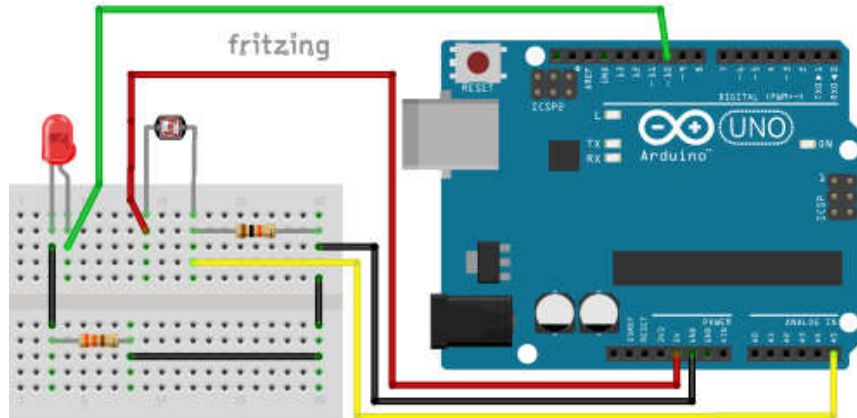
void setup() {
  Serial.begin(9600);
  pinMode(pushButton, INPUT);
}

void loop() {
  int buttonState = digitalRead(pushButton);
  Serial.println(buttonState);
  delay(1);
}
```

Inicialmente é criada a variável inteira **pushButton** com valor 2, o do pino de leitura do estado do botão. No setup, a velocidade de comunicação serial entre a placa e o computador é configurada para 9600 bits de dados por segundo. Depois o pino digital com valor definido na variável **pushButton** é configurado como uma entrada de dados. No loop, o estado do pino é lido e seu valor colocado na variável inteira **buttonState**. Em seguida, usando o comando **Serial.println**, este valor é enviado para a porta serial. Assim será escrito na caixa central do monitor serial uma sequência de "0"s, um abaixo do outro, se o botão estiver aberto ou de "1"s em caso contrário. Por último, A instrução **delay(1)** é usada para criar um atraso de um milissegundo entre as leituras e gerar estabilidade no programa.

Em outro exemplo, encontrado em Arduino LDR & LED (2019), similar ao encontrado em Vidadesilício LDR (2019) e cujo circuito elétrico é apresentado na Figura 27, um LDR controla o acionamento de um LED, acendendo-o quando a leitura na porta A5 é maior que 800 e apagando-o em caso contrário. Além disso, escreve na serial o valor encontrado na porta analógica.

Figura 27 – Circuito para monitorar o estado de um botão momentâneo.



Adaptado de Arduino LDR & LED (2019).

O seu *sketch* sem os comentários é:

```
int LedPin = 10;
int sensorPin = A5;
int sensorValor = 0;
```

```
void setup() {
  Serial.begin(9600);
  pinMode(LedPin, OUTPUT);
}
```

```
void loop() {
  int sensorValor = analogRead(sensorPin);
  Serial.println(sensorValor);
  if (sensorValor > 800) {
    digitalWrite(LedPin, HIGH);
  }
  else {
    digitalWrite(LedPin, LOW);
  }
}
```

No algoritmo, primeiro é criada a variável inteira **LedPin**, com valor 10, o do pino de leitura do estado do botão. Depois é criada a variável inteira **sensorPin**, com valor A5, o pino analógico que será usado. Em seguida é criada a variável inteira **sensorValor** para armazenar o estado do sensor e inicialmente com valor nulo. No setup, a velocidade de comunicação serial entre a placa e o

computador é configurada para 9600 bits de dados por segundo. Depois o pino digital com valor definido na variável **sensorValor** é configurado como uma entrada de dados. No loop, o estado do pino é lido e seu valor colocado na variável **sensorValor**. Em seguida, usando o comando **Serial.println**, este valor é enviado para a porta serial. Por fim um teste é realizado e o led é aceso se o valor da leitura do sensor for maior que 800 e apagado em caso contrário.

2.5.8 Transmissão via Bluetooth

Bluetooth é uma tecnologia desenvolvida pela Ericsson em 1994. Usa ondas de rádio com frequências entre 2,4 GHz e 2,483 GHz. O objetivo inicial era fazer conexão sem fio (*wireless*) em substituição aos cabos que conectam a curta distância entre computadores e seus periféricos. Posteriormente, ganhou o suporte de empresas tais como Intel, IBM, Toshiba, Nokia, Lucent, Motorola entre outras e juntas criaram o *Bluetooth Special Interest Group* (SIG), para ser responsável por escrever as especificações do protocolo e lançar versões atualizadas e novas funcionalidades. Atualmente, suas aplicações incluem o controle de mídia (áudio, vídeo e imagens) e de sistemas de comunicações com diversos dispositivos tais como fones de ouvido, equipamentos de som, TVs, home theaters, computadores, telefones celulares, câmeras digitais, dispositivos de interface humana (mouses, joysticks, teclados), entre outras (TELECO, 2019).

Os principais requisitos que norteiam o desenvolvimento do Bluetooth são o baixo consumo de energia, o baixo custo, a capacidade de transmissão de voz, dados e sinalização e por último, uma área de cobertura pequena, entre um e cem metros, sendo por isso classificada como rede sem fio PAN (*Personal Area Network*), que não precisa de autorização governamental para ser utilizada (TELECO, 2019).

Ainda, segundo Teleco (2019), uma rede Bluetooth é chamada *piconet*. Ela é composta por um dispositivo mestre e até sete outros dispositivos,

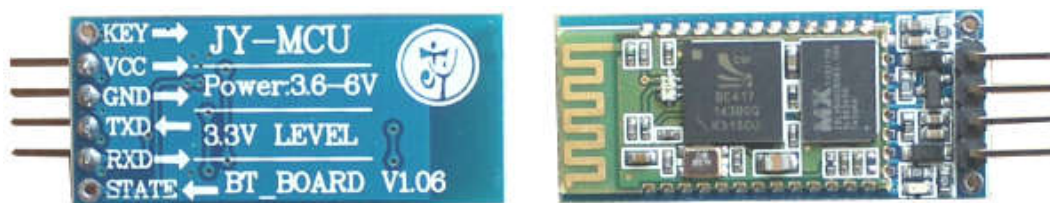
chamados escravos. O mestre pode solicitar e aceitar conexão (pareamento), enquanto o escravo só pode aceitar. Assim, toda comunicação ocorre somente entre o mestre e os escravos, não existindo comunicação direta entre dispositivos escravos.

Devido à facilidade de uso e preço relativamente baixo, o módulo Bluetooth HC-06 é um dos mais usados com Arduino. O HC-06 apresenta transmissão full duplex, ou seja, pode transmitir e receber ao mesmo tempo. Trabalha apenas em modo escravo com versão 2.0+EDR e o perfil SPP (*Serial Port Profile*). Tem potência de 2,5 mW e alcance aproximado de 10 metros (ELETROGATE, 2019).

Como mostra a Figura 28, ele apresenta quatro pinos:

- VCC que deve receber tensão de alimentação entre 3,6 V e 6,0 V;
- GND ou pino terra;
- TXD que envia sinais de transmissão com nível de tensão 3,3 V;
- RXD que recebe sinais de transmissão com nível de tensão de 3,3 V.

Figura 28 – Módulo HC-06.



Adaptado de: https://www.fecegypt.com/uploads/dataSheet/1522750308_bluetooth%20module%20hc06.pdf, p. 1. Acesso em: 28 jul. 2019.

O Arduino recebe sem problemas um sinal de 3,3 V vindo do pino TXD como nível lógico alto. Já no sentido de comunicação oposto, é necessário converter os 5 V saídos do pino TX do Arduino nos 3,3 V suportados pelo pino RXD. Para fazer essa redução a maneira mais simples e barata é usar um divisor de tensão em que, conforme a eq. 8, a resistência R_2 seja o dobro de R_1 ,

pois essa é a razão aproximada entre a tensão de 3,3 V no pino RXD e a queda de tensão entre os pinos do Arduino e do HC-06:

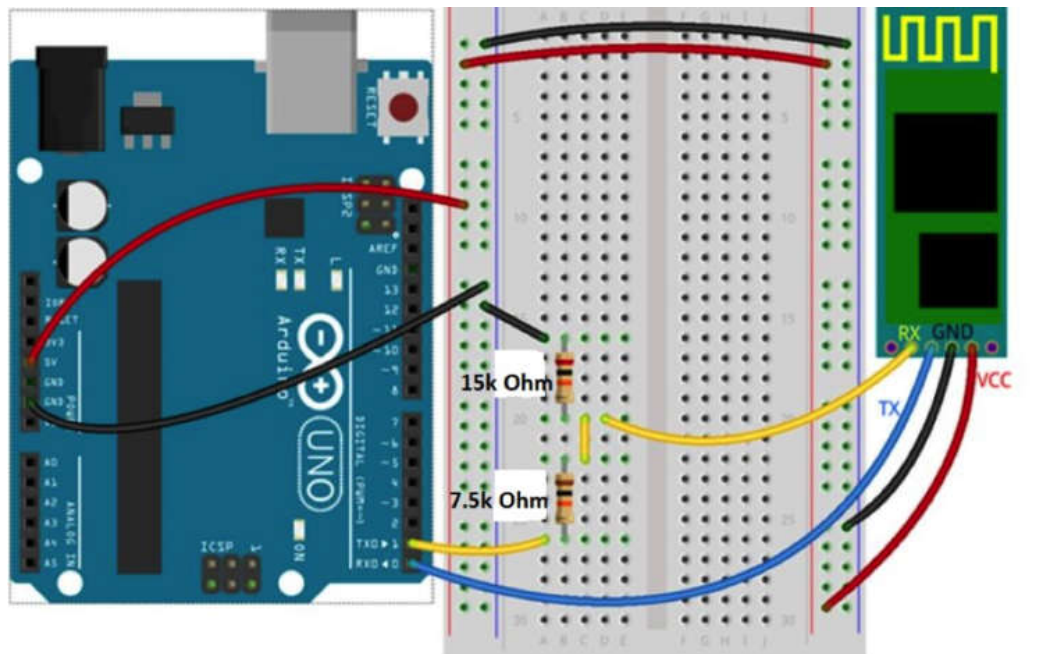
$$\frac{V_{out}}{V_{in} - V_{out}} = \frac{\frac{V_{in}}{1 + R_1/R_2}}{V_{in} - \frac{V_{in}}{1 + R_1/R_2}} = \frac{R_2}{R_1} \quad (9)$$

$$\frac{V_{out}}{V_{in} - V_{out}} = \frac{3,3 \text{ V}}{5 \text{ V} - 3,3 \text{ V}} = \frac{3,3}{1,7} \approx \frac{2}{1} \quad (10)$$

Por isso são usados para R_1 e R_2 , respectivamente, resistores de 1 k Ω e 2,2 k Ω em Eletrogate (2019), ou de 7,5 k Ω e 15 k Ω em Wustl (2019), ou ainda de 10 k Ω e 20 k Ω em ENGINEERS (2019).

A Figura 29 mostra um circuito que usa o HC-06 como interface de conexão Bluetooth para enviar comandos de um smartphone para o Arduino ativar e desativar o LED *on-board* em sua porta digital 13.

Figura 29 – Circuito para acionamento da placa HC-06.



Fonte: [https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_\(HC-06\)_%2B_Arduino](https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_(HC-06)_%2B_Arduino). Acesso em: 30. Jul. 2019.

Em Filipeflop (2019) encontra-se o *sketch* do projeto:

```
char buf;

void setup() {
  pinMode(13, OUTPUT);
  Serial.begin(9600);
}

void loop() {
  while(Serial.available() > 0) {
    buf = Serial.read();
    if (buf == 'L') {
      digitalWrite(13, HIGH);
    }
    if (buf == 'D') {
      digitalWrite(13, LOW);
    }
  }
}
```

A primeira linha do algoritmo cria a variável **buf** que irá armazenar os valores lidos na porta serial. No setup, o pino 13 é configurado para saída e a taxa de comunicação da serial estipulada em 9600. No loop, a primitiva **while** verifica se existe algum valor a ser lido na porta serial e se existir coloca este valor na variável **buf**. Em seguida, no primeiro **if**, verifica se ela é igual a 'L', colocando o pino 13 em HIGH e acendendo o *LED on-board*. Já o segundo **if** verifica se **buf** tem valor 'D', colocando o pino 13 em LOW e desligando o LED em caso verdadeiro.

Como no Arduino Uno as comunicações com o computador e com o módulo Bluetooth utilizam os mesmos pinos, somente após o final do upload do *sketch* é que devem ser conectados os pinos TX e RX do Arduino aos pinos RX e TX do HC-06. Para evitar isso, pode-se incluir no *sketch* uma chamada a biblioteca externa **SoftwareSerial**, nativa da IDE Arduino, para simular por software uma porta serial em dois outros pinos do Arduino e manter livre sua serial em hardware para ser utilizada no debug e gravação do código a ser executado (VIDADESILÍCIO HC06, 2020).

Quando o HC-06 está energizado, seu LED pisca repetidamente, indicando que o mesmo está aguardando conexão com um dispositivo mestre. Neste momento, utilizando algum aplicativo para gerenciamento de conexão Bluetooth instalado num celular, ao acessar a lista de varredura de dispositivos Bluetooth no aplicativo, deve-se encontrar o nome HC-06. Ao selecioná-lo, caso seja a primeira tentativa de conexão, será solicitada uma senha (que é 1234 por padrão). Com o pareamento estabelecido, o LED para de piscar indicando que a placa está apta a transmitir e receber dados (VIDADESILÍCIO HC06, 2020).

3 SERIAL BLUETOOTH TERMINAL

3.1 USANDO O APP SERIAL BLUETOOTH TERMINAL

Para evitar a nada simples tarefa de desenvolver um aplicativo de comunicação entre o smartphone e o Arduino foi utilizado nos experimentos o aplicativo Serial Bluetooth Terminal. Simples de ser obtido e instalado no smartphone, esse app (aplicativo) para Android utiliza uma interface muito fácil e amigável para enviar e receber dados por conexão Bluetooth. Ele é completamente gratuito, foi desenvolvido por Kai Morich e seu download pode ser feito pelo serviço de distribuição digital de aplicativos Google Play¹⁷.

O app utiliza uma interface bem simples e intuitiva tipo terminal/console orientado a linha para comunicação de dispositivo Bluetooth pareado a um smartphone Android (MORICH, 2019).

Ele suporta comunicação com diferentes dispositivos Bluetooth e utiliza diferentes versões do Bluetooth, mas no presente trabalho foi apenas utilizada a comunicação com a placa HC-06 com a versão do Bluetooth clássico.

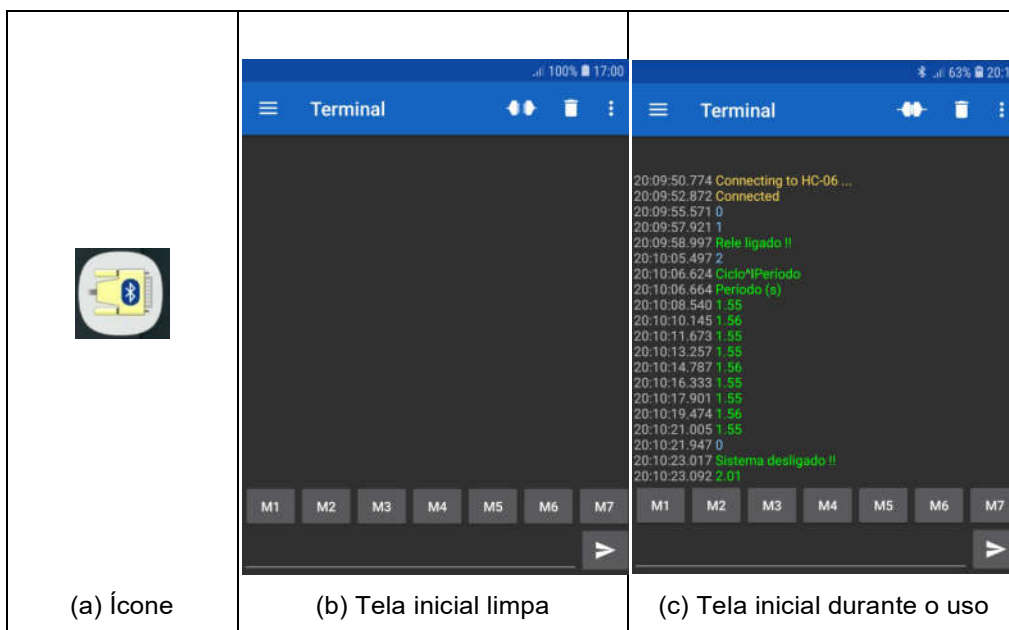
Após a instalação do app, o seu ícone, mostrado na 30 (a), aparece na tela do smartphone.

Ao abrir o app, entra-se na tela do **Terminal**, mostrada na 30 (b). A parte superior dessa tela contém quatro botões de comandos cujos ícones e funções encontram-se no Quadro 2. A região central da tela corresponde à área de mensagens e está inicialmente vazia. Ela registrará os dados de cada mensagem enviada e recebida pelo app, formando uma lista com o horário da mensagem, comandos e dados enviados/recebidos. Depois de algum tempo de

¹⁷ Disponível em: https://play.google.com/store/apps/details?id=de.kai_morich.serial_bluetooth_terminal. Acesso em: 20 fev. 2019.

uso, a área de mensagens passa a ter um aspecto similar ao encontrado na 30 (c). Finalmente, na parte inferior da tela encontram-se alinhados sete botões de macros e abaixo deles, a caixa de entrada para os dados a serem enviados pelo app.





30 – Ícone e tela inicial do app Serial Bluetooth Terminal.



Fonte: print screen do app no sistema Android.

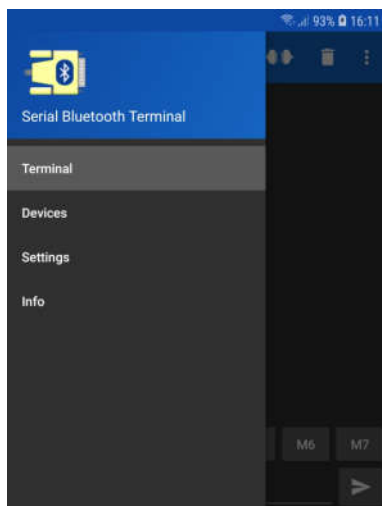
Caso seja a primeira vez de conexão é necessário fazer a configuração de pareamento. Para isso, com a placa HC-06 ativa e em busca de pareamento (com LED vermelho piscando) e com o smartphone já configurado para fazer conexão Bluetooth, deve-se acionar o botão de mudança de telas, localizado no canto superior esquerdo da tela inicial do Terminal e cujo ícone é mostrado no Quadro 2 (a).

Quadro 2 – Botões de comando do Serial Bluetooth Terminal.

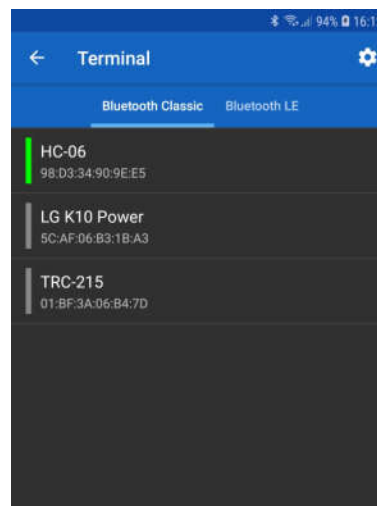
| | Ícone | Ação |
|-----|---|---------------------------|
| (a) |  | Mudança de telas |
| (b) |  | Conectar/Desconectar |
| (c) |  | Limpar lista de mensagens |
| (d) |  | Configurações |

Fonte: print screen do app no sistema Android.

Após acionar o botão de mudança de telas, aparece uma lista com quatro opções de telas disponíveis, como indica a Figura 31 (a).

Figura 31 – Lista de telas e Devices do Serial Bluetooth Terminal.

(a) Lista de Telas



(b) Tela Devices

Fonte: print screen do app no sistema Android.

Depois, acionar a opção **Devices** para ter acesso à lista com todos os dispositivos Bluetooth disponíveis, cujo aspecto é mostrado na Figura 31 (b). Nela existem duas abas: na primeira aba encontram-se os dispositivos disponíveis com a versão Bluetooth Classic; na segunda aba estão aqueles com

a versão Bluetooth LE. A placa HC-06 deve ser selecionada na aba Bluetooth Classic. Em seguida, será solicitada uma senha, que no caso da placa HC-06 é 1234 por padrão. Finalmente, a área de mensagens apresentará um aviso indicando o sucesso ou não do pareamento e o botão Conectar/Desconectar será representado por um dos dois ícones encontrados no Quadro 2 (b) que corresponda ao estado atual da conexão.

Caso a configuração de pareamento entre o smartphone e a placa HC-06 tenha sido configurada corretamente, acionar o botão Conectar/Desconectar tem como ação alternar entre os dois estados de conexão, sendo desnecessário fazer nova configuração em uso futuro.

Com a conexão ativa é possível enviar diversos comandos e receber inúmeras informações. Após algum tempo a tela enche-se de dados que não são mais úteis e assim passa a ser interessante fazer a limpeza da área de mensagens, através do botão com ícone mostrado no Quadro 2 (c).

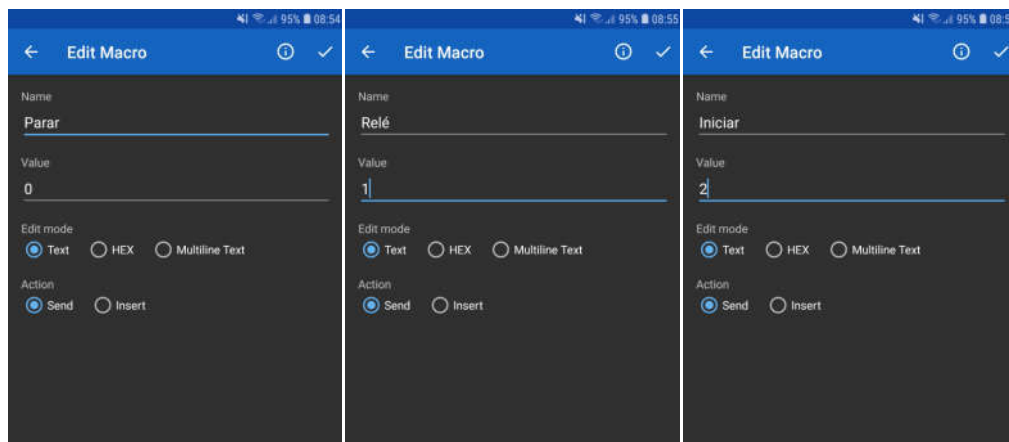
3.2 BOTÕES DE MACROS

É possível automatizar o envio de comandos e dados através do uso dos botões de macros. Para isso, basta pressionar um deles por dois ou três segundos até surgir a **Edit Macro** (tela de edição de macro). Com essa tela aberta, é possível inserir o nome para a macro, o valor desejado, o modo de edição (Texto, Hexadecimal ou Texto de múltiplas linhas) e a ação (Enviar ou inserir). Após preencher esses dados, deve-se acionar o botão de confirmação, encontrado no canto superior direito da tela para salvar as alterações.

Como exemplo, a Figura 32 (a), a Figura 32 (b) e a Figura 32 (c) mostram, respectivamente, as telas de edição de macro dos botões M1, M2 e M3 para automatizar o envio dos valores 0, 1 e 2 e trocar seus nomes para **'Parar'**, **'Relé'** e **'Iniciar'**. É importante observar que nesse caso o três botões devem enviar um

valor e, portanto, no item **Action** de cada um deles deve ser escolhida a opção **Send**.

Figura 32 – Configuração dos botões de macros M1, M2 e M3.



(a) Botão M1

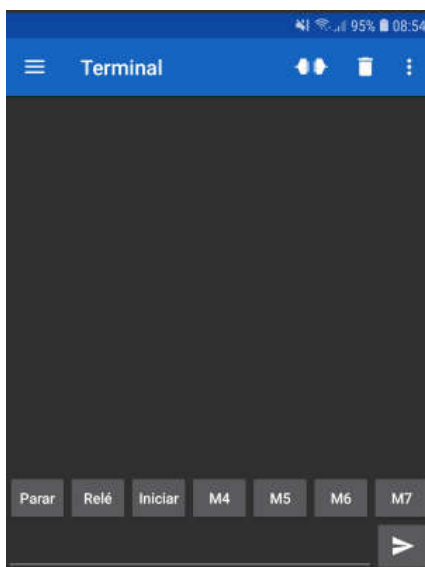
(b) Botão M2

(c) Botão M3

Fonte: print screen do app no sistema Android.

A Figura 33 mostra o aspecto da tela inicial após essa atualização.

Figura 33 – Resultado da configuração dos botões de macros M1, M2 e M3.



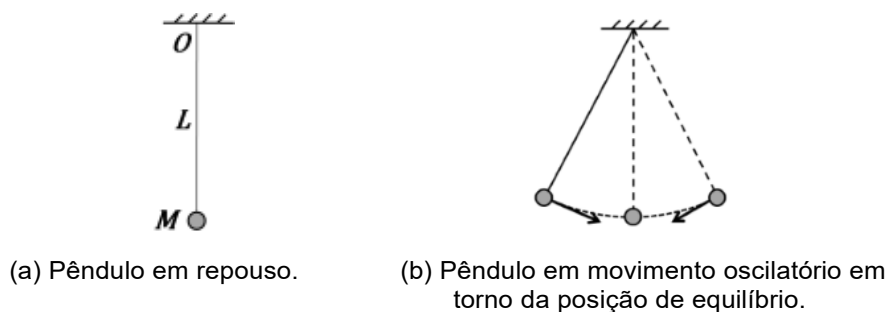
Fonte: print screen do app no sistema Android.

4 DISPOSITIVO DE MEDIDA DO PERÍODO

4.1 MONTAGEM DO PÊNDBULO SIMPLES

De forma geral, como mostra a Figura 34 (a), um pêndulo simples é constituído por um corpo de massa M concentrada e suspensa por um dos extremos de um fio inextensível, com massa desprezível, comprimento L e com o outro extremo fixo em um ponto O . Na condição de equilíbrio estático, quando o corpo está somente sob ação das forças gravitacional e força de tração no fio, o fio tem direção vertical. Quando o corpo é afastado de sua posição de equilíbrio e abandonado, como mostrado na Figura 34 (b), ele oscila em torno desta posição com um período de tempo T .

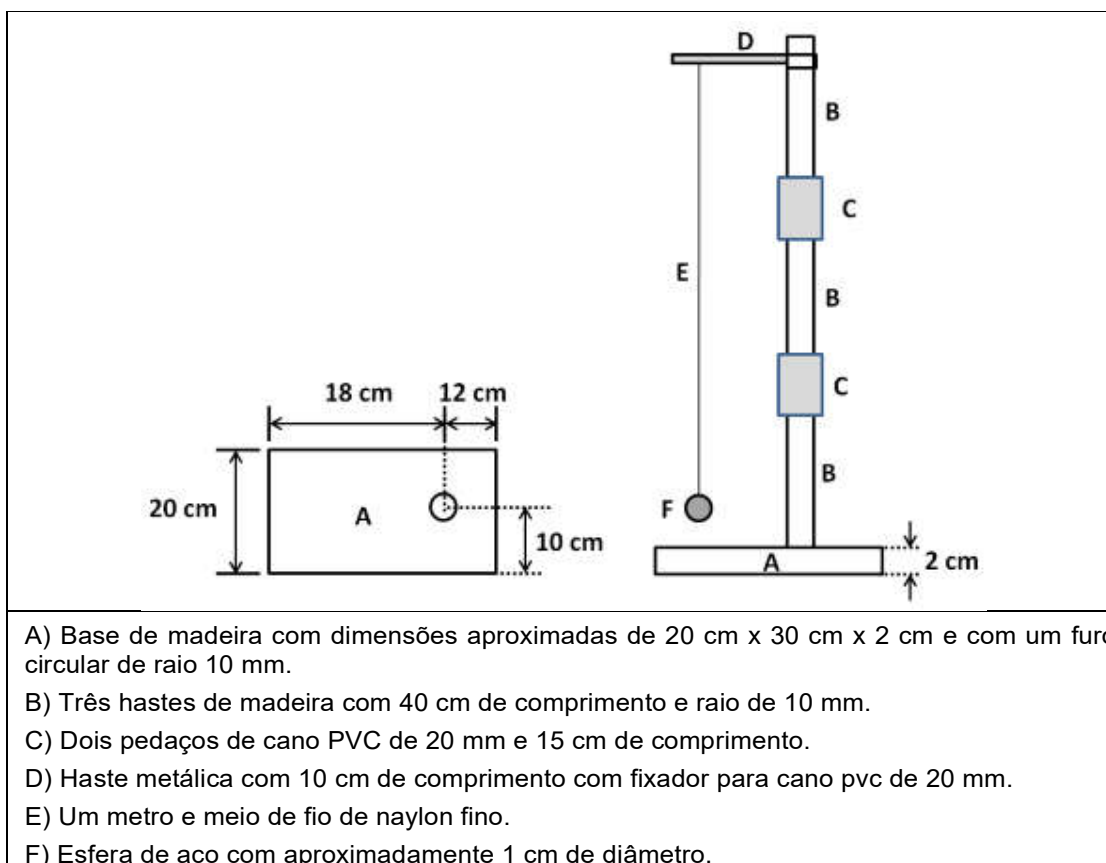
Figura 34 – Repouso e movimento do pêndulo simples.



Fonte: elaborado pelo autor.

A Figura 35 mostra o esquema usado para construção do pêndulo usado nesse trabalho. Nas medidas realizadas, L tinha valores entre 50 cm e 100 cm, por isso, foi utilizado um cabo de madeira com aproximadamente 1,2 m para ser a haste vertical. Com objetivo de facilitar o transporte, o cabo foi seccionado em três partes (item B da Figura 35), cada uma com aproximadamente 40 cm de comprimento e unindo-se facilmente com outra secção através de encaixes feitos com pedaços de tubos de PVC (item C da Figura 35). O conjunto foi então fixado à base de madeira (item A da Figura 35) pelo furo nela existente.

Figura 35 – Esquema de construção do pêndulo simples.



Fonte: elaborado pelo autor.

Como haste metálica (item D da Figura 35), foi usada uma haste galvanizada para telha com diâmetro de $\frac{1}{4}$ de polegada por 150 mm de comprimento, facilmente adquirida em qualquer loja de ferragens ou de material de construção. Ela foi serrada no lado rosqueado para atingir o comprimento de 10 cm. E com objetivo de facilitar a fixação do fio do pêndulo, ela também recebeu um furo de aproximadamente dois milímetros de diâmetro próximo a extremidade oposta àquela que foi cortada. Evidentemente, em seu lugar podem ser usados pedaços de madeira, pregos, parafusos ou arames, mas a haste para telha, além de ser barata e fácil de ser encontrada, tem boa resistência mecânica e tem diâmetro que se ajusta perfeitamente no furo de um fixador de tubos de PVC de 20 mm, esse também facilmente encontrado em qualquer loja de ferragens ou de material de construção.

Com objetivo de facilitar a mudança no comprimento do pêndulo, a haste para telha foi conectada ao fixador de tubos de PVC de 20 mm, como visto na Figura 36, e este encaixado à haste vertical de madeira. O encaixe é suficientemente rígido para suportar o movimento do pêndulo sem que a haste metálica saia de sua posição, mas permite que ela seja suspensa ou baixada quando necessário.

Finalmente, a esfera de aço com 1 cm de diâmetro foi adquirida em uma loja de aviamentos e bijuterias, mas pode também ser facilmente encontrada em sites da internet. Ela foi envolvida em um pedaço de plástico transparente comum com dimensões de 4 cm x 4 cm e amarrada ao fio do pêndulo, sendo o excesso de plástico aparado com uma tesoura.

Figura 36 – Haste metálica conectada ao fixador de tubos de PVC.



Fonte: elaborado pelo autor.

4.2 MEDIÇÃO DO PERÍODO DO PÊNDULO COM ARDUINO E LDR

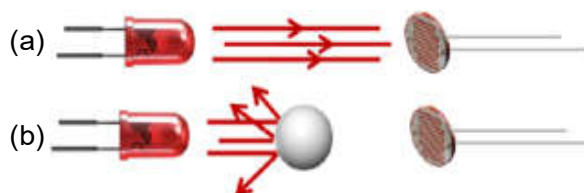
Todo sistema de aquisição de dados possui pelo menos um sensor que faz a coleta de alguma grandeza física (luminosidade, temperatura, força, pressão, som, etc), converte em um sinal elétrico e envia esse sinal a um microcomputador ou computador para a análise dos dados. Atualmente existe uma enorme variedade de sensores e componentes eletrônicos de baixo custo que podem ser empregados com esse objetivo (HAAG; ARAUJO; VEIT, 2005).

No caso particular de medição de tempo pode-se citar, entre outros, o uso de sensores de som, com em Aguiar e Pereira (2012), ultrassom, como em

Monk (2014, p. 79), infravermelho, como em Guaitolini Junior et al. (2016) e do par LED/LDR, como em Souza et al. (2011) ou em Soares e Borges (2010). Cada sensor apresenta vantagens e desvantagens uns em relação aos outros, mas cada um tem forma de aplicação e desempenho muito interessantes.

Para medir o período do pêndulo simples foi utilizado o par LED/LDR, seguindo a proposta de Soares e Borges (2010): Em uma situação inicial, como na Figura 37 (a), a luz do LED incide frontalmente sobre o LDR. Nessa situação a luminosidade que atinge o LDR é máxima; Ao ser colocado um objeto opaco entre os dois, como na Figura 37 (b), a luminosidade sobre o LDR diminui.

Figura 37 – Esquema de incidência da luz do LED sobre um LDR.



Adaptado de: Soares e Borges (2010), Mundo da Elétrica¹⁸ e Sigmatechbd¹⁹.

Como visto no Capítulo 2, ao fazer uma conexão entre o LDR e o Arduino com um resistor em modo *pull-up*, na situação mostrada na Figura 37 (a) será enviado um sinal elétrico baixo para o Arduino. Já na situação mostrada na Figura 37 (b) será enviado ao Arduino um sinal elétrico mais intenso.

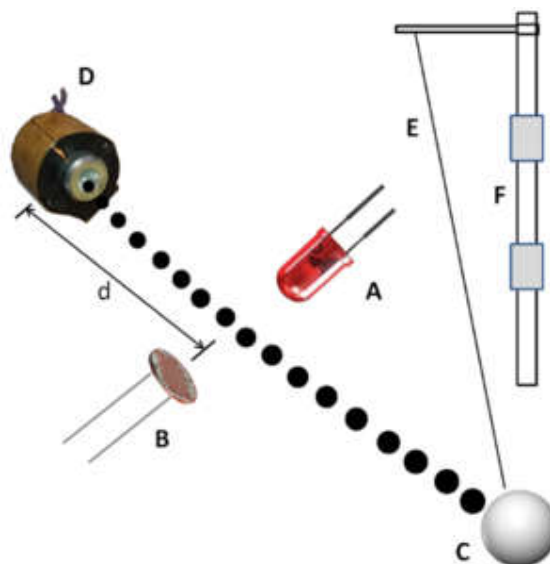
Se o objeto opaco for uma esfera de aço pendurada por um fio de nylon e representada pelo item C da Figura 38, ao ser liberada para mover-se, produzirá um movimento oscilatório tendo como trajetória um arco de circunferência passando entre o LED e o LDR. O período dessa oscilação será o intervalo de tempo necessário para a esfera passar entre o LED e o LDR, atingir um extremo

¹⁸ Disponível em: <https://www.mundodaeletrica.com.br//1050/led-exemplos-1-600.webp>. Acesso em: 30 jul. 2019.

¹⁹ Disponível <https://sigmatechbd.com/wp-content/uploads/2018/02/LDR-Sensor.jpg>. Acesso em: 30 jul. 2019.

da sua trajetória, passar novamente entre o LED e o LDR, atingir o extremo oposto e passar mais uma vez entre o LED e o LDR.

Figura 38 – Componentes do pêndulo e do dispositivo de medida.



- | | |
|------------------|-----------------------|
| A: LED | D: Eletroímã |
| B: LDR | E: Fio de nylon |
| C: Esfera de aço | F: Suporte do pêndulo |

Adaptado de: Mundo da Elétrica²⁰ e Sigmatechbd²¹.

Em resumo, com um resistor conectado em modo *pull-up* ao Arduino, no curto intervalo de tempo em que a esfera passa entre o LED e o LDR ocorre redução da luminosidade sobre o LDR, aumento da sua resistência e um sinal de valor alto é lido pelo Arduino.

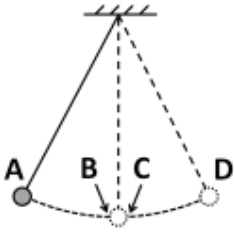
Usando a notação do Quadro 3, o período do pêndulo pode ser medido como o tempo necessário para a esfera fazer o movimento B–C–D–C–B–A–B. O sinal lido pelo Arduino é alto quando a esfera está no setor 1 e baixo quando ela está nos setores 0 e 2.

²⁰ Disponível em: <https://www.mundodaeletrica.com.br/i/1050/led-exemplos-1-600.webp>. Acesso em: 30 jul. 2019.

²¹ Disponível <https://sigmatechbd.com/wp-content/uploads/2018/02/LDR-Sensor.jpg>. Acesso em: 30 jul. 2019.

Quadro 3 – Setores de um ciclo de movimento da esfera.

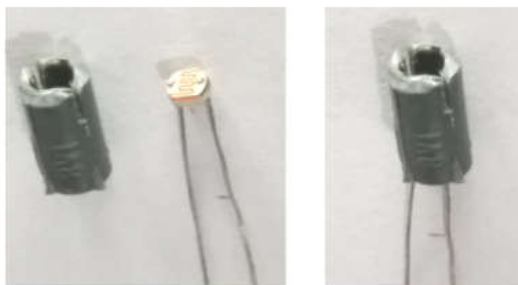
| Posição inicial | Posição Final | signal | setor | sentido |
|-----------------|---------------|--------|-------|---------|
| A | B | baixo | 0 | 0 |
| B | C | alto | 1 | 0 |
| C | D | baixo | 2 | 0 |
| D | C | baixo | 2 | 1 |
| C | B | auto | 1 | 1 |
| B | A | baixo | 0 | 1 |



Fonte: elaborado pelo autor.

Dessa maneira, a esfera entra no setor 1 em duas situações: vinda do ponto A para o ponto B, quando está no setor 0 e tem sentido 0; ou vinda do ponto D para o ponto C, quando está no setor 2 e tem sentido 1. Assim, o movimento B–C–D–C–B–A–B sempre inicia quando o sinal recebido pelo Arduino passa de baixo para alto, instante em que a posição da esfera passa do setor 0 para o setor 1 e tem sentido de movimento obrigatoriamente igual a 0.

É importante observar que a luminosidade ambiente também atua sobre o LDR e a variação de sua intensidade pode gerar medidas que sinalizam ao Arduino uma falsa mudança no setor em que a esfera se localiza e assim produzir medidas completamente distorcidas. Uma maneira eficiente de diminuir bastante esse problema é usar um invólucro de material opaco em torno do LDR. Por exemplo, como mostrado na Figura 39, uma bucha para fixar parafusos em concreto, se cortada pela metade pode ser uma solução simples e muito eficiente para diminuir a ação da luz ambiente sobre o LDR.

Figura 39 – Invólucro opaco do LDR contra efeitos da luz ambiente.

Fonte: elaborado pelo autor.

4.3 CONTROLE DO INÍCIO DO MOVIMENTO DO PÊNDBULO

A esfera do pêndulo tem diâmetro aproximado de um centímetro e como é feita de aço, um material relativamente denso, seu atrito com o ar é desprezível para a medição do período. O aço da esfera tem ainda outra vantagem: permite que seja usado um eletroímã (item D da Figura 6.5) para iniciar o movimento, garantindo que a velocidade inicial da esfera seja sempre nula e também que ela seja solta sempre da mesma posição, o que torna tanto o ângulo inicial do fio do pêndulo com a reta vertical e a direção do movimento sempre os mesmos em todas as medidas. Isso evita que a esfera seja lançada em direção ao LED ou LDR, podendo causar algum tipo de dano a esses dispositivos. Para tentar evitar que isso ocorra, o lançamento com as mãos em geral torna-se mais lento, não sendo incomum a necessidade de repetição do lançamento por acontecer o choque da esfera com um deles.

Outra vantagem do uso do eletroímã é garantir a condição harmônica do pêndulo. Sendo d a metade da distância entre os pontos extremos da trajetória do pêndulo e sendo θ_0 o ângulo inicial do fio da esfera em relação ao eixo vertical, é possível obter os valores de θ_0 mostrados na Tabela 2.

Tabela 2 – Relação entre o raio e o ângulo inicial do pêndulo.

| L (cm) | d (cm) | $\text{sen}(\theta_0) = d/L$ | θ_0 (rad) | θ_0 (grau) |
|--------|--------|------------------------------|------------------|-------------------|
| 50 | 9 | 0,18000 | 0,18099 | 10,36976 |
| 60 | 9 | 0,15000 | 0,15057 | 8,62693 |
| 70 | 9 | 0,12857 | 0,12893 | 7,38705 |
| 80 | 9 | 0,11250 | 0,11274 | 6,45945 |
| 90 | 9 | 0,10000 | 0,10017 | 5,73917 |
| 100 | 9 | 0,09000 | 0,09012 | 5,16361 |

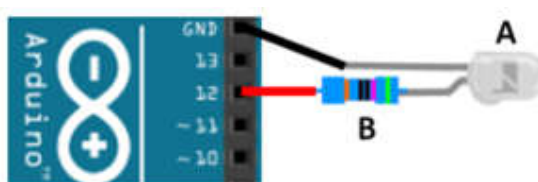
Fonte: elaborado pelo autor.

Assim, a pior situação de precisão da medida ocorre para $L = 50$ cm, ainda dentro do intervalo de aproximadamente 10 graus ou menos para a validade da condição harmônica (SERWAY e JEWETT JR; 2011, p. 3).

4.4 ESQUEMAS ELÉTRICOS DO DISPOSITIVO DE MEDIÇÃO

Existem quatro circuitos elétricos necessários ao funcionamento do projeto. O primeiro deles, mostrado na Figura 40 consiste no circuito do LED.

Figura 40 – Circuito elétrico de ligação do LED ao Arduino.



A: LED de alto brilho B: Resistor de $570 \Omega \pm 1\%$

Adaptado de: programa Fritzing.

Seu ânodo (+ 5 V) está conectado a um resistor e este ao pino 12 do Arduino. Apesar de poder ser usado um resistor de menor valor em série ao LED, a resistência de 570Ω permitiu ao LED ter desempenho satisfatório.

O segundo circuito é o do LDR, mostrado na Figura 41. O resistor é conectado em modo *pull-up*. O LDR tem seus terminais conectados ao pino analógico A0 e ao pino GND. Assim, quando o LED ilumina diretamente o LDR, o Arduino recebe um sinal elétrico baixo. Mas quando a esfera passa entre o LED e o LDR, diminui a luminosidade incidente no LDR e o Arduino passa a receber um sinal elétrico mais intenso.

Figura 41 – Circuito elétrico de ligação do LDR ao Arduino.

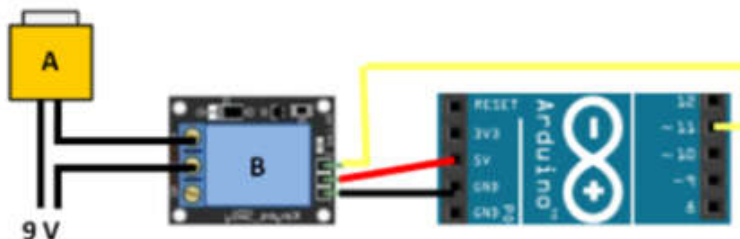


A: LDR **B:** Resistor de $10\text{ k}\Omega \pm 5\%$

Adaptado de: programa Fritzing.

O terceiro circuito é do eletroímã, mostrado na Figura 42 O pino IN do rele está conectado ao pino 11 do Arduino.

Figura 42 – Circuito elétrico de ligação do eletroímã ao Arduino.



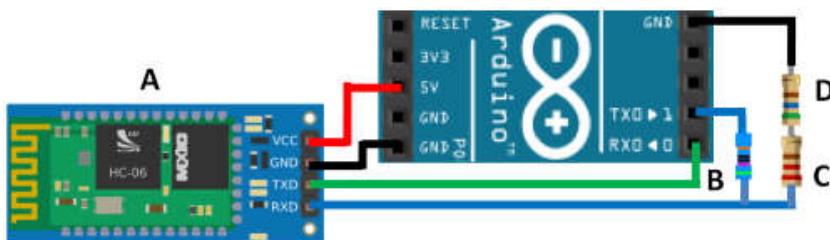
A: Eletroímã **B:** Módulo relé de um canal

Adaptado de: programa Fritzing.

Para evitar alguma interferência no Arduino devido ao eletroímã é aconselhável que eles tenham fontes de alimentação elétrica independentes.

O quarto e último circuito é o de comunicação Bluetooth, mostrado na Figura 43.

Figura 43 – Circuito elétrico de ligação do Bluetooth ao Arduino.



A: HC-06 **B:** $570\ \Omega \pm 1\%$ **C:** $220\ \Omega \pm 5\%$ **D:** $560\ \Omega \pm 5\%$

Adaptado de: programa Fritzing.

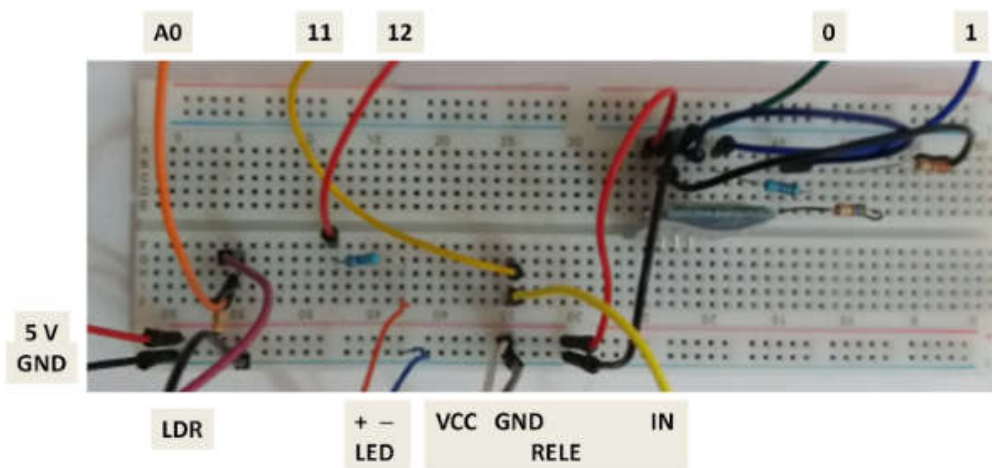
Um resistor de 570Ω (item B da Figura 43) é usado em um dos lados do divisor de tensão. No outro lado do divisor são usados dois resistores ligados em série com resistências de 220Ω (item C da Figura 43) e 560Ω (item D da Figura 43), fazendo o HC-06 (item A da Figura 43) receber a ddp de:

$$\frac{560 + 220}{570 + 560 + 220} \cdot 5 \text{ V} = \frac{780}{1350} \cdot 5 \text{ V} \approx 2,89 \text{ V} \quad (10)$$

Essa tensão foi suficiente para o HC-06 funcionar satisfatoriamente e não ter sido verificado nenhum problema em sua operação. Inicialmente, seriam usados resistores com valores iguais a $1 \text{ k}\Omega$ e $2,2 \text{ k}\Omega$, seguindo a proposta encontrada em Eletrogate (2019).

Na Figura 44 é apresentado, montado numa protoboard, o circuito elétrico do dispositivo de medição do período. No lado esquerdo encontra-se o circuito referente ao LDR. Junto a ele e mais à direita, encontra-se o circuito do LED. Ao centro, está o circuito de controle do relé do solenóide e por fim, do lado direito da protoboard encontra-se o circuito do HC-06.

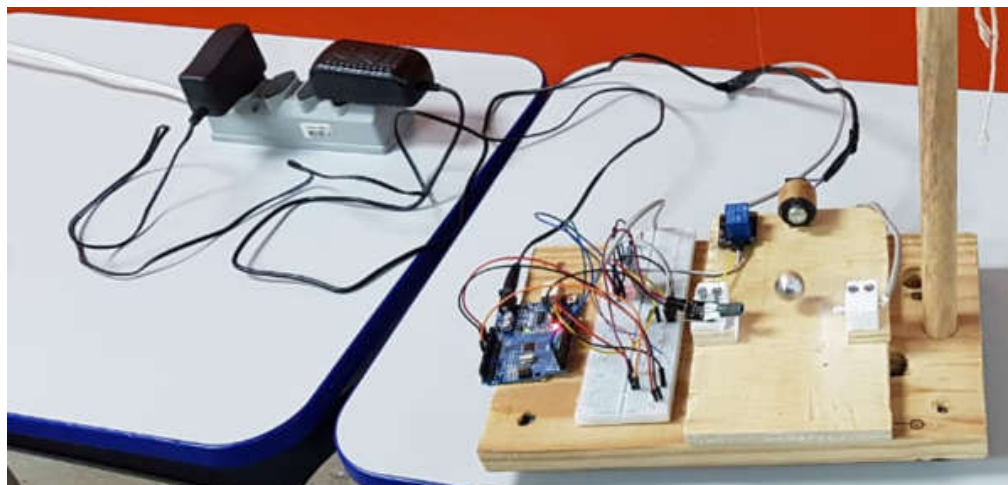
Figura 44 – Circuito de medição do período na protoboard.



Fonte: elaborado pelo autor.

A Figura 45 mostra o dispositivo de medição do período juntamente com a base de sustentação do pêndulo e as fontes de tensão elétrica usadas para alimentar o Arduino (9 V – 1 A) e o eletroímã (9 V – 2 A).

Figura 45 – Circuito de medição e fontes de tensão.



Fonte: elaborado pelo autor.

Finalmente, a Figura 46 mostra o dispositivo em uso durante a realização de um experimento.

Figura 46 – Dispositivo de medida do período do pêndulo simples.



Fonte: elaborado pelo autor.

4.5 DETERMINAÇÃO DA FREQUÊNCIA OPERAÇÃO DO LOOP

O loop do *sketch* deve ter sua frequência de funcionamento limitada pela máxima frequência de operação do LDR que é da ordem de alguns quilohertz (BRAGA, 2019). Esse limite é fundamental para evitar leituras incorretas do LDR.

Uma estratégia para determinar a frequência de operação de leitura do LDR é controlar o circuito elétrico de medição do período com o *sketch* de teste:

```
int tempo0

void setup() {
  tempo0 = millis();
  Serial.begin(9600);
}

void loop() {
  int valor = analogRead(0);
  int tempo1 = millis();
  Serial.println(tempo1-tempo0);
  tempo0 = tempo1;
}
```

Nesse procedimento, inicialmente é criada a variável inteira **tempo0**. A seguir, no **setup**, o tempo em milissegundos decorrido desde o início da execução do *sketch* é obtido pela função **millis()** e armazenado em **tempo0**. Depois, a porta serial é configurada para trabalhar a 9600 bps. No **loop**, primeiro a variável inteira **valor** é usada para receber a leitura do LDR. Em seguida, nova leitura de **millis()** é feita e armazenada na variável **tempo1**. O intervalo de tempo entre dois loops consecutivos é dado pela diferença **tempo1 - tempo0** e o resultado enviado para a porta serial. Finalmente, antes do loop reiniciar, a variável **tempo0** é atualizada e passa a ter o valor de **tempo1**.

Após a realização de inúmeras medições usando esse *sketch* de teste, o valor médio encontrado para a frequência do loop foi da ordem de 10 000 ciclos

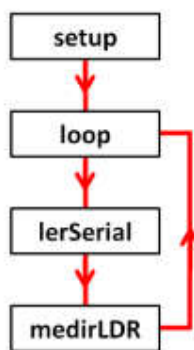
por segundo, muito acima dos poucos quilohertz que permitem uma leitura confiável do LDR.

Como o *sketch* de teste tem uma frequência próxima à do *sketch* realmente usado na medida do período do pêndulo, será necessário usar uma estratégia para reduzir seu valor a um nível que permita uma medida confiável do LDR.

4.6 SKETCH PARA CONTROLE DO DISPOSITIVO DE MEDIÇÃO

Para controlar o sistema de medição foi desenvolvido o *sketch* **PENDULO_SIMPLES** (Apêndice I). Ele é constituído por cinco partes: a lista de variáveis globais e procedimentos, **setup**, **loop**, **lerSerial**, e **medirLDR**. O seu fluxograma é mostrado na Figura 47.

Figura 47 – Fluxograma do sketch de controle



Fonte: elaborado pelo autor.

4.6.1 Variáveis globais do sketch

A lista de variáveis globais é:

```

int pinoRELE = 11; // pino de conexão do Rele
int pinoLED = 12; // pino de conexão do LED
int pinoLDR = 0; // pino de entrada do LDR
  
```

```
int ciclo; // número de ciclos medidos
int setor; // setor atual da posição da esfera
int sentido; // sentido do movimento da esfera
int valorBase; // valor mínimo da medida do LDR
float taxa = 1.1; // taxa indicativa de início de ciclo
int tempo0; // instante de início do ciclo
char buf0 = '0'; // valor anteriormente lido na porta serial
```

Para identificar os pinos de conexão do relé, do LED e do LDR são usadas as variáveis globais **pinoRELE**, **pinoLED** e **pinoLDR**. O número de ciclos medidos é registrado na variável global inteira **ciclo**.

De acordo com a nomenclatura presente no Quadro 3, os valores do setor e do sentido de movimento da esfera serão respectivamente armazenados nas variáveis globais **setor** e **sentido**.

Uma pequena mudança na luminosidade ambiente ou até mesmo uma pequena alteração na distância entre a esfera e o LDR no momento em que ela obstrui a luz vinda do LED modificam a intensidade da luminosidade sobre o LDR e pode gerar medidas que levem a um erro na determinação da posição da esfera. Para diminuir esses inconvenientes foi tomada uma estratégia que faz uso das variáveis **valorBase**, **valor** e **taxa**.

Quando o eletroímã está energizado e a esfera conectada a ele, de acordo com o Quadro 3, ela encontra-se no ponto A do setor 0. Nesse instante, o Arduino recebe um sinal baixo do LDR. Quando o eletroímã é desligado e a esfera é liberada para mover-se, o Arduino registra o valor do sinal enviado pelo LDR na variável global inteira **valorBase**. Esse é o menor valor lido pelo LDR durante toda a medição, desde que a luminosidade ambiente não sofra aumento.

Durante o movimento da esfera, milhares de loops são realizados pelo *sketch* e em cada um deles o procedimento **lerLDR** recebe a leitura do sinal do LDR. O valor desse sinal é registrado na variável local inteira **valor** e em

seguida comparado com **valorBase** para determinar, de acordo com o Quadro 3, em que setor se encontra a esfera.

A comparação **valor > taxa * valorBase**, em que **taxa** é uma variável global de ponto flutuante, determina quanto maior deve ser **valor** em relação a **valorBase** para ser considerado um sinal que corresponda à passagem da esfera entre o LED e o LDR. Testes com diversos níveis de luz ambiente e com pequenas alterações na distância entre a esfera e o LDR no ponto em que ela fica entre ele e o LED, indicaram que **taxa** igual a 1.1 era suficiente para eliminar quase por completo medidas incorretas devido a esses efeitos indesejados.

A variável **tempo0** será usada no procedimento **lerLDR** para armazenar o instante de tempo que um ciclo tem início.

Por último, é definida a variável global do tipo **char** e de nome **buf0**, cujo valor inicial é '0'. Ela será usada no procedimento **lerSerial** para armazenar a ação enviada pelo smartphone.

4.6.2 Procedimento setup

No setup encontram-se os seguintes comandos:

```
void setup() {  
  pinMode(pinoRELE, OUTPUT);  
  pinMode(pinoLED, OUTPUT);  
  digitalWrite(pinoRELE, LOW);  
  digitalWrite(pinoLED, LOW);  
  Serial.begin(9600);  
  Serial.println((String)"Sistema desligado !!!");  
}
```

As duas primeiras linhas informam ao Arduino que os pinos digitais 11 e 12 são do tipo OUTPUT. As terceira e quarta linhas informam que esses pinos estão no estado LOW, ou seja, desligados.

A penúltima linha inicia a comunicação serial com uma taxa de 9600 bps e a última linha escreve na porta serial a frase “Sistema desligado !!!” e a envia ao smartphone via Bluetooth.

4.6.3 Procedimento loop

O loop apresenta apenas três comandos:

```
void loop() {  
  lerSerial();  
  medirLDR();  
  delay(2);  
}
```

Os procedimentos **lerSerial** e **MedirLDR** serão executados repetidamente até que o Arduino seja desligado ou reiniciado. Testes realizados sem uso da função **delay** mostraram para o loop uma frequência de funcionamento da ordem de 10 kHz. Como visto anteriormente, isso causa um problema, pois o LDR opera com frequência de operação da ordem de alguns kHz. A função **delay(2)** causa um atraso de 2 milésimos de segundo entre o fim de um loop e o início do próximo. Como o experimento tem a medida do período realizada apenas com centésimos de segundo, esse pequeno atraso é irrelevante para a sua precisão, porém é fundamental para serem obtidas leituras corretas do LDR que não causem erros de lógica na execução do *sketch*.

4.6.4 Procedimento lerSerial

O aparelho smartphone se comunica com o Arduino enviando códigos de ação de acordo com o Quadro 4. O procedimento **lerSerial** é responsável por receber esse código e comparar com o valor de **buf0**, que armazena o valor recebido na leitura anterior. Quando o valor recebido for diferente de **buf0** e for 0, 1 ou 2, a ação correspondente a ele é executada. Essa comparação é

importante para evitar que sucessivos e desnecessários envios do mesmo código por parte do usuário sejam indevidamente executados pelo Arduino.

Quadro 4 – Códigos de ação enviados pelo smartphone.

| Valor lido | Ação |
|------------|--|
| 0 | Desligar o eletroímã, desligar o LED e parar a medição |
| 1 | Ligar só o eletroímã. |
| 2 | Desligar o eletroímã, ligar o LED e iniciar medição |

Fonte: elaborado pelo autor.

O procedimento **lerSerial** tem a seguinte estrutura:

```

void lerSerial(){
  while(Serial.available() > 0){
    char buf = Serial.read();
    if (buf != buf0){
      buf0 = buf;
      if (buf == '0'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, LOW);
        Serial.println((String)"Sistema desligado !!!");
      }
      else if (buf == '1'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, HIGH);
        Serial.println((String)"Relé ligado !!!");
      }
      else if (buf == '2'){
        ciclo = 0;
        setor = 0;
        sentido = 0;
        digitalWrite(pinoLED, HIGH);
        delay(1000);
        digitalWrite(pinoRELE, LOW);
        Serial.println((String)"Período (segundos:)");
        valorBase = map(analogRead(pinoLDR),0,1023,0,100);
      }
    }
  }
}

```


A instrução `while` se repete enquanto houver algum byte disponível para leitura da porta serial. Caso exista, seu valor é posto na variável local `buf`. Em seguida o teste `buf != buf0` (`buf` é diferente de `buf0`?) é realizado. Em caso verdadeiro, o procedimento modifica o valor de `buf0` para torna-se igual ao de `buf` e em seguida realiza a comparação do valor de `buf` com os códigos das ações contidas no Quadro 4.

Com `buf` igual 0, de acordo com o Quadro 4, uma ação de parar o experimento é executada. Assim, os pinos 12 do LED e 11 do relé são desligados e é enviada para a porta serial uma mensagem de encerramento.

Caso `buf` seja igual a 1, novamente de acordo com o Quadro 4, o relé do eletroímã é energizado e o LED desenergizado. Também é enviada para a porta serial a mensagem de acionamento do relé.

Finalmente, se `buf` for igual a 2, de acordo com o Quadro 4, a medição do experimento deve iniciar. Assim, as variáveis globais `ciclo`, `setor` e `sentido` são iniciadas com valores iguais a 0. Em seguida o LED é energizado e após um segundo o relé é desligado. O uso de `delay(1000)` é necessário para garantir que a esfera atinja a posição entre o LED e o LDR com este já iluminado pelo LED e apto a fazer leituras corretas. Outros valores menores poderiam ser usados, mas por segurança o intervalo de um segundo foi usado. Em seguida uma mensagem indicando que a medição se inicia é enviada para a porta serial. Por fim, o valor lido pelo pino do LDR é lido e remapeado para um valor entre 0 e 100 e esse valor é armazenado na variável `valorBase`.

A função `map` remapeia um número de um intervalo para outro, truncando o resultado para um número inteiro se necessário. Dessa forma, a expressão `map(x, 0, 1023, 0, 100)` irá converter o valor de `x` para:

$$x \cdot \frac{100 - 0}{1023 - 0} \quad (11)$$

Assim, por exemplo, os valores de 502 até 511 serão convertidos para o valor 49. Essa conversão será útil para diminuir a sensibilidade do sinal enviado pelo LDR e aumentar a segurança de acerto na comparação entre **valor** e **valorBase**, realizada no procedimento **medirLDR**.

4.6.5 Procedimento medirLDR

De acordo com o Quadro 3, a leitura do Arduino será de nível elétrico alto somente no setor 1 e baixo nos setores 0 e 2. Assim, para terminar um ciclo e iniciar outro, a esfera deve sair do setor 0 e entrar no setor 1 com movimento no sentido 0. Somente com essas duas condições atendidas é que deve ser iniciada a medição do período de um novo ciclo. Seguindo essa estratégia, o procedimento **medirLDR** tem como código:

```
void medirLDR(){
  int valor = map(analogRead(pinoLDR),0,1023,0,100);
  if (valor > taxa * valorBase){ // Teste I
    if (setor == 0) { // Teste II
      setor = 1;
      sentido = 0;
      int tempo = millis();
      if (ciclo > 0){ // Teste III
        float periodo = (tempo-tempo0) / 1000.;
        Serial.println((String)periodo);
      }
      tempo0 = tempo;
      ciclo = ciclo + 1;
    }
    else if (setor == 2){ // Teste IV
      setor = 1;
      sentido = 1;
    }
  }
  else{
    if (setor == 1){ // Teste V
      if (sentido == 0){ Teste VI
        setor = 2;
      }
    }
    else{
      setor = 0;
    }
  }
}
```

```

    }
  }
}

```

Primeiro, o valor lido pelo LDR e enviado ao Arduino pelo pino analógico A0 é remapeado entre 0 e 100 e o resultado armazenado na variável **valorBase**. Em seguida, no **Teste I** é feita a comparação **valor > taxa * valorBase**. Caso esse teste seja verdadeiro, a luz do LED que incide no LDR está sendo encoberta parcialmente ou totalmente pela esfera. Assim, pelo Quadro 3, a esfera se localiza no setor 1. Caso o **Teste I** seja falso, a esfera está no setor 0 ou no setor 2.

Com o **Teste I** sendo verdadeiro é necessário saber de onde a esfera veio. Por isso é realizado o **Teste II**. Nele é feita a comparação **setor == 0** (setor é igual a 0?). Caso ele seja verdadeiro, a variável **setor** é atualizada para 1 e a variável **sentido** para 0 (Quadro 3). Em seguida é executada a instrução **int tempo = millis()**. Ela cria a variável local inteira de nome **tempo** e nela armazena o intervalo total em milissegundos que o *sketch* está em execução. Em outras palavras, como a esfera acabou de entrar no setor 0, **tempo** armazena o instante final do ciclo anterior e que também é o instante inicial do novo ciclo. O **Teste III** verifica se foi completado pelo menos o primeiro ciclo. Se verdadeiro, será executada a instrução **float periodo = (tempo-tempo0) / 1000**. O valor da variável **periodo** é então enviado para a porta serial e depois para o smartphone. Finalizando o bloco de instruções da condição verdadeira do **Teste II**, a variável **tempo0** é atualizada e recebe o valor de **tempo** e a variável **ciclo** é incrementada de 1.

Caso o **Teste II** seja falso, é realizado o **Teste IV**. Nele é feita a comparação **setor == 2** (setor é igual a 2?). Se ele for verdadeiro, significa que a esfera entrou no setor 1 vinda pelo setor 2 e com sentido 1, assim a variável **setor** é atualizada para 1 e a variável **sentido** para 1.

O **Teste I** sendo falso significa que o LED recebe luz direta do LDR e a esfera está no setor 0 ou no setor 2. Assim, é necessário apenas atualizar a

variável **setor** se ela ainda estiver com valor 1. Por isso é realizado o **Teste V**, em que é feita a comparação **setor == 1** (setor é igual a 1?). Somente caso esse teste seja verdadeiro é feito o **Teste VI**, em que a comparação **sentido == 0** (sentido é igual a 0?) é realizada. Se ele for verdadeiro, a variável **setor** é atualizada para 2, caso contrário ela é atualizada para 0.

5 PROPOSTA DIDÁTICA DE APLICAÇÃO

A proposta aqui apresentada é baseada em uma abordagem construtivista encontrada no trabalho de Rosa (2012). Nela, a aplicação do experimento foi dividida em três etapas, cada uma em uma aula com 50 minutos de duração. E como material de apoio, sendo usada a Ficha de Experimento do Pêndulo Simples (Apêndice II). É importante observar que essa ficha foi desenvolvida originalmente para ser aplicada em momento posterior a apresentação da teoria sobre o pêndulo simples e por isso, modificações podem ser necessárias caso seja desejado fazer sua aplicação junto com a teoria.

A Ficha de Experimento do Pêndulo Simples faz o uso de logaritmos. Como essa ferramenta matemática é apontada de forma unânime pelos estudantes do ensino médio como difícil de entender, seu uso foi apontado como a maior dificuldade encontrada para a realização do experimento. Assim, pode ser interessante e até mesmo necessária a realização de uma revisão em momento anterior a aplicação do experimento.

A primeira etapa de aplicação do experimento, chamada Pré-experimental, deve contar com: Rápida revisão teórica, com base no livro texto, sobre o pêndulo simples; Debate sobre por que é indispensável se estabelecer um roteiro bem definido para realização de um experimento físico e importância do entendimento de cada etapa; Apresentação aos educandos da Ficha de Experimento do Pêndulo Simples (Apêndice II), seguida por uma discussão detalhada da Etapa 1 dessa ficha, em particular a apresentação e análise da hipótese orientadora do experimento; Escolha de quatro educandos para serem monitores; Divisão do restante dos educandos em grupos de quatro pessoas; Apresentação aos educandos do dispositivo de medição do período e do app Serial Bluetooth Terminal.

Ao final desse dessa etapa é interessante combinar com os monitores um momento extra para que eles possam treinar o experimento e assim na próxima etapa poder melhor orientar os colegas.

Na segunda etapa, chamada Experimental, orientados preferencialmente pelos monitores, os vários grupos de educandos realizam efetivamente as medidas do período do pêndulo para os comprimentos predeterminados e posterior preenchimento da Tabela de Registro de Dados encontrada na Ficha de Experimento.

Por último, a terceira etapa, chamada Pós-Experimental, é basicamente para a realização da análise dos dados com: Preenchimento da Tabela de Conversão Logarítmica a partir da Tabela de Registro de Dados; Produção do gráfico da dependência $\log L \times \log T$ a partir do método de linearização simples com uso de lápis e régua; Obtenção da relação do período; Comparação Teoria x Prática; Conclusão final com apresentação de todos os grupos dos seus resultados e posterior debate sobre diversos tópicos, entre os quais, a comparação entre a previsão teórica e o resultado experimental para validar a hipótese inicial, o levantamento de todas as dificuldades na realização do experimento e discussão sobre dúvidas em geral.

6 CONSIDERAÇÕES FINAIS

Uma diversidade crescente de microcontroladores, sensores, atuadores e dispositivos de comunicação com custo cada vez menor e maior facilidade de aquisição no mercado especializado simplificam o desenvolvimento de experimentos de aquisição de dados e controle de dispositivos com Arduino.

Fazer uso dessa tecnologia integrada a um smartphone é uma excelente forma de dinamizar as aulas, atrair a atenção dos educandos e tornar as aulas de Física mais produtivas, pois ela permite acesso rápido a uma grande quantidade de dados que podem facilmente ser processados, convertidos em gráficos e analisados. Nesse sentido, como primeiro objetivo do presente trabalho, os capítulos dois e três apresentaram um guia de estudo para desenvolver esse tipo de experimento. É importante observar que foi apresentado apenas o início de um estudo, um pequeno recorte da totalidade, mas que é a base para desenvolvimento de qualquer projeto que use essa tecnologia.

Outro ponto a se destacar é que pode ser interessante ou até mesmo necessário o desenvolvimento de um aplicativo customizado para substituir o Serial Bluetooth Terminal. Dependendo do tempo disponível e do interesse de aprendizagem, existem diversas possibilidades. Uma delas, o MIT App Inventor, usa uma interface gráfica de desenvolvimento, sendo mais adequado para aqueles com pouca ou nenhuma experiência em programação. Outra possibilidade, o Android Studio, exige um conhecimento maior em programação e é indicado para um desenvolvimento mais profissional.

Como segundo objetivo do presente trabalho, numa aplicação direta do guia de estudo visto nos capítulos dois e três, foi apresentado no capítulo quatro o desenvolvimento de um aparato de medição experimental para obter a relação

matemática entre o comprimento do pêndulo simples e seu período de movimento.

No capítulo cinco foi apresentada uma proposta de aplicação de experimento com uso da ficha encontrada no Apêndice II, em que é feito o uso de logaritmos. Como essa ferramenta matemática é apontada de forma unânime pelos estudantes do ensino médio como difícil de entender, seu uso foi apontado como a maior dificuldade encontrada para a realização do experimento. Assim, pode ser interessante evitar o seu uso e simplificar o experimento. Para isso, pode-se substituir a hipótese $T = c \cdot L^b$ por $L = k \cdot T^2$ e então usar as transformações $x = T^2$ e $y = L$ obtendo-se $y = kx$. Assim a perda com a diminuição de generalização da hipótese seria compensada com a simplificação da análise do experimento, tornando a atividade mais acessível aos educandos uma vez que as funções polinomiais do primeiro e segundo grau são de uso muito mais comum entre eles.

De forma similar, apenas alterando os objetivos e a metodologia do experimento e sem fazer nenhuma modificação no aparato, é possível, por exemplo, calcular o valor da gravidade local ou analisar a dependência do período do pêndulo com outros parâmetros, tais como a massa da esfera, o material que a constitui, seu raio, seu volume, etc.

Já com pequenas modificações no aparato é possível, por exemplo, estudar o movimento vertical de queda livre e para isso basta retirar a linha de nylon, colocar o eletroímã num ponto acima do LED e do LDR de modo que quando a esfera for liberada passe entre eles. Evidentemente é preciso fazer um ajuste na programação do Arduino. Também é possível estudar o movimento de um corpo no plano inclinado. Para tanto, basta fixar um prego em um pequeno bloco de madeira de modo que ele possa ser atraído pelo eletroímã e permaneça em repouso no topo de uma tábua plana e inclinada. Ao liberar o bloco para mover-se, ele deve passar entre o LED e o LDR.

A utilização de sensores como LDR, infravermelho IR, pressão, temperatura, som, ultra-som, temperatura, vazão, corrente elétrica, campo magnético, velocidade e aceleração permitem fazer aquisição de dados em todas as áreas de estudo da física do ensino médio. E se combinados com atuadores como motores, aquecedores elétricos, emissores infravermelho, entre outros e conectados por USB, ethernet, Bluetooth, wifi ou GSM permitem o desenvolvimento de uma gigantesca quantidade de projetos com alto potencial pedagógico.

Essa integração entre tecnologia, teoria e experimento é capaz de contribuir substancialmente na aprendizagem dos educandos, preparando-os para lidar com as mudanças e avanços das tecnologias e que geram continuamente novas exigências para a sociedade moderna. Dentre esses avanços tecnológicos está o smartphone, que adquiriu um papel muito importante na atualidade e por isso, não faz mais sentido que fique fora do processo de ensino e aprendizagem.

É necessário ressaltar que as tecnologias são apenas recursos e seu uso adequado pode facilitar e estimular em muito a aprendizagem dos educandos, mas é muito importante lembrar que a ação do professor é fundamental nesse processo, sendo ele quem deve definir quais tecnologias usar e quando e como usá-las.

Por fim, esse trabalho mostra que a partir do domínio de certos conteúdos por parte do professor, o smartphone e a placa robótica Arduino se usados com uma metodologia que crie um ambiente ativo e colaborativo de aprendizagem podem se tornar uma ferramenta muito poderosa para o desenvolvimento de experimentos físicos com grande potencial de aprendizagem aos educandos.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIAR, C. E.; PEREIRA, M. M. O computador como cronômetro, Revista Brasileira de Ensino de Física, v. 34, n. 3, art. 3303, 2012. Disponível em: <https://www.scielo.br/pdf/rbef/v34n3/a03v34n3.pdf>. Acesso 20 jul. 2019.

ARDUINO BLINK. Blink. Disponível em: <https://www.arduino.cc/en/tutorial/blink>. Acesso em: 01 abr. 2019.

ARDUINO BUTTON. ARDUINO TUTORIAL. Button. 2015. Disponível em: <https://www.arduino.cc/en/tutorial/button>. Acesso em: 20 jul. 2019.

ARDUINO CLONES. Send in the clones. Disponível em: <https://blog.arduino.cc/2013/07/10/send-in-the-clones/>. Acesso em: 05 mar. 2019.

ARDUINO CONSTANTES. Arduino Constantes. Disponível em: <https://www.arduino.cc/reference/en/language/variables/constants/constants/>. Acesso em: 06 abr. 2019.

ARDUINO DIGITAL READ. digitalRead(). Disponível em: <https://www.arduino.cc/reference/en/language/functions/digital-io/digitalread/>. Acesso em: 10 jun. 2019.

ARDUINO ELSE. Else. Disponível em: <https://www.arduino.cc/reference/en/language/structure/control-structure/else/>. Acesso em: 16 jun. 2019.

ARDUINO HOME. Arduino Home. Disponível em: <https://www.arduino.cc>. Acesso em: 03 fev. 2019.

ARDUINO IF. IF. Disponível em: <https://www.arduino.cc/reference/en/language/structure/control-structure/if/>. Acesso em: 16 jun. 2019.

ARDUINO INTRODUÇÃO. Arduino Introduction. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 10 fev. 2019.

ARDUINO LDR & LED. LDR & LED Light. Disponível em: https://create.arduino.cc/projecthub/Kenpoca_Dias/ldr-led-light-1147c3. Acesso em: 20 jul. 2019.

ARDUINO LINGUAGEM. Arduino Language Reference. Disponível em: <https://www.arduino.cc/reference/en/>. Acesso em: 06 abr. 2019.

ARDUINO PINOS ANALÓGICOS. Analog Input Pins. Disponível em: <https://www.arduino.cc/en/Tutorial/AnalogInputPins>. Acesso em: 09 ago. 2019.

- ARDUINO PINOS DIGITAIS. Digital Pins. Disponível em:
<https://www.arduino.cc/en/Tutorial/DigitalPins>. Acesso em: 10 ago. 2019.
- ARDUINO PRODUTOS. Arduino Products. Disponível em:
<https://www.arduino.cc/en/Main/Products>. Acesso em: 04 abr. 2019.
- ARDUINO PUSHBUTTON. Pushbutton. Disponível em:
<https://www.arduino.cc/en/Tutorial/Pushbutton/>. Acesso em: 20 ago. 2019.
- ARDUINO READ SERIAL. Digital Read Serial. Disponível em:
<https://www.arduino.cc/en/Tutorial/DigitalReadSerial>. Acesso em: 25 ago. 2019.
- ARDUINO SERIAL. Serial. Disponível em:
<https://www.arduino.cc/reference/en/language/functions/communication/serial/>.
Acesso em: 20 ago. 2019.
- ARDUINO SOFTWARE. Arduino Software. Disponível em:
<https://www.arduino.cc/en/main/software>. Acesso em: 20 jan. 2019.
- ARDUINO START. Getting Started with Arduino and Genuino products.
Disponível em: <https://www.arduino.cc/en/Guide/HomePage>. Acesso em: 04 abr. 2019.
- ARDUINO UNO. Disponível em: <https://store.arduino.cc/usa/arduino-uno-rev3>.
Acesso em: 04 abr. 2019.
- ATMEGA. ATmega808/1608/3208/4808 – 32-Pin. Disponível em:
<http://ww1.microchip.com/downloads/en/DeviceDoc/ATmega808-1608-3208-4808-32-Pin-40002017C.pdf>. Acesso em: 13 ago. 2019.
- AUTOCOREROBOTICA. Funcionamento dos resistores de pull-up e pull-down.
Disponível em: <https://autocorerobotica.blog.br/resistores-pull-up-e-pull-down-como-funcionam/>. Acesso em: 04 abr. 2019
- BANZI, Massimo. Getting Started with Arduino. Second Edition. U.S.A: O'Reilly Media, 2011.
- BRAGA, C. Newton. Eletrônica básica para mecatrônica. São Paulo: Saber, 2005.
- BRAGA, C. Newton. LDR (ALM332) Disponível em:
<https://www.newtoncbraga.com.br/index.php/almanaque-tecnologico/201-l/7547-lidr-alm332>. Acesso em: 20 mar. 2019.
- ENGINEERS. Stepper Motor Speed and Direction Control Using Arduino and Bluetooth HC-06 Module through an Android App. 2019. Disponível em:
<https://www.engineersgarage.com/arduino/stepper-motor-controlled-with-hc06-bluetooth-module/>. Acesso em: 10 fev. 2020.

ELECTRICARENA. Arduino Onboard embedded LED's Their Function and Color. Disponível em <http://electricarena.blogspot.com/2015/01/arduino-onboard-embedded-leds-function.html>. Acesso em: 08 maio 2019.

ELETROGATE. Módulos Bluetooth HC05 e HC06 para comunicação com dispositivos móveis com Arduino. Disponível em: <https://blog.eletrogate.com/modulos-bluetooth-hc05-e-hc06-para-comunicacao-com-dispositivos-moveis-com-arduino/>. Acesso em: 17 ago. 2019.

EMBARCADOS LÂMPADA. Acionamento de uma lâmpada com Arduino. Disponível em: <https://www.embarcados.com.br/acionamento-de-uma-lampada-com-arduino/>. Acesso em: 15 maio 2019.

EMBARCADOS SERIAL. Arduino - Comunicação Serial. Disponível em: <https://www.embarcados.com.br/arduino-comunicacao-serial/>. Acesso em: 10 set. 2019.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. Arduino em ação. São Paulo: Novatec, 2013. Tradução de Camila Paduan.

GUAITOLINI JUNIOR, Judismar Tadeu; RAMOS, Gabryel Silva; SILVA, Samir Lacerda da; GAMA, Aline Costalonga. Avaliação do módulo da aceleração da gravidade com Arduino. Caderno Brasileiro de Ensino de Física, v. 33, n. 2, p. 619-640, ago. 2016. Disponível em: <https://periodicos.ufsc.br/index.php/fisica/article/download/2175-7941.2016v33n2p619/32333>. Acesso em: 05 abr. 2020.

GUEDDES, Mark. Manual de projetos Arduino. São Paulo: Novatec, 2017.

FILIFELOP. Como usar o Arduino Bluetooth HC-05 em modo mestre. 2019. Disponível em: <https://www.filipeflop.com/blog/tutorial-arduino-bluetooth-hc-05-mestre/>. Acesso em: 10 fev. 2019.

HAAG, Rafael; ARAUJO, Ives Solano; VEIT, Eliane Angela. Por Que e Como Introduzir a Aquisição Automática de Dados no Laboratório Didático de Física? Física na Escola, v. 6, n. 1, p.69-74, maio 2005. Disponível em: <https://lume.ufrgs.br/handle/10183/116432>. Acesso em: 23 mar. 2020.

KAMEI, Camila A. N.; Barbosa, Lucas F. L. Introdução a Programação Programando com Intel Galileo. Experimento 1. Recife: UFPE - Centro de Informática-Grupo de Engenharia da Computação. Disponível em: https://www.cin.ufpe.br/~lflb/Tutorial%201%20-20conceitos_basicos_galileo.pdf. Acesso em: 14 set. 2019.

LED. PL16 Series 5mm (T1¾) Ultra Bright White LED. Disponível em: <https://pdf1.alldatasheet.com/datasheet-pdf/view/292012/P-TEC/PL16C-WCW13Z.html>. Acesso em: 20 fev. 2019.

MCROBERTS, Michael. Arduino Básico. São Paulo: Novatec, 2011.

MONK, Simon. Projetos com Arduino e Android: use seu smartphone ou tablet para controlar o arduino. Porto Alegre: Bookman, 2014.

MORICH, Kai. Disponível em:
<https://github.com/kaimorich/SimpleBluetoothTerminal>. Acesso em: 20 fev. 2019.

NETTO, Luiz Ferraz. Circuitos Eletrônicos. Disponível em:
<http://netsaber.com.br/apostilas/apostilas/38.doc>. Acesso em: 14 ago. 2019.

OLIVEIRA, Cláudio Luís Vieira. Aprenda Arduino - Uma abordagem prática. – Duque de Caixas: Katzen Editora, 2018.

PAMBOUKIAN, Sergio Vicente Denser; PEDROTTI, Jairo José. Desenvolvimento de Instrumentação Analítica e Software para Controle, Aquisição e Tratamento de Dados Analíticos. Revista Mackenzie de Engenharia e Computação, v. 12, n. 1, p. 32-43, 2012. Disponível em:
<http://editorarevistas.mackenzie.br/index.php/rmec/article/view/3783/3802>. Acesso em: 29 maio 2020.

ROGGIA, Leandro. Automação industrial / Leandro Roggia, Rodrigo Cardozo Fuentes. – Santa Maria: Universidade Federal de Santa Maria, Colégio Técnico Industrial de Santa Maria, Rede e-Tec Brasil, 2016.

ROSA, Cleci T. Werner da; ROSA, Álvaro Becker da. Aulas experimentais na perspectiva construtivista. Física na Escola, v. 13, n. 1, 2012. Disponível em:
<http://www1.fisica.org.br/fne/phocadownload/Vol13-Num1/a021.pdf>. Acesso em: 23 mar. 2020.

SERWAY, Raymond A.; JEWETT JR, John W. Física para cientistas e engenheiros, volume 2: oscilações, ondas e termodinâmica. – São Paulo: Cengage Learning, 2011.

SOARES, Reginaldo R.; BORGES, Paulo de F. O plano inclinado de Galileu: uma medida manual e uma medida com aquisição automática de dados. Revista Brasileira de Ensino de Física, São Paulo, v. 32, n. 2, p.2501.1-2501.11, abr. 2010. disponível em: <https://www.scielo.br/pdf/rbef/v32n2/v32n2a12.pdf>
Acesso em: 25 mar. 2020.

SOUZA, Anderson R. de; PAIXÃO, Alexsander C.; Uzêda, Diego D.; M.A. DIAS, Marco A; DUARTE, Sérgio e Amorim Helio S. de. A placa Arduino: uma opção de baixo custo para experiências de física assistidas pelo PC. Revista Brasileira de Ensino de Física 33, 1702 (2011). Disponível em:
<https://www.scielo.br/pdf/rbef/v33n1/26.pdf>. Acesso em: 23 mar. 2020

TELECO. Bluetooth: O que é? Disponível em:
https://www.teleco.com.br/tutoriais/tutorialblue/pagina_1.asp. Acesso em: 1 set. 2019.

TUTORIALSPPOINT. Arduino – Descrição da placa. Disponível em: https://www.tutorialspoint.com/arduino/arduino_board_description.htm. Acesso em: 04 abr. 2019.

UNESP. Eletroímã. Disponível em <<http://www2.fc.unesp.br/experimentosdefisica/ele11.htm>>. Acesso em: 14 set. 2019.

USP. Aplicações do 1º fenômeno eletromagnético. Disponível em: http://efisica.if.usp.br/eletricidade/basico/campo_corrente/aplic_prim_fenom_eletromag/. Acesso em: 14 set. 2019.

VIDADESILÍCIO LDR. Sensor de Luz – Aprendendo a usar o LDR com Arduino. Disponível em: <https://portal.vidadesilicio.com.br/sensor-de-luz-com-ldr/>. Acesso em 15 set. 2019.

VIDADESILÍCIO HC06. MÓDULO BLUETOOTH HC-06 E HC-05 – ARDUINO. Disponível em: <https://portal.vidadesilicio.com.br/modulo-bluetooth-hc-05-e-hc-06/>. Acesso em: 27 fev. 2020.

VIDADESILÍCIO RELÉ. Módulo relé – Acionando cargas com Arduino. Disponível em: <https://portal.vidadesilicio.com.br/modulo-rele-com-arduino/>. Acesso em: 15 maio 2019.

WIRED. Build It. Share It. Profit. Can Open Source Hardware Work? . Disponível em: <https://www.wired.com/2008/10/ff-openmanufacturing/>. Acesso em: 09 fev. 2019.

WIRING. Disponível em: <http://wiring.org.co/>. Acesso em: 06 maio 2019.

WUSTL. Bluetooth Module (HC-06) + Arduino. Disponível em: [https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_\(HC-06\)_%2B_Arduino&oldid=12261](https://classes.engineering.wustl.edu/ese205/core/index.php?title=Bluetooth_Module_(HC-06)_%2B_Arduino&oldid=12261). Acesso em: 10 mar. 2019.

APÊNDICE I – Sketch PENDULO_SIMPLES

```

int pinoRELE = 11; // pino de conexão do Rele
int pinoLED = 12; // pino de conexão do LED
int pinoLDR = 0; // pino de entrada do LDR
int valorBase; // valor mínimo da medida do LDR
float taxa = 1.1; // taxa sobre o valor medido no LDR indicativa de início de ciclo
int ciclo; // número de ciclos medidos
int trecho = 0;
int sentido = 0;
int tempo0; // tempo início do ciclo
char buf0 = ' '; // valor anteriormente lido na porta serial

void setup(){
  pinMode(pinoRELE, OUTPUT);
  pinMode(pinoLED, OUTPUT);
  digitalWrite(pinoRELE, LOW);
  digitalWrite(pinoLED, LOW);
  Serial.begin(9600);
  Serial.println("Medindo o período");
}

void loop(){
  lerSerial();
  medirLDR();
  delay(2);
}

void lerSerial(){
  while(Serial.available() > 0){
    char buf = Serial.read();
    if (buf != buf0){
      buf0 = buf;
      if (buf == '0'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, LOW);
        Serial.println((String)"Sistema desligado !!!");
      }
      else if (buf == '1'){
        digitalWrite(pinoLED, LOW);
        digitalWrite(pinoRELE, HIGH);
        Serial.println((String)"Relé ligado !!!");
      }
      else if (buf == '2'){
        ciclo = 0;
        trecho = 0;
        sentido = 0;
      }
    }
  }
}

```

```

    digitalWrite(pinoLED, HIGH);
    delay(1000);
    digitalWrite(pinoRELE, LOW);
    Serial.println((String)"Período (segundos):");
    valorBase = map(analogRead(pinoLDR),0,1023,0,100);
  }
}
}
}

```

```

void medirLDR(){
  int valor = map(analogRead(pinoLDR),0,1023,0,100);
  if (valor > taxa * valorBase){
    if (trecho == 0){
      trecho = 1;
      sentido = 0;
      int tempo = millis();
      if (ciclo > 0){
        float periodo = (tempo-tempo0) / 1000.;
        Serial.println((String)periodo);
      }
      tempo0 = tempo;
      ciclo = ciclo + 1;
    }
    else if (trecho == 2){
      trecho = 1;
      sentido = 1;
    }
  }
  else{
    if (trecho == 1){
      if (sentido == 0){
        trecho = 2;
      }
      else{
        trecho = 0;
      }
    }
  }
}
}

```


APÊNDICE II – Ficha de Experimento do Pêndulo Simples

Escola Técnica Estadual Cícero Dias

Educandos:

Disciplina: Física

Turma: _____

Data: ___/___/_____

FICHA DE EXPERIMENTO

Atividade:

A expressão de dependência entre o período e o comprimento de um pêndulo simples.

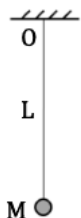
1. ETAPA PRÉ-EXPERIMENTAL

Organização

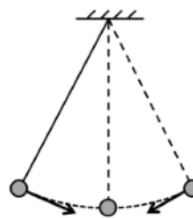
1.1. Objetivo

Determinar a expressão que relaciona o período com o comprimento de um pêndulo simples a partir da medição experimental dos períodos de pêndulos simples com diversos comprimentos.

1.2. Introdução



Pêndulo em repouso.

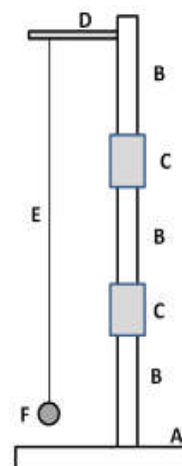


Pêndulo em movimento oscilatório em torno da posição de equilíbrio.

Um pêndulo simples é constituído por um objeto de massa M concentrada e suspenso em um ponto fixo O por um fio inextensível e de comprimento L . Quando afastado de sua posição de equilíbrio e abandonado, o corpo oscila em torno desta posição com um período de tempo T .

1.3. Material utilizado no dispositivo do pêndulo

- A. Uma base de madeira tendo dimensões de 20 cm x 30 cm x 2 cm aproximadamente e com um furo circular de raio 10 cm.
- B. Três hastes de madeira com 40 cm de comprimento e raio de 10 mm.
- C. Dois pedaços de cano PVC de 20 mm e 15 cm de comprimento.
- D. Uma haste metálica com 15 cm de comprimento com fixador para tubos de PVC de 20 mm.
- E. 1,5 m de fio de nylon fino.
- F. Uma esfera de aço com aproximadamente 1 cm de diâmetro.



Outros materiais

- a) Aparato de medição de tempo composto por placa arduino, sensor de luz, solenóide, LED e placa Bluetooth.
- b) Smartfone (ou tablet, ou computador) com apps de comunicação serial por Bluetooth e planilha eletrônica.
- c) Fita métrica com 1,5 m.
- d) Lápis.
- e) Borracha.
- f) Régua.

1.4. Montagem

Preparação da parte mecânica:

- a) Para facilitar o transporte, a haste de madeira do pêndulo é constituída por três pedaços que são facilmente encaixados um ao outro usando os tubos de PVC. Deve-se primeiro fixar um deles na base de madeira e só depois fazer o encaixe dos outros dois.
- b) Amarrar num extremo da linha de nylon a esfera de aço. Uma sugestão é envolver a esfera em um pedaço de pano ou plástico e amarrá-lo com o nylon.
- c) Amarrar o outro extremo da linha de nylon na haste metálica.
- d) Encaixar a haste metálica na haste de madeira a uma altura de aproximadamente 50 cm.
- e) Colocar o aparato de medição sobre a base de madeira e diretamente abaixo da esfera metálica.

1.5. Hipótese

O período do pêndulo é proporcional a uma potência de seu comprimento:

$$T = c \cdot L^b$$

Então

$$\log T = \log(c \cdot L^b)$$

$$\log T = \log c + \log L^b$$

$$\log T = \log c + b \cdot \log L$$

Fazendo

$$y = \log T$$

$$a = \log c$$

$$x = \log L$$

Encontramos $y = a + b \cdot x$

Ou seja, o logaritmo do período é uma função do primeiro grau do logaritmo do comprimento do pêndulo.

2. ETAPA EXPERIMENTAL

Coleta de dados

2.1. Procedimento de coleta de dados

- a) Ajuste a distância L entre a haste metálica e o centro da esfera para ter 50 cm.
- b) Afastar a esfera da vertical aproximadamente 15 cm, tendo o cuidado de deixar a linha de nylon esticada;
- c) Acionar o smartphone para liberar a esfera e medir o valor do período T_I da oscilação;
- d) Repetir outras três vezes para fazer as medidas de T_{II} , T_{III} e T_{IV} , registrando na Tabela 1;
- e) Repetir a medida do período para todos tamanhos de L na Tabela 1;
- f) Calcular o período médio da oscilação para cada comprimento na Tabela 1.

2.2. Tabela de registro de dados

$$T = \frac{T_I + T_{II} + T_{III} + T_{IV}}{4}$$

| Tabela 1 | | | | | |
|----------|-----------|--------------|---------------|--------------|-------|
| L (m) | T_I (s) | T_{II} (s) | T_{III} (s) | T_{IV} (s) | T (s) |
| 0,50 | | | | | |
| 0,60 | | | | | |
| 0,70 | | | | | |
| 0,80 | | | | | |
| 0,90 | | | | | |
| 1,00 | | | | | |

OBS: Trabalhar com apenas duas casas decimais

3. ETAPA PÓS-EXPERIMENTAL

Análise de dados

3.1. Conversão das medidas para seus logaritmos

Como visto em 1.5:

Hipótese: $T = c \cdot L^b$

Então $\log T = \log(c \cdot L^b)$
 $\log T = \log c + \log L^b$
 $\log T = \log c + b \cdot \log L$

Fazendo $y = \log T$
 $a = \log c$
 $x = \log L$

Encontrá-se $y = a + b \cdot x$

Para achar a e b, preencha a tabela 2 seguindo as orientações a seguir:

Transcreva para a coluna 3 todos os períodos para cada comprimento do pêndulo da tabela 1;

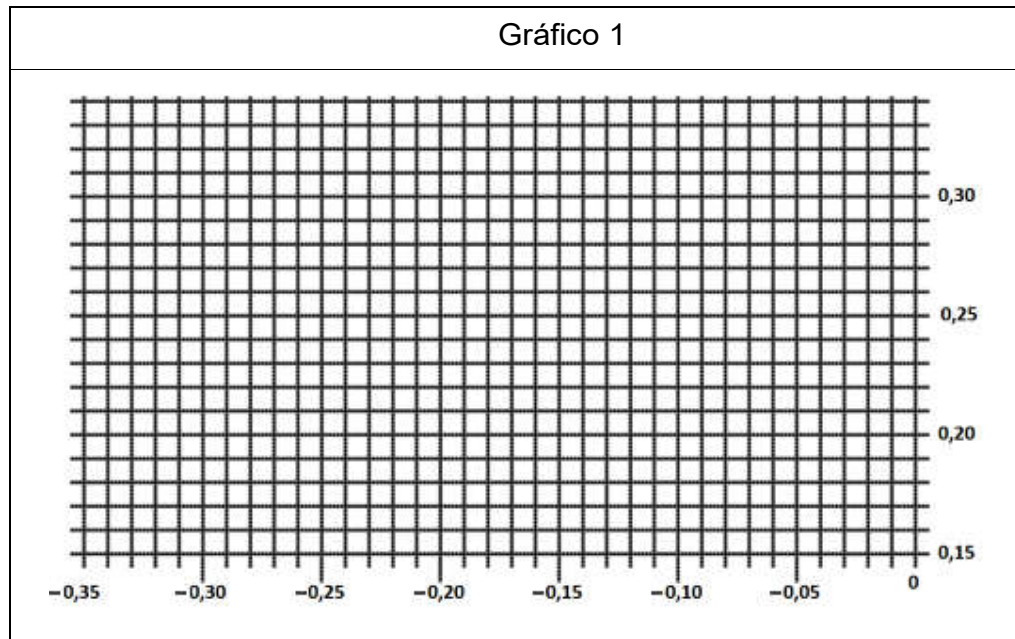
A. Calcule $x_i = \log L_i$ para cada comprimento e registre o resultado na coluna 4;

B. Calcule $y_i = \log T_i$ para cada período e registre o resultado na coluna 5;

| Tabela 2 | | | | |
|----------|-----------|-----------|-------|-------|
| i | L_i (m) | T_i (s) | x_i | y_i |
| 1 | 0,50 | | | |
| 2 | 0,60 | | | |
| 3 | 0,70 | | | |
| 4 | 0,80 | | | |
| 5 | 0,90 | | | |
| 6 | 1,00 | | | |

3.2. Fazendo o gráfico x por y

- a) Transcreva os valores de x e y para o Gráfico 1;
 b) Trace a melhor reta para os pontos no Gráfico 1;



3.3. Obtendo a expressão de dependência a partir do gráfico

- a) Como $y = a + b \cdot x$, o valor de a é igual ao valor de y para $x = 0$, assim

$$a = \underline{\hspace{2cm}}$$

- b) Selecione dois pontos (x_I, y_I) e (x_{II}, y_{II}) sobre a reta e calcule

$$b = \frac{y_{II} - y_I}{x_{II} - x_I} = \underline{\hspace{2cm}} = \underline{\hspace{2cm}}$$

- c) Calcule o valor de c usando

$$a = \log_{10} c \quad \rightarrow \quad c = 10^a = \underline{\hspace{2cm}}$$

- d) Finalmente escreva a relação

$$T = c \cdot L^b \quad \rightarrow \quad \mathbf{T = \underline{\hspace{1cm}} \cdot L}$$

3.4. Comparação Teoria x Prática

A expressão teórica é:

$$T = 2\pi \sqrt{\frac{L}{g}}$$

$$T = \frac{2\pi}{\sqrt{g}} \sqrt{L}$$

$$T \approx \frac{2 \cdot 3,14}{\sqrt{9,8}} \sqrt{L}$$

$$T \approx 2,007 \cdot L^{0,5}$$

A expressão experimental é:

$$T = \text{_____} \cdot L$$

3.5 Observações e Conclusões
