



UNIVERSIDADE FEDERAL DO TOCANTINS
CAMPUS UNIVERSITÁRIO DE ARAGUAÍNA
CURSO DE LICENCIATURA EM FÍSICA

CLEITON MALVESSI

**A UTILIZAÇÃO DO ARDUINO COM OBJETOS DE APRENDIZAGEM
NO ENSINO DE FÍSICA**

ARAGUAÍNA

2016

CLEITON MALVESSI

**A UTILIZAÇÃO DO ARDUINO COM OBJETOS DE APRENDIZAGEM
NO ENSINO DE FÍSICA**

Trabalho de conclusão de curso submetido ao curso de Licenciatura em Física da Universidade Federal do Tocantins, como requisito parcial para a obtenção de título de Licenciado em Física.

Orientador: Prof. Dr. Matheus Pereira Lobo.

ARAGUAÍNA

2016

CLEITON MALVESSI

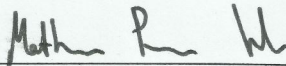
A UTILIZAÇÃO DO ARDUINO COM OBJETOS DE APRENDIZAGEM
NO ENSINO DE FÍSICA

Trabalho de conclusão de curso submetido ao curso de Licenciatura em Física da Universidade Federal do Tocantins, como requisito parcial para a obtenção de título de Licenciado em Física.

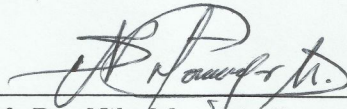
Orientador: Prof. Dr. Matheus Pereira Lobo.

Aprovado em: 06 / 07 / 2016.

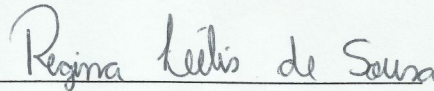
BANCA EXAMINADORA



Prof. Dr. Matheus Pereira Lobo (orientador)



Prof. Dr. Nilo Mauricio Sotomayor Choque



Prof. Dr. Regina Lélis de Sousa

Dedico esse trabalho aos meus pais, Adão e Vilma, em que o bom exemplo constitui o melhor e mais eficaz sistema de educar seus filhos. Dedico também aos meus queridos e preciosos irmãos, Leandro e Cleide.

AGRADECIMENTOS

Agradeço a Deus por ter me conduzido até este momento, concedendo dom e sabedoria, por ter me dado a força para suportar as inúmeras dificuldades imposta no decorrer do curso.

A minha família por ter me apoiado e sempre ter acreditado no meu potencial. Especialmente a minha mãe, presente e solícita durante toda graduação. Aos meus irmãos que me apoiaram e direcionaram com suas experiências de vida.

Ao meu orientador que me orientou com competência e flexibilidade, pela amizade e pelo trabalho de orientação, dando o suporte e incentivo necessário para a elaboração e concretização do projeto final.

Aos professores que compõem o quadro docente, que durante meus quatro anos de universidade fizeram um belo trabalho na contribuição à minha formação profissional, sendo um marco na minha vida acadêmica. Foi de grande valia para o entendimento dos fenômenos tratados neste trabalho.

Aos meus amigos e colegas que sempre me apoiaram nos momentos em que precisei, por terem me acompanhado e apoiado neste processo tão importante, em especial a Josiane S. Nunes, Ana Paula L. Basto, Paola Seidel, Michele Soares, Antônio José G. Carvalho, Welliton L. da Silva, Geisson Dos S. Bastos, Eduardo D. R. Sousa, João Pedro A. Sales.

E a todos que colaboraram de alguma forma com o desenvolvimento e a realização deste trabalho.

A falsa ciência gera ateus; a verdadeira ciência leva os homens a se curvar diante da divindade.

Voltaire

RESUMO

Este trabalho apresenta uma alternativa para os principais desafios encontrados no desenvolvimento de materiais instrucionais de baixo custo para a realização de experimentos de Física correlacionados aos conteúdos trabalhados nos livros didáticos acompanhados com a plataforma de simulação PhET, utilizando a placa Arduino UNO como gerenciador dos experimentos. O material instrucional foi desenvolvido amparado na Teoria de Aprendizagem Significativa de David Ausubel e na metodologia PIE (Predizer, Interagir e Explicar). Por se tratar de um material didático de baixo custo, elaborado dentro de um conceito de Recursos Educacionais Abertos (REA), com uso de softwares livres e hardware aberto, é altamente disseminável e muito útil para o desenvolvimento de projetos e práticas educacionais no Ensino de Física. Atribuímos licenças permissivas a todo o material desenvolvido neste trabalho a fim de que professores, pesquisadores, estudantes e demais membros da sociedade possam usar, estudar, modificar e compartilhar livremente todo o seu conteúdo.

Palavras-chave: Arduino. Ensino de Física. Aulas experimentais. PhET.

ABSTRACT

This paper presents an alternative to the main challenges encountered in the development of low-cost instructional materials for conducting physics experiments related to the content worked in textbooks accompanied with PhET simulation platform using the Arduino UNO board as manager of the experiments. The instructional material was developed supported the theory of David Ausubel meaningful learning and PIE methodology (Predicting, Interact and explain). Because it is a didactic material of low cost, designed within the concept of Open Educational Resources (OER), using free software and open hardware, is highly disseminável and very useful for the development of educational projects and practices in Education Physical. We attribute permissive licenses to all the material developed in this work so that teachers, researchers, students and other members of society to use, study, modify and freely share all of your content.

Keywords: Arduino. Physics Teaching. Experimental classes. PhET.

Sumário

1	INTRODUÇÃO	10
2	Fundamentação teórica	15
2.1	Delineamento Metodológico	15
2.1.1	Teoria da Aprendizagem Significativa de David Ausubel	15
2.1.2	Metodologias PIE e POE	16
2.2	Computação Física/Sistemas Embarcados	17
2.2.1	Arduino	18
2.2.2	Programação	28
2.3	Simulações como Objetos de Aprendizagem	37
2.3.1	PhET	38
2.3.2	Alcance das Simulações	39
3	Materiais e Método	41
3.1	Materiais	41
3.2	Metodologia	42
4	Roteiro Experimental	43
4.1	Óptica	43
4.1.1	Guia Pedagógico - Formação das Cores	43
4.1.2	Conceitos teóricos	44
4.1.3	Simulação	48
4.1.4	Prática	49
4.1.5	Guia de Atividade - Formação das Cores	52
5	Considerações Finais	53

Lista de Figuras

2.1	Arduino UNO Rev. 3. Representação da frente e do verso da placa.	18
2.2	Entradas e seus respectivos Adaptadores/Conectores.	20
2.3	Conectores de alimentação: Saídas 3, 3V, 5V e a entrada Vin.	20
2.4	Os 14 pinos digitais.	21
2.5	Os 6 pinos analógicos.	22
2.6	Resumo dos recursos.	23
2.7	Valor de degrau para uma conversão em 10 bits com referência em 5V.	24
2.8	Junção dos módulos: I2C e LCD.	26
2.9	Ligação dos respectivos pinos do módulo I2C com o LCD ao Arduino.	27
2.10	Desenho esquemático de uma Protoboard.	27
2.11	Principais símbolos e seus significados.	31
2.12	Fluxograma simples da troca de uma lâmpada com repetição.	31
2.13	Caracterização do ambiente de desenvolvimento do Arduino (IDE).	35
2.14	Texto de um programa redigido na IDE do Arduino.	37
4.1	Sistema aditivo RGB em monitor LCD, cada pixel é uma sequência das três cores base do sistema.	45
4.2	Síntese do sistema aditivo.	46
4.3	Características das fontes luminosas.	47
4.4	Formação das cores, fontes de luz e o uso de filtros.	48
4.5	Esquema de montagem do experimento e o resultado esperado na montagem.	51

Lista de Tabelas

2.1	Resumo das especificações técnicas.	22
2.2	Operadores	34
3.1	Material de base	42
4.1	Intensidades das cores formadas no sistema aditivo de cores (R,G,B)	46
4.2	Materiais utilizados na prática de óptica.	49

Capítulo 1

INTRODUÇÃO

Este trabalho contempla minha trajetória durante a graduação em licenciatura em Física. Durante meus quatro anos de graduação, tive a oportunidade de participar do Programa Institucional de Bolsas de Iniciação à Docência (PIBID) e do Programa de Iniciação Científica Voluntária (PIVIC), esses dois programas e claro todas as disciplinas em que cursei contribuíram significativamente na construção da minha didática pessoal frente aos desafios em que eu irei enfrentar como futuro professor. Nesse trabalho vou apresentar uma síntese do que eu estudei, aprendi e coloquei em prática.

Em minha trajetória no PIBID, que foram de aproximadamente 24 meses, tive a oportunidade de atender um dos colégios de Araguaína, cidade base de um dos sete campi da Universidade Federal do Tocantins. Neste período de atendimento ao colégio, enfrentei inúmeras dificuldades com relação à transmissão de conteúdos para aqueles alunos que compareciam as tutorias. A maioria dos alunos já tinha uma mentalidade formada com relação à disciplina, de que era difícil, chata e sem serventia ou que era mais uma extensão da matemática.

Sendo assim, percebi que minha didática precisava de uma revolução urgente, para revelar como a Física e a Matemática está a invadir as nossas vidas e está por trás de tudo. Muitas pessoas, de diferentes gerações, dizem que odeiam Física. **Por que razão isso acontece?** Muita gente tem uma relação traumática com a disciplina. Uma das razões tem a ver com o fato de que o ensino da Física, tal como é feito na grande maioria das escolas, dar demasiada ênfase à resposta, ao número e não a compreensão dos resultados.

O ensino baseia-se quase exclusivamente em testes e em ver quem é mais rápido a encontrar a resposta. E muitos se sentem confusos e inferiores porque não conseguem fazê-lo. Essa dor fica. Até podem depois não se lembrar do incidente de forma concreta, mas o trauma ficou lá. Por outro lado, a escola não expõe os alunos à verdadeira beleza da Física. **Por quê?** A maioria dos conteúdos que são ensinados nas aulas tem mais de dois séculos e isso é verdadeiramente escandaloso e seria impensável para as aulas de

ciências (CAVALCANTE; BONIZZIA, GOMES, 2009).

Seria o mesmo que continuarmos a ensinar os alunos que a Terra é plana ou que é o Sol que gira em volta dela. Continuamos a repetir fórmulas antigas, sem estabelecer nenhuma ponte com o mundo atual. Para a maioria das pessoas, a Física parece demasiadamente abstrata, uma extensão da Matemática, desnecessária, sem aplicação prática. Isso porque infelizmente são ensinadas dessa forma.

Mas toda a tecnologia está cada vez mais presente nas nossas vidas e todos os indivíduos estão compelidos no mundo moderno a sua utilização, incluindo computadores, internet, *smartphones*, videogames, etc. Tudo isso se baseia na Matemática e em princípios físicos. Por trás de todo caixa eletrônico ou dos *gadgets*¹ estão algoritmos muito sofisticados. É como se fôssemos escravos desses algoritmos. Por isso, é fundamental que os compreendamos para não sermos manipulados por eles. Muitas vezes a escola possui um laboratório de informática, mas a sua utilização acaba ficando restrita à digitação de trabalhos e pesquisas na internet, não ultrapassando o uso de tabelas para a construção de gráficos (SOARES; BORGES, 2010).

O que tem então de mudar no ensino? É preciso uma revolução. Temos de preservar o sentido de mistério e de descoberta que existe na Física e apresentá-la aos alunos quase como um romance policial. E tem de haver paixão por parte dos professores. Eles próprios têm de amar a Física. Além disso, é fundamental mudar o currículo, para incluir conteúdos mais modernizados e relacioná-los com o mundo real.

Neste momento, querer aprender Física não é uma questão de escolha. É uma questão de necessidade, porque ela está literalmente a invadir as nossas vidas e nos colocamos em risco ao sermos ignorantes. As pessoas não sabem e pior do que isso, têm medo de perguntar. Está em curso uma reestruturação profunda do mundo e da forma como interagimos uns com os outros e com a tecnologia, é preocupante que as pessoas não estejam a prestar atenção ao que está a acontecer.

Quando entramos em uma loja virtual como a Amazon, seus anúncios fazem-nos recomendações de livros e as pessoas seguem-nas, muitos sem questionar. Não percebem que por trás disso há algoritmos que podem ser manipulados, tanto por questões financeiras, porque há empresas que pagam para os seus livros serem recomendados e para outros não aparecerem, como por razões políticas, ideológicas, para que sejam divulgadas certas ideias e não outras.

A empresa propõe em um futuro próximo enviar encomendas aos seus clientes por sistemas automatizados, os *Drones*,² que pode ser perigoso caso ocorra alguma falha ou até alterados os destinos da própria encomenda. O fato de ser perigoso está relacionado

¹Dispositivos eletrônicos portáteis.

²São todo e qualquer tipo de aeronave que não seja tripulada, mas comandada por seres humanos a distância.

à vulnerabilidade dos sistemas que podem ser explorados por pessoas com más intenções.

É necessário uma conscientização, e para proporcionarmos isso, esta nova realidade deve ser estudada com os jovens e adolescentes com métodos diferenciados dos convencionais. O estudo da Física tem um papel muito importante em desvendar e compreender melhor essa nova revolução da informação.

Durante o período em que estive no PIVIC tive a oportunidade de desenvolver projeto com a placa de prototipagem Arduino UNO, percebi que essa plataforma pode ser um laboratório completo para uma professor. Quando explorada com sensores e atuadores, é possível trabalhar os conteúdos de Física, de uma forma divertida e produtiva.

Neste trabalho vou apresentar alguns experimentos que foram montados com a plataforma Arduino que tem como objetivos: chamar a atenção dos alunos aos fenômenos envolvidos, proporcionar uma aprendizagem significativa, facilitar a transmissão de conhecimento. Esses experimentos podem ser assistidos com simulações, ou seja, o professor pode trazer uma simulação seja da plataforma PhET ou outro como um aplicativo de celular que demonstre os fenômenos envolvidos no experimento, podem ser realizadas antes, durante ou depois do experimento, e tem como finalidade explorar aspectos que não sejam evidentes.

Essa abordagem é um complemento e segue o conteúdo programático dos livros didáticos. Minha intenção neste trabalho é mostrar que tal abordagem é possível e de fácil acesso a todos que desejam inovar suas aulas. Muitas vezes o desejo de trazer esse tipo de abordagem é presente nos professores, mas a estrutura da escola não lhe proporciona um ambiente adequado, com um laboratório que possua equipamentos que muitas das vezes são caros e de difícil manipulação.

Se juntarmos o conhecimento de um professor de Física à utilização do Arduino somado com sensores e atuadores é suficiente para reproduzir a maioria dos equipamentos gastando muito pouco, o que muitas das vezes pode haver uma conscientização de reaproveitamento de eletroeletrônico para se implementar essas aulas. O que pode talvez desmotivar os interessados muitas das vezes é a necessidade de programar o Arduino conforme seu modelo montado. Mas existem diversas fontes de exemplos dos programas utilizados na plataforma Arduino, com exemplos comentados que podem ser adaptados com muita facilidade, e com pouco tempo se adquire habilidade para fazer seus próprios *sketches*³.

Considerando esse contexto, apresentamos no presente trabalho uma proposta voltada para o ensino de Física a partir de experimentos didáticos de baixo custo que podem gerar dados em tempo real, e que posteriormente podem ser utilizados para a visualização de gráficos e interpretações. Desta forma, é possível fazer teste de hipóteses sobre os

³Um *sketch* é o nome que o Arduino usa para um programa. É a unidade de código que é enviado para a memória em uma placa Arduino.

fenômenos físicos em estudo, correlacionando com os pressupostos teóricos que são estabelecidos na Física.

O produto educacional desenvolvido neste trabalho é composto por: uma placa Arduino UNO, *protoboard* para montagem dos circuitos, sensores para detectar os fenômenos estudados, atuadores para manipularmos determinadas grandezas físicas, *Shield* LCD que serve como interface para exposição dos dados, *Shield* I2C para comunicação do LCD com a placa Arduino, *software* livre para a usarem como o Arduino e guia de atividade para professores e alunos.

Todo o material instrucional desenvolvido neste projeto de Ensino é baseado no conceito de Recursos Educacionais Abertos (REA) o que possibilita professores, alunos ou qualquer outro interessado fazer a utilização deste material sem nenhum tipo de custo relacionado à aquisição da licença de uso, dando a todos a liberdade para usar, estudar, modificar e distribuir o material de acordo com os termos de licenças permissivas. O guia de atividades foi desenvolvido tendo como base a Teoria de Aprendizagem Significativa de David Ausubel e a metodologia Predizer, Interagir e Explicar (PIE) (DORNELES, 2010).

Nesta metodologia, os alunos são motivados a predizer por escrito, através da resposta às questões presentes no guia, o comportamento de um determinado fenômeno físico a partir da visualização de um experimento realizado pelo professor. Em seguida, eles podem interagir com o equipamento e, discutindo em grupo, buscar explicações para as possíveis divergências ou convergências entre as predições que fizeram e o que foi efetivamente observado no experimento. Todos os programas utilizados para a prática estão em anexos e todas as informações necessárias para o professor fazer a prática é detalhada neste TCC.

Segundo Marandino et al (2004), “que a educação em ciências é uma prática social que vem sendo cada vez mais ampliada e desenvolvida nos chamados espaços não formais de educação e nas diferentes mídias.” Essa ideia é complementada com o que Gohn (2001) retrata sobre a educação nesses espaços como “um amadramento de elementos que capacitam os indivíduos para a organização em prol dos objetivos da comunidade e solução de problemas coletivos.” Vieira, Bianconi e Dias (2005) complementam que “espaços não formais, onde se procura transmitir ao público estudantil conteúdos de Ciências, podem favorecer a aquisição de tal bagagem cognitiva.” Para Godinho e Ferracioli (2006),

“Os espaços não formais de educação têm o papel de despertar vocações para as carreiras científicas, instigar a curiosidade e a fascinação pelo conhecimento, sendo estratégias utilizadas para reduzir as imensas distâncias entre as atividades rotineiras diárias e as ciências, favorecendo para a cidadania, todas essas leituras justificam que a didática utilizada na composição deste trabalho tem um grande potencial em proporcionar uma aprendizagem significativa.”

No decorrer do texto vamos apresentar uma alternativa para os principais desafios encontrados no desenvolvimento de materiais instrucionais de baixo custo para a realização de experimentos de Física correlacionados aos conteúdos trabalhados nos livros didáticos que podem ainda ser acompanhados com a plataforma de simulação PhET, utilizando a placa Arduino UNO como gerenciador do experimento.

Para isto foi desenvolvido um material didático, organizado da seguinte forma: *I*) Introdução com definição e exemplos de Computação Física, *II*) Referencial teórico contendo três partes, *a*) Delineamento metodológico, *b*) Estudos da Computação Física com guia da placa Arduino UNO Rev. 3 e o detalhamento das principais características e modo de uso, *c*) Utilização de linguagem de programação; *III*) Os principais materiais utilizados no roteiro, *IV*) Guia de atividade, Guia pedagógico para orientar o professor na montagem e aplicação da aula proposta.

Capítulo 2

Fundamentação teórica

Este trabalho fundamenta-se nas seguintes metodologias: Teoria da Aprendizagem Significativa (TAS) de Ausubel (Ausubel, 1980), e na metodologia Predizer, Interagir e Explicar (PIE) de Dorneles (2010) foi criada baseada no método Predizer, Observar e Explicar (POE) proposto por Tao e Gunstone (1999). O referencial requerido para que os experimentos sejam realizados consiste no estudo da computação Física, programação e do uso de simulações.

2.1 Delineamento Metodológico

2.1.1 Teoria da Aprendizagem Significativa de David Ausubel

Da teoria de Ausubel (1980) levamos em conta neste trabalho o conceito de aprendizagem significativa, as condições para sua ocorrência e os princípios de diferenciação progressiva e reconciliação integradora. Ausubel define a aprendizagem significativa como um “processo pelo qual o significado de um novo conhecimento resulta da interação, que ocorre de maneira substantiva e não-arbitrária, entre este e um aspecto especificamente relevante da estrutura cognitiva do aprendiz”, denominado subsunçor. Ausubel também propõe duas condições para que ocorra aprendizagem significativa: *i*) o material deve ser potencialmente significativo, isto é, o conteúdo do material a ser estudado deve ter relação com a estrutura cognitiva do aluno, de maneira não-arbitrária e não-literal e *ii*) o aluno deve manifestar disposição para relacionar o novo material, potencialmente significativo, de forma substantiva e não-literal, à sua estrutura cognitiva.

“A essência do processo de aprendizagem significativa é que as idéias expressas simbolicamente são relacionadas às informações previamente adquiridas através de uma relação não arbitrária e substantiva (não literal). Uma relação não arbitrária e substantiva significa que as idéias são relacionadas a algum aspecto relevante existente na estrutura cognitiva do aluno, como, por exemplo, uma imagem, um símbolo, um conceito ou uma proposição.”(AUSUBEL; NOVAK; HANESIAN, 1980, p.34).

Em relação à dinâmica de sala de aula Ausubel propõe dois princípios programáticos a diferenciação progressiva e a reconciliação integradora.

- Diferenciação progressiva: é o princípio pelo qual o assunto deve ser programado de forma que as ideias mais gerais e inclusivas da disciplina sejam apresentadas antes, e progressivamente diferenciadas. (MOREIRA; MASINI, 1982, p. 21).

Ausubel baseia-se em duas hipóteses: *i*) é mais fácil para seres humanos captar aspectos diferenciados de um todo mais inclusivo previamente aprendido, do que chegar ao todo a partir de suas partes diferenciadas previamente aprendidas; *ii*) a organização do conteúdo de uma certa disciplina, na mente de um indivíduo, é uma estrutura hierárquica na qual as ideias mais inclusivas e gerais estão no topo e, progressivamente, incorporam proposições, conceitos e fatos menos inclusivos e mais diferenciados.

- Reconciliação integrativa: onde “o material instrucional deve ser elaborado para que se possa fazer uma relação entre as ideias, apontar similaridades ou diferenças significativas”. (MOREIRA; MASINI, 1982, p. 21).

2.1.2 Metodologias PIE e POE

Para promover o engajamento cognitivo e a interatividade, concebemos guias segundo um método que denominamos de PIE (Predizer, Interagir e Explicar) que fora adaptado do método POE (Predizer, Observar e Explicar) proposto por Tao e Gunstone (1999). No PIE, inicialmente são apresentadas perguntas sobre a evolução de determinada situação e os alunos são convidados a predizer, antes de qualquer interação com o recurso instrucional, qual será a ocorrência. A seguir devem interagir com a simulação computacional ou com o material experimental para gerarem resultados e então avaliarem o que efetivamente ocorre e, finalmente, devem explicar as divergências e convergências de suas previsões em relação ao que foi observado.

O uso da metodologia PIE para a realização das atividades experimentais tem por finalidade incentivar os alunos a abrir espaços para a discussão de ideias e fazerem previsões

sobre o comportamento das grandezas físicas que fazem parte do experimento, elaborando e testando suas próprias hipóteses na tentativa de explicar e justificar as questões. Durante este processo procuramos promover a diferenciação progressiva dos conceitos, onde os conceitos mais gerais são progressivamente diferenciados em termos de detalhes e especificidade do experimento realizado.

Para alcançar tais objetivos, elaborou-se questões cujas respostas poderiam ser comparadas com os resultados obtidos após a realização das atividades práticas, o que caracteriza a metodologia PIE (Predizer, Interagir e Explicar). Nesse caso observa-se que há o vínculo entre a teoria e a prática, reforçado pelo uso da metodologia PIE nas atividades o que torna o material instrucional potencialmente significativo.

2.2 Computação Física/Sistemas Embarcados

O Arduino é o que chamamos de plataforma de computação física ou embarcada¹, compreendido como um conjunto de sistemas computacionais completos e independentes, mais simples que um computador de propósito geral, encarregado de executar apenas uma função determinada, ou seja, tarefas pré-determinadas, com requisitos específicos, que pode interagir com seu ambiente por meio de *hardware* e *software*.

Em termos práticos, um Arduino é um placa de circuito contendo um pequeno processador que você pode programar para processar entradas e saídas entre o dispositivo e os componentes externos conectados a ele. Estes projetos utilizam entradas e saídas de sensores e atuadores para dialogar com ambientes físicos, especialmente protótipos que interagem com componentes eletrônicos de percepções visuais, sonoras e táteis.

A comunicação se dá pela capacidade de associar dados analógicos e digitais através de um processo chamado transdução². A transdução, segundo Igoe & O'Sullivan (2004), consiste na conversão de formas de energia física em tensão elétrica variável, e recebe este nome devido ao elemento de conversão chamado transdutor. É o processo que ocorre em interfaces como microfones, teclados, mouse, entre outros.

O elemento que torna esse processo possível é o microcontrolador. Trata-se de um dispositivo que faz a mediação entre os *inputs* e *outputs* analógicos e digitais, permitindo a configuração das relações entre ambos os fluxos de energia. Essa comunicação ocorre por elementos de *hardware* e *software*, e o *software* controla os eventos que ocorrem a cada tempo para determinar um plano de interação com o *hardware*. A partir disso, o usuário pode criar relações reais e virtuais entre os aparatos. Ao ser associado a sensores e atuadores, torna possível a obtenção de diferentes resultados na medida em que permite

¹Um sistema embarcado é um sistema microprocessado no qual é dedicado ao dispositivo ou sistema que ele controla, e desenvolvido para uma tarefa específica.

²Processo pelo qual uma energia se transforma em outra de natureza diferente.

a manipulação de várias interfaces ao mesmo tempo.

2.2.1 Arduino

Comercializado sob o nome Arduino, o projeto teve seu início em 2005, na Itália, na busca por interação em projetos escolares, como maneira de reduzir custos dos sistemas de prototipagem da época. Seu sucesso foi sinalizado com o recebimento de uma menção honrosa na categoria *Digital Communities* em 2006, pela *Prix Ars Electronica*, além da marca de mais de 50.000 placas vendidas até outubro de 2008.

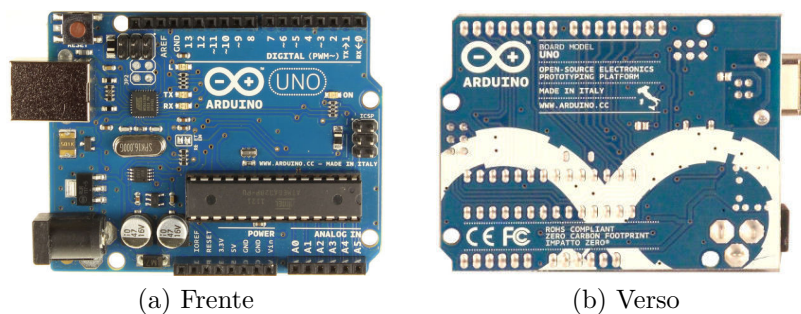


Figura 2.1: Arduino UNO Rev. 3. Representação da frente e do verso da placa.

Arduino é um projeto *open-source*³ fundada por Massimo Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, e David Mellis. Baseia-se no trabalho de muitas pessoas, projetos e instituições. O Arduino foi desenvolvido inicialmente no *Interaction Design Institute Ivrea*, no norte da Itália, onde UNO nos indica que é da primeira família de placas do Arduino e Rev.3 nos informa qual revisão.

Trata-se de uma plataforma de prototipagem eletrônica de hardware livre e de placa única, projetada com um microcontrolador Atmel AVR, com suporte de E/S (entrada e saída) embutido e uma linguagem de programação padrão, com origem em *Wiring*⁴.

O objetivo deste projeto se manteve o mesmo do princípio, que era o de criar ferramentas que são acessíveis, de baixo custo, flexíveis e fáceis de se usar tanto por profissionais, amadores e principalmente para aqueles que não teriam acesso a soluções e ferramentas mais sofisticadas. Uma típica placa Arduino é composta por um controlador, pinos de E/S digital e analógica, além de uma interface serial ou USB, para interligar-se ao computador, com a interface de programação instalada para programá-la e interagir em tempo real. A placa em si não possui qualquer recurso de rede, porém é comum combinar um ou mais Arduinos deste modo, usando extensões apropriadas chamadas de *shields* ou módulos. A interface é simples, podendo ser escrita em várias linguagens. A mais

³Código aberto, que pode ser adaptado para diferentes fins.

⁴É uma plataforma de prototipagem eletrônica de hardware livre composta por uma linguagem de programação, um ambiente de desenvolvimento integrado (IDE) e um microcontrolador.

popular é a própria linguagem do Arduino (que acompanha a Arduino IDE), mas outras que podem comunicar-se com a conexão serial são: Processing, Max/MSP, Pure Data, SuperCollider, ActionScript e Java.

A finalidade do Arduino em um sistema é facilitar a prototipagem de novos produtos e ideias, implementação ou emulação do controle de sistemas interativos, em nível amador, doméstico, comercial ou móvel. Com ele é possível enviar ou receber informações de basicamente qualquer sistema eletrônico, como identificar a temperatura de um ambiente utilizando o sensor LM35 e controlar o tempo em que o ar condicionado fica ligado, reduzindo assim o gasto energético.

Os campos onde podem ser utilizados Arduino e suas funcionalidades são inúmeros, trazem benefícios e vantagens, podendo ter aplicações na área de impressão 3D, robótica, engenharia de transportes, engenharia agrônômica e musical, etc.

O projeto Arduino é agora suportado por uma empresa internacional, com escritórios e pessoas de todo o mundo. O Arduino leva o nome de um bar em Ivrea, que foi nomeado após um antigo rei da Itália de Ivrea.

Características

Nessa seção vamos abordar as principais características do Arduino Uno, uma vez que as especificações do fabricante são indispensáveis. Para preservar a estabilidade e confiabilidade em seu funcionamento é preciso observar os parâmetros dos dispositivos contidos. A placa é equipada de um microcontrolador baseado no ATmega328 chip utilizado na confecção, que é o centro de processamento das entradas e saídas que são 14 pinos de E/S dos quais 6 digitais que podem ser usados como saídas PWM⁵, 6 entradas analógicas, um cristal oscilador de 16MHz, uma conexão USB, uma entrada de alimentação uma conexão ICSP⁶ e um botão de reset.

A placa contém todos os componentes necessários para suportar o microcontrolador, mas devemos ficar atentos à tensão de alimentação que pode ser por uma conexão com a porta USB ou alimentada por uma fonte com a tensão recomendada de 7 a 12V que tenha um plugue P4 macho ou com uma bateria de 9V utilizando um conector.

⁵“*Pulse Width Modulation*” ou Modulação de Largura de Pulso, através da largura do pulso de uma onda quadrada é possível o controle da tensão.

⁶ICSP significa *In Circuit Serial Programming*. É um protocolo de comunicação. Este protocolo é usado na maioria dos gravadores existentes para a maioria dos microcontroladores. Neste protocolo utiliza-se apenas 5 sinais: VCC (+ Alimentação) - GND (- Alimentação) - VPP (Tensão de gravação) - CLOCK (Sinal sincronismo) e DATA (é o dado propriamente dito).

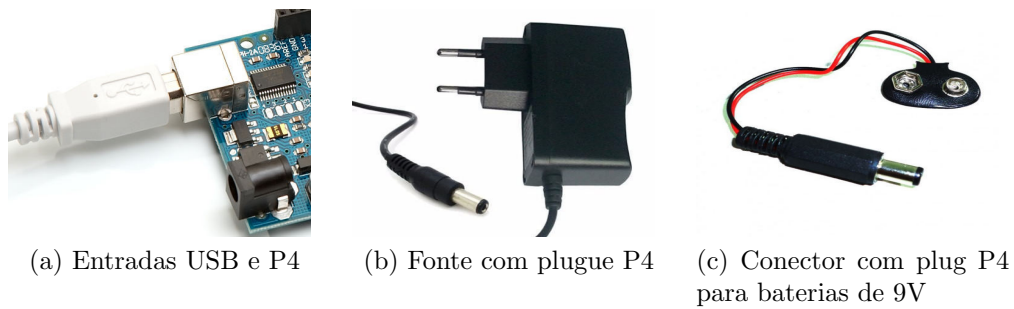


Figura 2.2: Entradas e seus respectivos Adaptadores/Conectores.

A alimentação é selecionada automaticamente. Alimentação externa (não USB) pode ser tanto de um adaptador CA⁷ para CC⁸ ou bateria. Há um conector para alimentação de 2,1mm com o positivo no centro. Cabos vindos de uma bateria podem ser inseridos diretamente nos pinos Gnd⁹ e Vin¹⁰ do conector de alimentação, (ver figura 2.3). Esta placa pode funcionar com uma fonte de alimentação externa de 6 a 20 volts. No entanto, se a alimentação for inferior a 7V, o pino 5V pode fornecer menos de cinco volts e a placa pode se mostrar instável. E se a alimentação for maior do que 12V, o regulador de voltagem pode superaquecer e danificar a placa. A faixa recomendada é de 7 a 12 volts.

Os pinos de saída que vão fazer a alimentação do projeto pode ser de 5V ou 3,3V, a depender dos dispositivos utilizados. A fonte de 3,3V é gerada pelo regulador embarcado e a corrente máxima suportada é de 50mA. A fonte de 5V fornece tensão de 5V também regulada para alimentação de *shields* e circuitos externos.

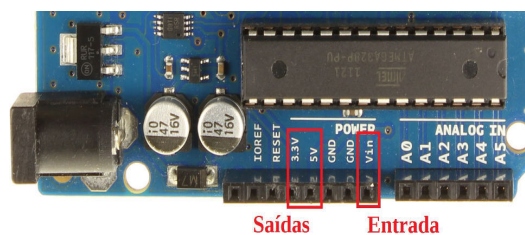


Figura 2.3: Conectores de alimentação: Saídas 3,3V, 5V e a entrada Vin.

Na figura 2.3 observa-se mais três pinos: Reset, esse pino permite que coloquemos um botão externo para resetar o microcontrolador. IOREF, esse pino fornece uma tensão de referência para que *shields* possam selecionar o tipo de interface apropriada. Dessa forma, *shields* que funcionam com a placa Arduino que são alimentadas com 3,3V podem se adaptar para ser utilizados em 5V e vice-versa. Os pinos GNDs mostrado na figura 2.3 são os dois de três pinos terra que a placa possui.

⁷Corrente alternada

⁸Corrente continua

⁹Pinos terra.

¹⁰Entrada de alimentação para a placa Arduino quando se está utilizando uma fonte de alimentação externa ligado direto na placa.

Cada um dos quatorze pinos digitais do Uno pode ser utilizado como uma entrada ou uma saída, basta que se faça as devidas declarações no programa, seis destes podem ser utilizados para gerar pulsos PWM, caso deseje usar os demais como PWM dese ser indicado no programa. Eles operam a 5V. Cada pino pode fornecer ou receber um máximo de 40mA. Além disso alguns pinos tem funções especializadas.

Os pinos (0) → (RX) e (1) → (TX) são usados para comunicação, sendo que o pino (0) para receber e o pino (1) para transmitir dados seriais. Estes pinos são conectados aos ATmega328P que se encarregam de fazer a conversão serial para USB. LEDs conectados ao (RX) e (TX) piscarão enquanto dados estiverem sendo transmitidos pela conexão USB mas não para comunicação serial direto nos pinos (0) e (1). Ainda sobre a comunicação do Arduino, ele pode ser utilizado para ser conectado a um computador, a uma rede, ou até mesmo à Internet para recuperar e enviar dados do Arduino e atuar sobre eles. Em outras palavras, ele pode enviar um conjunto de dados recebidos de alguns sensores para um site, dados estes que poderão, assim, ser exibidos na forma de um gráfico por exemplo.

Os pinos (2) e (3) → (I2C) podem ser configurados para disparar uma interrupção de acordo com alguma variação sensível pelo circuito, para maiores detalhes ver a função (*attachInterrupt()*). Os pinos (4) → (SDA) e (5) → (SCL), fornecem suporte à comunicação I2C utilizando a biblioteca Wire. Os pinos (10) → (SS), (11) → (MOSI), (12) → (MISO), (13) → (SCK), dão suporte à comunicação SPI utilizando a biblioteca SPI. No pino (13) há um LED integrado ao pino digital. Quando este pino está no valor *HIGH*, este LED está aceso, quando o pino está em *LOW*, o LED está apagado.

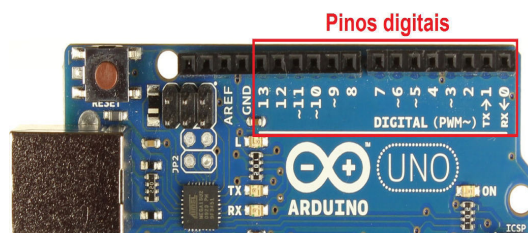


Figura 2.4: Os 14 pinos digitais.

Na figura 2.4 observa-se o terceiro GND e o Pino AREF, esse pino pode ser usado como voltagem de referência para as entradas analógicas, utilizado com a função (*analogReference()*).

A placa ainda conta com 6 entradas analógicas, etiquetadas de A0 a A5, cada uma tem 10 bits de resolução (1024 valores diferentes). Por padrão esses pinos analisam valores de 0 a 5V, embora seja possível alterar o limite superior utilizando o pino AREF e a função (*analogReference()*).

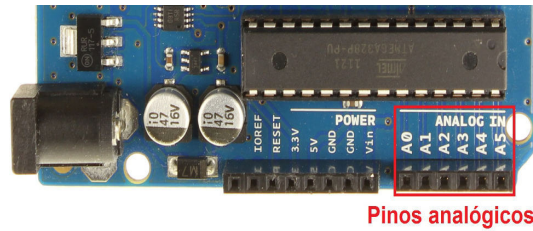


Figura 2.5: Os 6 pinos analógicos.

As dimensões da PCB¹¹ do Arduino Uno são 68,58 de largura e 53,34mm de comprimento, respectivamente, os conectores USB e de alimentação estendendo-se além destas dimensões. A placa ainda conta com quatro orifícios para parafusos que permitem sua fixação a uma superfície ou *case* não condutor. Observado a figura 2.1b percebe-se que as soldas não são isoladas, deve ser utilizado sempre um superfície não condutora para evitar um curto no circuito.

O chip utilizado na confecção, ATmega328P, tem 32KB (dos quais 0,5 são utilizados pelo *bootloader*¹², também tem 2KB de SRAM e 1KB de EEPROM (que pode ser lido ou gravado com a biblioteca EEPROM).

Na tabela 2.1 e na figura 2.6 a seguir temos os resumos das principais características de uma placa Arduino UNO Rev. 3.

Tabela 2.1: Resumo das especificações técnicas.

Microcontrolador	ATmega328P
Tensão Operacional	5 V
Tensão de entrada (Recomendado)	7 - 12 V
Tensão de entrada (limite)	6 - 20 V
Pinos de E/S Digitais	14 (6 destes são PWM)
Pinos de entrada analógica	6
Corrente máxima para o pino 5 V	40 mA
Corrente máxima Para o pino 3,3 V	50 mA
Memória flash	32 KB, 0,5 KB é utilizado pelo bootloader
Memória SRAM	2 KB
Memória EEPROM	1 KB
Velocidade de Processamento	16 MHz

¹¹É a abreviação de “Printed circuit board”, ou seja, placa de circuito impresso.

¹²Programa específico para inicialização de um sistema.

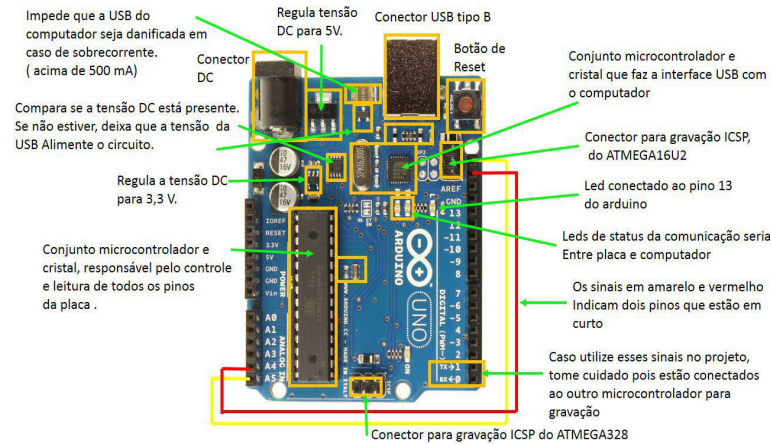


Figura 2.6: Resumo dos recursos.

Conversão Analógico/Digital

Na seção anterior vimos que as entradas digitais só podem assumir dois estados, *HIGH* e *LOW*, ou seja, 5V ou 0V. De modo que só é possível ler apenas dois estados. Por exemplo, verificar se uma porta está aberta ou fechada, identificar se um botão está pressionado ou não, etc. Com as entradas digitais você aplica lógica discreta para controle de seus projetos, porém em muitas situações a variação das grandezas envolvidas acontece de forma analógica, ou seja, variam continuamente em relação ao tempo e podem assumir infinitos valores dentro de uma faixa. Como por exemplo temperatura, pressão e umidade são grandezas que variam dessa forma.

No entanto, o microcontrolador do Arduino (ATmega328P) trabalha internamente com dados digitais, ((0) *LOW* e (1) *HIGH*). Portanto, é necessário traduzir o sinal analógico para um valor digital. A técnica utilizada para leitura de um sinal analógico pelo Arduino é a conversão analógica digital (A/D), em que converter o sinal analógico para um valor digital. Dessa forma se pode quantificar o sinal presente no pino. Esse processo é feito pelo conversor Analógico digital, ADC ou conversor A/D.

O conversor A/D quantifica o valor analógico conforme a quantidade de bits da sua resolução. A resolução de um conversor A/D é dada pela seguinte equação:

$$\text{Resolução} = \frac{V_{Ref}}{2^n}, \quad (2.1)$$

Onde V_{Ref} é a tensão de referência utilizada no conversor A/D; (n) é número de bits do conversor.

O microcontrolador ATmega328P possui 10 bits de resolução e pode trabalhar com três tipos distintos de referência, 5V (V_{CC}) da placa, 1,07V referência interna selecionável ou ainda uma referência externa. O ranger de tensão de entrada pode variar de 0V até o valor de V_{CC} .

Dessa forma quando está trabalhando com a referência em VCC, o menor valor que pode ser lido será:

$$\text{Resolução} = \frac{5V}{2^{10}} = \frac{5V}{1024} \approx 4,89\text{mV}. \quad (2.2)$$

O valor $\approx 4,89\text{mV}$ representa o degrau para uma conversão em 10 bits com referência em 5V.

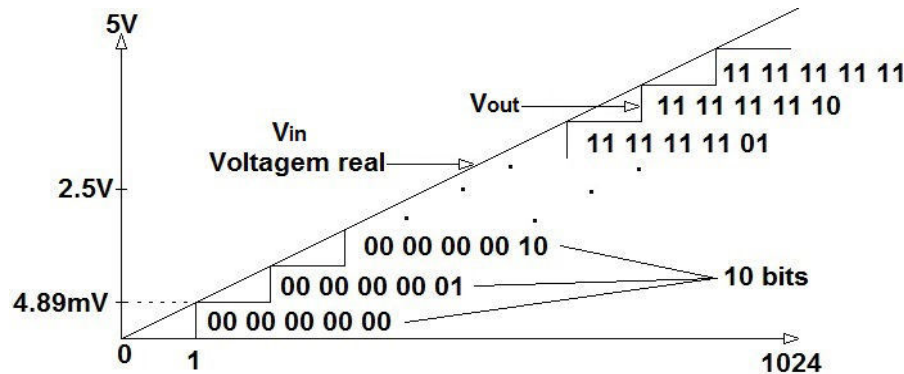


Figura 2.7: Valor de degrau para uma conversão em 10 bits com referência em 5V.

Caso trabalhe com a referência interna de 1,1V, a resolução será:

$$\text{Resolução} = \frac{1.1V}{2^{10}} \approx 1,07\text{mV}. \quad (2.3)$$

Nota-se que ao passo que diminui a tensão de referência, maior será a resolução. Se a referência externa for selecionada, a resolução dependerá do valor de tensão aplicada ao pino AREF.

A placa Arduino UNO possui 6 canais de conversor analógico digital. Essas entradas são nomeadas de A0 a A5 e são exibidas na figura 2.5. Demora cerca de 100 microssegundos (0,0001s) para ler uma entrada analógica, então a taxa máxima de leitura é cerca de 10.000 vezes por segundo.

Ao utilizar o sensor de temperatura LM35 com a referência VCC, pela equação 2.2, temos que o valor de uma divisão é $\approx 4,89\text{mV}$. O comando (`analogRead(Pino);`) afere a tensão da corrente elétrica que transita no sensores conectados em um dos pinos de entradas analógicas.

O valor retornado de (`analogRead(Pino);`) ou (leitura de ADC) está entre 0 a 1023, com 0 representando 0V volts e 1023 intervalos iguais da tensão de funcionamento em uso. Isto é devido à resolução do ADC que converte a tensão de entrada a um número binário de 10 bits. Quando o Arduino toma uma leitura analógica, ele compara a tensão medida no pino analógico sendo usado contra o que é conhecido como a tensão de referência.

Caso o valor na V_{out} estiver de 0V a 4,89mV tem uma representação binária (0000000000),

caso os valores de tensão analógica estejam entre 4,89mV e 9,78mV tem uma outra representação binária de (0000000001) e assim por diante. A leitura analógica da tensão irá variar de cerca de 0V a cerca de 1,75V.

Podemos usar a “regra de três” para calcular a representação binária de uma tensão de entrada analógica. Por exemplo, vamos calcular a representação binária de valor sobre a tensão analógica de 0,25 volts:

$$\left\{ \begin{array}{cc} 5V & 1024 \\ 0,25V & X \end{array} \right\} \rightarrow X = \frac{1024 \times 0,25V}{5V} = 51,2 \approx 51. \quad (2.4)$$

A tensão analógica de 0,25V será representada pelo número decimal 51 ou pelo número binário (0000110100). De maneira semelhante, podemos encontrar uma representação binária para qualquer nível desejado de tensão de entrada nos pinos analógicos. A conversão do valor lido em temperatura em graus Celsius é da seguinte forma:

$$\text{Tensão em A0} = \frac{\text{Valor lido em A0} \times 5V}{1024}, \quad (2.5)$$

$$\text{Temperatura} = \frac{\text{Tensão em A0}}{10mV} = \frac{0,25mV}{10mV} = 25^{\circ}C. \quad (2.6)$$

Logo:

$$\text{Temperatura} = \frac{(\text{Valor lido em A0}) \frac{5V}{1024}}{10mV}. \quad (2.7)$$

Então:

$$\text{Temperatura} = \frac{5 \cdot 51 \cdot 100}{1024} = 24,90 \approx 25^{\circ}C. \quad (2.8)$$

A propósito, o valor de 10mV é intrínseco ao sensor LM35 e corresponde à variação de 1°C, essa informação está na ficha técnica do sensor (o *Datasheet*)¹³.

Shields ou Escudos/Módulos

Os *Shields* são utilizados para se estender as propriedades do Arduino, em tradução livre significa escudos, para essa aplicação tem significado de módulos. Os *Shields* são placas de circuito equipadas com outros dispositivos (por exemplo, receptores GPS, *displays* de LCD, módulos de *Ethernet* etc.), que você pode simplesmente conectar ao seu Arduino para obter funcionalidades adicionais, desde que utilize a biblioteca fornecida pelo fabricantes ou construa seu próprio código.

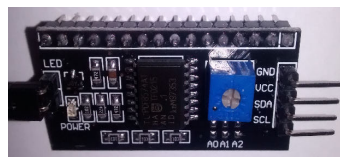
¹³Catálogo ou Folha de dados para circuitos integrados.

Os *Shields* também estendem os pinos até o topo de suas próprias placas de circuito, para que você continue a ter acesso a todos eles. Você não tem de utilizar um *Shield* se não quiser; basta montar todo um circuito que opere a função desejada e adicionar ao projeto, deve ser montado o circuito utilizando uma *Protoboard*, ou criando sua própria PCB.

Podemos encontrar uma lista dos Shields oficiais e não oficiais compatível para usar com Arduino em *shieldlist*¹⁴. É de grande utilidade esse recurso pois muitas vezes em que montamos os projetos é desejável ou necessário; exemplo disso é o uso de um LCD para mostrar alguns dados em tempo real ou para seleção de menus etc, ou quando temos um sistema que consome mais corrente que é fornecido pelo Arduino, então é necessário usar um destes módulos Exemplo é quando vamos usar um motor DC que consome mais de 40mA, nesse caso é recomendado usar um circuito externo como *Shield ponte H* que permite usar uma tensão diferente da que é alimentada o Arduino, também uma maior corrente pode ser consumida e além dessas características é possível inverter a rotação do motor, variar a velocidade de rotação por PWM.



(a) Módulo LCD 16X2.



(b) Módulo de comunicação Serial I2C.



(c) Verso de (a) e a junção com (b).

Figura 2.8: Junção dos módulos: I2C e LCD.

Na imagem 2.8 vemos a junção de dois *Shields*, essa junção não é necessariamente obrigatória, o fato de usarmos um modulo de comunicação serial é que simplifica a conexão de um modulo LCD que normalmente utiliza oito conexões elétricas todas as vezes que for montar o *Display*, e seriam utilizados muitas das portas do Arduino. Nesse caso somente quatro ligações são necessárias, são elas: GND fio preto, VCC → 5V fio vermelho, SDA (pino 4) → RX fio laranja e SCL (pino 5) → TX fio azul, ou seja, dois pinos para tráfegos de dados serial e a alimentação, simplificando a montagem e reduzindo assim o uso de pinos, o esquema das ligações pode ser conferida na figura 2.9, e a montagem na figura 2.8.

¹⁴<http://shieldlist.org/>

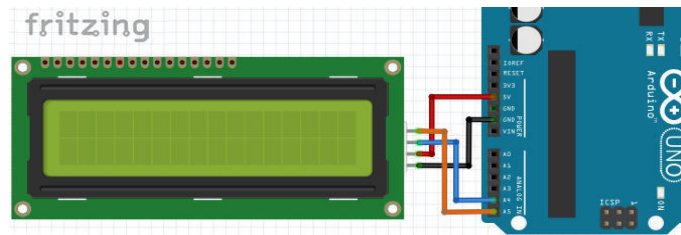


Figura 2.9: Ligação dos respectivos pinos do módulo I2C com o LCD ao Arduino.

Protoboard & Jumpers

Para organizarmos a montagem dos dispositivos e do Arduino vamos utilizar a *Protoboard* e os *Jumpers*. O *Jumper* é uma ligação móvel entre dois pontos de um circuito eletrônico. É, geralmente, uma pequena peça plástica isolante que contém uma peça metálica em seu interior, responsável pela condução de eletricidade, nas extremidades existem plugues que podem ser do tipo macho e do tipo Fêmea. Já a Protoboard é uma plataforma de ensaio ou matriz de contato, é uma placa com furos e conexões condutoras para montagem de circuitos elétricos experimentais. Os furos são dispostos em uma matriz numerada, o que na montagem de circuitos eletrônicos facilita a inserção de componentes, uma vez que não necessita soldagem, os pinos das bordas da placa tem uma disposição diferente dos do interior, eles são marcados com duas linhas, uma vermelha que deve ser usada com o +5V e uma azul para ser o GND, essas linhas são contínuas na horizontal, já as do interior da placa são contínuas de cinco em cinco furos e na vertical, veja na figura 2.10 um modelo.

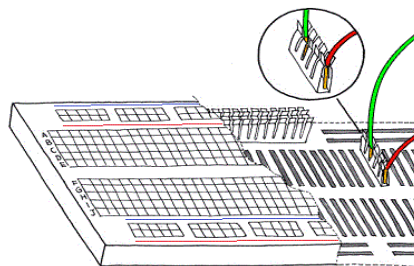


Figura 2.10: Desenho esquemático de uma Protoboard.

[Fonte: <https://roboott.wordpress.com/2014/11/04/protoboards-como-utilizar/>]

Sensores e Transdutores

O sensor é o elemento sensível do sistema, são componentes usados para ler e interpretar variáveis físicas do ambiente, como: intensidade de luz, som, objetos, temperatura etc., podem ser medidos e traduzidos como uma determinada diferença de potencial em seus terminais, a variação do potencial nos terminais são causados pela variação do fenômeno físico, ou seja, variação da energia, a esse fenômeno dá-se o nome de transdução. Todo

sensor possui um transdutor, o sensor detecta a variação física de interesse, como por exemplo, pressão, temperatura, força, etc., e o transdutor converte a energia de variação física em uma outra energia, por exemplo energia elétrica.

Atuadores

Os atuadores são todos os dispositivos responsáveis pela variação de parâmetros do processo a ser controlado. Recebem um sinal proveniente do controlador e agem sobre o sistema controlado executando uma determinada ação física, por exemplo, um motor é um elemento que produz movimento, um LED é um elemento que produz iluminação, etc. Sempre que precisarmos que alguma ação seja realizada como declarada no programa, devemos usar um atuador específico para a ação. Assim se desejamos emitir um sinal sonoro devemos usar um alto falante, e assim por diante.

2.2.2 Programação

O objetivo desta seção é apresentar, de uma forma simples e rápida, o básico de programação para que possamos começar a utilizar o Arduino e em seguida dar prosseguimento ao projeto, sem ter que ler muitos livros ou artigos sobre programação.

Um *sketch*, ou programa, é uma sequência de instruções que são enviadas para o computador. O microprocessador (cérebro) por sua vez, entende um conjunto de instruções, ou seja, a linguagem de programação, o “idioma”. Também chamamos esse “idioma” de linguagem de máquina. As linguagens de máquina são no fundo as únicas linguagens que os computadores conseguem entender, só que elas são muito difíceis para os seres humanos entenderem. É por isso usamos a linguagem de programação.

Nós seres humanos precisamos converter as nossas ideias para uma forma que os computadores consigam processar, ou seja, a linguagem de máquina. Então precisamos de um outro “idioma” especial para instruir o computador a fazer as tarefas que desejamos. Esse “idioma” é uma linguagem de programação, existem muitas delas. Essas linguagens de programação também são chamadas de linguagens de programação de alto nível. A linguagem de programação utilizada no Arduino é a linguagem C++ (com pequenas modificações), que é uma linguagem muito tradicional e conhecida. Essa é a linguagem que utilizaremos ao longo deste trabalho.

Para compreendermos como e de que forma o processador executa as funções estipuladas pelo operador, precisamos antes entender alguns pontos-chaves como: Lógica de Programação, Fluxo de processamento dos dados, as palavras que vamos usar para instruir o microprocessador a executar as instruções, ou seja, a sintaxe da linguagem C/C++ e onde podemos inserir as sintaxes e construir nossas instruções.

Lógica de Programação

A lógica de programação é essencial para trabalhar com desenvolvimento de um *sketch* para o Arduino. E podemos definir como um conjunto de técnicas para encadear pensamentos a fim de atingir determinado objetivo. O objetivo fundamental de toda programação é construir o algoritmo que vai executar exatamente o que queremos.

Nesse sentido usamos a lógica para programar a melhor sequência de atividades que levam à solução de um problema. Até mesmo as soluções para os problemas cotidianos mais simples podem ser descritas por sequências lógicas de atividades. Vamos entender no exemplo a seguir.

Suponha que desejamos trocar uma lâmpada, a sequência de ações para se trocar a lâmpada é a lógica de programação e pode ser feita de várias formas:

Problema: Troca de uma lâmpada.

Sequência de passos para Solução.

1. Ligue o Interruptor;
2. Pegue uma escada;
3. Pegue uma lâmpada nova;
4. Suba na escada;
5. Retire a lâmpada velha;
6. Coloque a lâmpada nova.

Problema: Troca de uma lâmpada.

Sequência de passos para Solução.

1. Pegue uma lâmpada nova;
2. Se a lâmpada não acender;
3. Posicione a escada embaixo da lâmpada;
4. Suba na escada;
5. Retire a lâmpada velha;
6. Coloque a lâmpada nova.

A sequência de passos seguidos para atingir o objetivo (trocar a lâmpada), é nossa lógica. Note que todos os passos devem ser executados a fim de que o objetivo seja alcançado.

Porém, há situações nas quais a execução de alguns passos pode depender de decisões a serem tomadas. Dessa forma, algum fato indicará se um ou mais passos devem ser executados ou não, ou ainda pode ser requerido uma repetição em algum momento caso o objetivo não seja completado, vejamos a seguir.

Problema: Troca de uma lâmpada.

Sequência de passos para Solução.

1. Ligue o Interruptor;
2. Se a lâmpada não acender;
 - 2.1 Pegue uma escada;
 - 2.2 Posicione a escada embaixo da lâmpada;
 - 2.3 Pegue uma lâmpada nova;
 - 2.4 Suba na escada;
 - 2.5 Retire a lâmpada velha;
 - 2.6 Coloque a lâmpada nova;
 - 2.7 Se a lâmpada não acender
 - 2.7.1 Retire a lâmpada;
 - 2.7.2 Coloque uma outra lâmpada
 - 2.7.3 Se a lâmpada ainda não acender;
 - 2.7.3.1 Retire a lâmpada;
 - 2.7.3.2 Coloque uma lâmpada nova,

Assim, neste nova sequência, enquanto a condição definida na linha 2.7 for verdadeira (ou seja, enquanto a lâmpada não acender), as ações definidas em 2.7.1 e 2.7.2 serão repetidas.

Abstraímos os fatos de ter de descer da escada para pegar uma lâmpada nova e subir novamente, porque para o homem é óbvio que algo assim deve ser feito, se você não tem uma lâmpada nova tem que providenciar, tendo que ir buscá-la onde for necessário. No caso de construirmos uma instrução para uma máquina a instrução deve conter todos os passos. O melhor programa é aquele em que a sequência é completada com todas as falhas possíveis e as devidas soluções, a melhor organização desses passos é a lógica utilizada para se alcançar o objetivo.

Fluxograma

Fluxograma é uma representação gráfica de um processo. São utilizados símbolos para descrever passo a passo quanto à natureza e o fluxo desta operação. O objetivo é mostrar de forma descomplicada o fluxo das informações e elementos, além da sequência operacional que caracteriza a operação que está sendo executada. As etapas do fluxograma são apresentadas utilizando-se figuras geométricas que podem ser círculos, triângulos, retângulos, linhas ou setas, sendo que cada símbolo possui um significado importante (ver figura 2.11).

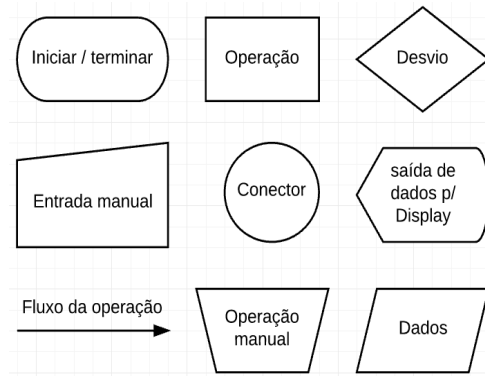


Figura 2.11: Principais símbolos e seus significados.

Para descrevermos as operação do experimento, vamos montar o fluxograma desta operação, vamos utilizar a forma linear (fluxograma Linear), ver figura 2.12. O fluxograma linear é um diagrama que exhibe a sequência de trabalho passo a passo que compõe o processo. Esta ferramenta ajuda a identificar retrabalhos, redundâncias ou etapas desnecessárias.

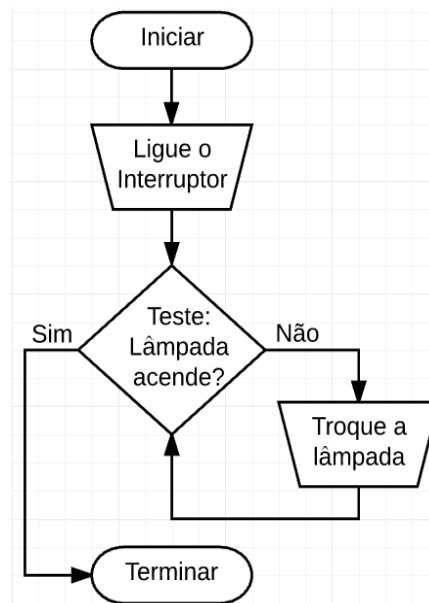


Figura 2.12: Fluxograma simples da troca de uma lâmpada com repetição.

Após a elaboração do fluxograma que descreve o processo, deve-se preocupar em levantar e responder as seguintes questões:

- O processo obtido pode ser redesenhado ou remodelado de forma a tornar-se mais efetivo ou eficiente?
- Existem lacunas para falhas não exploradas no processo?
- O processo é suficientemente eficaz para desempenhar um papel fundamental na operação?

Os benefícios abrangem maior entendimento com relação aos seguintes itens:

- Quais são os principais passos do processo;
- Quem é responsável por uma atividade;
- Quais são os principais momentos de decisão;
- Quais são as entradas e saídas do processo;
- Como flui a informação;
- Quais recursos estão envolvidos no processo.

É importante ressaltar que quando um fluxograma é elaborado visando identificar melhorias de um processo, deve-se sempre ter a preocupação de pensar no processo exatamente como é e não como ele deveria ser. Somente com um cenário realista é possível identificar pontos de melhoria. Ao visualizar todo o processo, poderá evitar complexidades desnecessárias, identificar erros ou duplicidade de procedimentos. Os fluxogramas simplificam e racionalizam o trabalho, facilitando a compreensão, otimização e melhorias do programa.

Sintaxe de Programação

O texto contendo o programa em uma linguagem de programação de alto nível também é conhecido como o código fonte do programa, esse código é formado por uma arranjo de palavras chaves, as sintaxes da linguagem utilizada (C/C++), e para toda linguagem de programação as sintaxes podem ser divididas em três partes principais: estrutura, valores (variáveis e constantes), e funções. E programar é o ato de combinar essas sintaxes na melhor forma. A estrutura de um programa deve conter essas três partes de forma harmoniosa.

Os algoritmos construídos na IDE do Arduino são compostos de duas estruturas que é o (*Void setup()*{ *Aqui vai o código que se repete uma única vez assim que se inicia.*}). O processador armazena os valores definidos e usa posteriormente quando necessário, mas esses valores ou definições não são alterados. E a segunda estrutura base é a (*Void loop()*{ *Aqui vai o código que se repete enquanto estiver energizado*}). Aqui os valores são recalculados continuamente, como exemplo podemos citar a leitura de temperatura de um sensor LN35, o microcontrolador vai fazer cerca de dez mil leituras em um único segundo, e claro que esse tempo de leitura pode ser programado com a função (*delay(Aqui vai o tempo desejado em milésimo de segundo)*;). Antes dessas duas estruturas base podemos também predefinir alguns parâmetros, como bibliotecas a utilizar, pinagem a usar, constantes etc. (ver figura 2.14).

Uma variável é um espaço na memória do microcontrolador para armazenar um determinado dado e é composta por (tipo, nome e valor), exemplo: (*int redPin = 9;*), onde nome \rightarrow redPin, o tipo \rightarrow int e o valor \rightarrow 9. O tipo nos diz que a variável armazena os números inteiros, assim se desejamos armazenar caracteres ou outro tipo de valores deve-se usar a variável correspondente.

As funções são os comandos que têm uma função específica como mencionado anteriormente. A função *delay* contabiliza o tempo entre um processo e o próximo, outros exemplos são:

- (*pinMode(pino, INPUT ou OUTPUT);*) \rightarrow configura o pino digital especificado que pode se comportar tanto como uma entrada ou uma saída;
- (*digitalWrite(pino, HIGH ou LOW);*) \rightarrow escrever um valor a um pino digital se ALTA ou BAIXA;
- (*digitalRead(pino de entrada);*) \rightarrow lê o valor de um pino digital especificado, se está em ALTO ou BAIXO.

As funções para os pinos analógicos mais usuais são:

- (*analogRead(pino);*) \rightarrow lê o valor da tensão no pino analógico especificado;
- (*analogWrite(pin, valor entre 0 a 255);*) \rightarrow escreve um valor analógico a um pino digital por modulação de pulso (onda PWM), o valor do ciclo de trabalho é entre 0 (sempre desligado) e 255 (sempre ligado);
- (*map(variável que sera remapeada, valo inicial velho, valor final velho, valor inicial novo, valor final novo);*) \rightarrow re-mapeia um número a partir de um intervalo para outro.

As sintaxes das estruturas são aquelas que não têm uma função específica ou são um operador, a função de uma estrutura pode ser usada de várias maneiras, podemos fazer comparações com uma determinada variável e escolher uma ação, isso é o que a estrutura de controle *if* ou (se (condição 1 - operador condicional - condição 2){fazer alguma coisa}) é estruturado da seguinte forma (*if (x <= 0){fazer alguma coisa}*), é utilizada para desvio de fluxo quando um determinado parâmetro é ou não satisfeita. Já a estrutura (*setup(){digite seu Código aqui.}*) é chamada quando um esboço começa, o código nessa estrutura só é lido uma vez quando inicia-se a estrutura, já a estrutura (*loop (){digite seu Código aqui.}*) faz exatamente o que seu nome sugere, leitura do código consecutivamente, permitindo que rode por tempo indeterminado, alterando as variáveis e executando as ações requeridas. Os operadores (aritméticos, de comparação, booleanos,

compostos etc.) também são considerados estruturas. Na tabela a seguir 2.2 podemos ver os operadores e suas características.

Tabela 2.2: Operadores

Operadores aritméticos	Operadores de comparação	Operadores booleanos
= Atribuição	== Igual a	&& E
+ Adição	!= Diferente de	Ou
- Subtração	< Menor que	! Não
* Multiplicação	> Maior que	Operadores compostos
/ Divisão	<= Menor ou igual a	++ Incremento
% Modulo	>= Maior ou igual a	- Decremento

Uma estrutura muito útil na hora de construir um *sketch* é incluir comentários nas principais linhas do código. Quando usamos (`// Digite seu comentário aqui.`) para uma linha ou (`/* Digite seu comentário aqui. */`) para múltiplas linhas, Eles são ignorados pelo compilador e não são exportados para o processador para que eles não ocupem qualquer espaço do microcontrolador.

Nessa seção demos somente uma breve explicação das sintaxes que iremos utilizar para a construção e controle do experimento, basicamente as sintaxes mencionadas aqui são as utilizadas no *sketch* deste trabalho, para um análise mais profunda, o site oficial disponibiliza uma página¹⁵ contendo todas as estruturas, variáveis e funções necessárias para a montagem de outros *sketchs*. Também é possível ver modelos e a explicação de como utilizar cada uma das sintaxes da linguagem utilizada para programar o Arduino.

Ambiente de Desenvolvimento

Para converter um programa escrito em uma linguagem de alto nível para linguagem de máquina, nós utilizamos um compilador. A ação de converter um *sketch* para linguagem de máquina é chamada compilar. Para compilar o *sketch*, normalmente se utiliza um ambiente de desenvolvimento (ou IDE, do inglês *Integrated Development Environment*), que é um aplicativo de computador que possui um compilador integrado, onde você pode escrever o seu *sketch* e compilá-lo. No caso do Arduino, esse ambiente de desenvolvimento é o Arduino IDE.

¹⁵<https://www.arduino.cc/en/Reference/HomePage>

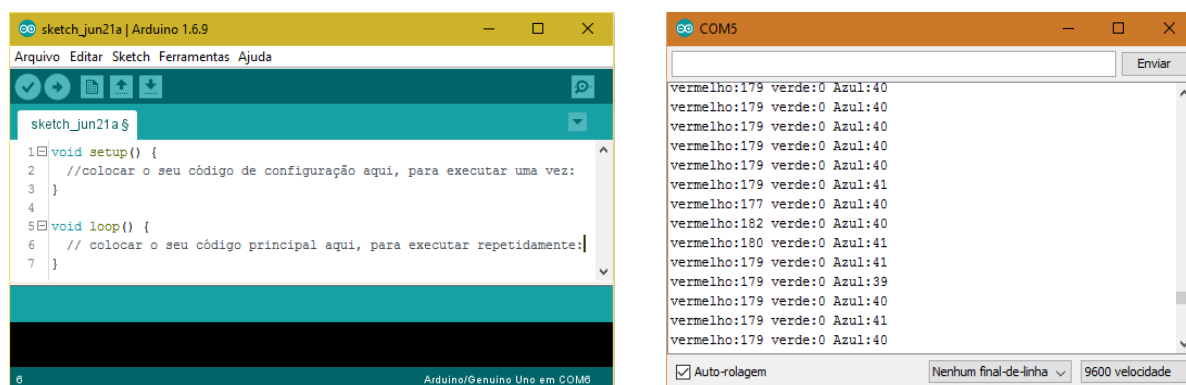


Figura 2.13: Caracterização do ambiente de desenvolvimento do Arduino (IDE).

Como podemos ver, o aplicativo tem uma interface limpa, com poucas opções de menus, mas é completo e ideal para se fazer o *sketch*, basicamente o aplicativo é dividido em três partes: *i*) Barra de Ferramentas; *ii*) Ambiente de Programação; e *iii*) Área de mensagens.

A barra de ferramentas contém menus e os seis botões que compõem os comandos do Arduino. É possível fazer basicamente todo tipo de configuração nos menus que são compostos por submenus onde podemos alterar o idioma, escolher qual placa Arduino estamos usando, em qual porta USB nosso arduino está plugado, podemos também abrir exemplos de *sketches* ou abrir um já salvo pelo usuário, etc.

A funcionalidade de cada botão da barra de ferramentas permite que possamos **Verificar** o código sem compilar, já o botão **Compilar** carrega os *sketches* no Arduino, o botão **Criar** abre uma nova janela do IDE, o botão **Abrir sketches** antigos salvos pelo usuário, e o botão **Salvar** salva os *sketches* novos e o botão **Monitor Serial** abre um bloco de notas onde é possível ver os dados processados no Arduino em tempo real direto na IDE (ver figura 2.13b), esses botões estão dispostos respectivamente da esquerda para a direita (ver figura 2.13a). Para maior esclarecimento o site o oficial disponibiliza as informações das possibilidades da barra de ferramentas, basta seguir o link¹⁶. Preferências

No centro da janela do IDE temos o espaço onde inserimos as sintaxes dos *sketches*, o aplicativo separa os comandos por cores e por ordem de processamento, no caso de termos um comando dentro de outro, os comandos mais internos são alocado mais para a direita. É possível ainda enumerar as linhas do código, basta irmos na barra de ferramentas em **Arquivo**, depois em **Preferências** e selecionarmos a caixa que diz **Mostrar número de linhas**.

Ao final da janela do aplicativo temos a terceira parte, uma área de mensagem que

¹⁶<https://www.arduino.cc/en/Guide/Environment>

nos dá um *feedback* ao **verificar** e/ou **compilar**, se ocorrer algum erro é exibido nessa parte.

Para fazer a instalação do aplicativo no computador, basta baixar do site oficial do Arduino o arquivo instalador correspondente ao sistema operacional; para seguir um tutorial completo e oficial de instalação dos sistemas operacionais Linux¹⁷, Windows¹⁸ ou Mac¹⁹, basta seguir os links no rodapé referente ao sistema operacional desejado.

Bibliotecas

Quando se cria um código/programa usando uma linguagem de programação, neste caso C/C++, existe a possibilidade de usar um conjunto de funções pré-criadas por outros programadores que já resolvem determinados problemas, poupando imenso trabalho. A esse conjunto de funções damos o nome de bibliotecas, do inglês, *library*.

Um exemplo de biblioteca que iremos utilizar ao longo do experimentos é: LiquidCrystal. Esta é utilizada quando temos de usar o *display* LCD 16x2 acoplado ao Arduino para saída de dados em tempo real. Assim, de uma forma intuitiva, consegue-se enviar caracteres para a tela sem precisar fazer uma programação rebuscada de uma matriz de pixel, apenas precisamos usar as funções necessárias para enviar os dados para a tela, como (`lcd.print(Coloque seu Comando ou Palavra aqui);`) esse comando escreve na tela do LCD a palavra contendo a variável ou o texto desejado.

Na prática, para se utilizar uma biblioteca basta incluí-la no início do código. Por exemplo, para usar a biblioteca LiquidCrystal; deve ter uma linha no início do código escrita com o seguinte formato (`#include <LiquidCrystal.h >`) sem os parênteses.

Algoritmos

O algoritmo, ou simplesmente as linhas de código de um programa, é uma forma de dizer para um microcontrolador o que ele deve fazer, de uma forma que nós humanos conseguimos entender facilmente. Os algoritmos normalmente implementados em linguagens de programação de alto nível. Isso se aplica a praticamente qualquer microcontrolador, inclusive o Arduino, onde um algoritmo também é conhecido como *sketch*. Para simplificar, o algoritmo é composto de uma sequência de comandos, normalmente escritos em um arquivo de texto (ver figura 2.14), que contém as devidas instruções que serão processadas no microcontrolador. Esse texto é redigido na IDE do Arduino e as palavras que utilizaremos são as sintaxes contidas na linguagem C/C++. A partir de agora nós vamos nos referir aos algoritmos, programas ou *sketches* simplesmente como “*sketch*”.

¹⁷<https://www.arduino.cc/en/Guide/Linux>

¹⁸<https://www.arduino.cc/en/Guide/Windows>

¹⁹<https://www.arduino.cc/en/Guide/MacOSX>

```
1 int ledPin = 13; // LED conectado ao pino digital 13.
2 int inPin = 7; // Botão de pressão ligado ao pino digital 7.
3 int val = 0; // Variável para armazenar o valor lido.
4
5 void setup()
6 {
7   pinMode(ledPin, OUTPUT); // Define o pino digital 13 como saída.
8   pinMode(inPin, INPUT); // Define o pino digital 7 como entrada.
9 }
10
11 void loop()
12 {
13   val = digitalRead(inPin); // Lê o pino de entrada.
14   digitalWrite(ledPin, val); // Define o LED para valor do botão.
15 }
```

Figura 2.14: Texto de um programa redigido na IDE do Arduino.

2.3 Simulações como Objetos de Aprendizagem

Como já mencionamos na introdução, a Física é considerada por muitos dos alunos como uma disciplina de difícil compreensão, o que gera desconforto e logo perdem o interesse e desistem de estudá-la. Um dos motivos para esse desinteresse dos alunos é o fato de a escola manter-se alheia aos avanços tecnológicos que a sociedade tem passado nas últimas décadas. Mesmo com a evolução dos meios de comunicação e a popularização dos *smartphones*, computadores e da internet, a maior parte dos professores continua a lecionar apenas com quadro e seus utensílios, como era transmitido quando não existia esses recursos atuais.

Existe a possibilidade de se adotar os experimentos baseados no contexto tecnológico atual para facilitar a visualização dos fenômenos físicos, o que é muito importante e eficiente, pois chama a atenção dos alunos e mostra que a Física é muito mais do que cálculos matemáticos, que mesmo que saibam manipular muitos das tecnologias atuais eles não compreendem seu funcionamento ou qual ciência está envolvida. Se utilizarmos a informática atual para auxiliar na transmissão de conteúdos, podemos gerar uma aprendizagem significativa e muito eficiente para facilitar a compreensão dos fenômenos pelos alunos, esse é o campo onde os simuladores têm um importante papel.

Os simuladores, como o próprio nome já diz, permitem que façamos simulações de situações experimentais. Um exemplo de site muito famoso que contém vários desses aplicativos é o PHET, da Universidade do Colorado, que apresenta materiais para Física, Matemática, Química e Biologia, mas existem outras fontes de simuladores muito poderosos, um dos acervos mais ricos que podemos ter acesso é os aplicativos para *smartphones*, um simples aplicativo de afinação de violão pode ser utilizado de diversas formas para se ensinar Física.

Fiolhais e Trindade (2003), em seu trabalho sobre o uso de computadores como fer-

ramenta no ensino de física, reconhecem as dificuldades que os estudantes têm com as disciplinas da área das ciências exatas e posicionam-se a favor do uso de computadores como medida atual e significativa para tentar mudar a visão dos alunos de ensino médio em relação à física:

“A necessidade de diversificar métodos para combater o insucesso escolar, que é particularmente nítido nas ciências exatas, conduziu ao uso crescente e diversificado do computador no ensino da Física. O computador oferece atualmente várias possibilidades para ajudar a resolver os problemas de insucesso das ciências em geral e da Física em particular”. (FIOLHAIS; TRINDADE, 2003, p.259).

O uso de simuladores apresenta algumas vantagens, como interatividade e participação mais efetiva dos alunos, baixo custo, pouco tempo de duração e os alunos podem trabalhar um experimento com muitas variáveis.

Além disso, esses softwares também podem ser usados pelos alunos no conforto de suas casas. Essa metodologia cada vez mais é aceita e procurada pelos estudantes. Durante a minha graduação sempre que me deparava com um determinado conteúdo que não consegui extrair a essência, uma forma de sanar essa dificuldade era a busca de uma simulação que na maioria das vezes era suficiente.

2.3.1 PhET

PhET é uma plataforma de simulações interativas que foi fundada em 2002 pelo ganhador do Prêmio Nobel Carl Wieman. O projeto PhET, Simulações Interativas da Universidade de Colorado Boulder, cria simulações gratuitas. As Simulações PhET são baseadas em educação, extensiva pesquisa e envolvimento dos alunos através de um ambiente intuitivo, *game-like*, onde os alunos aprendem através da exploração e descoberta.

No site do PhET podemos encontrar simulações divertidas de Física, Química, Biologia e Matemática. Todas as simulações passam por testes e avaliações rigorosas para assegurar a eficácia educacional. Estes testes incluem entrevistas de estudantes e observação do uso de simulação em salas de aula. As simulações são escritas em Java, Flash ou HTML5, e podem ser executadas on-line ou baixadas para o computador. Todas as simulações são de código aberto (REA). Essa iniciativa conta com vários patrocinadores que apoiam o projeto PhET, permitindo que estes recursos sejam livres para todos os estudantes e professores. O sucesso da plataforma de simulação PhET foi agraciada com diversos prêmios durante esse anos de atividades, são eles:

- SIGOL *Online Learning Award*, 2º lugar (Abril de 2012);
- *Tecnologia Award* e o Prêmio Educação Microsoft (Outubro de 2011);

- Concessão de *NSF & Science* Revista Internacional de Ciência e Engenharia Visual Challenge (2007);
- Prêmio MERLOT *Classics* em Física (2006);
- Prêmio de MERLOT *editor Choice* (2006) (PhET em MERLOT).

As simulações da plataforma PhET contam com diversos princípios de design para ajudar o envolvimento dos alunos em ciências e matemática através de inquérito, são eles:

- Incentivar a investigação científica;
- Tornar visível o invisível;
- Usar conexões com o mundo real;
- Dar aos usuários a orientação implícita (por exemplo, através de controles de limite) na exploração produtiva;
- Criar uma simulação que possa ser flexivelmente usada em muitas situações educacionais.

Conta também com várias ferramentas nas quais as simulações fornecem uma experiência interativa:

- Clicar e arrastar para interagir com recursos da simulação;
- Usar controles deslizantes para aumentar e diminuir os parâmetros;
- Escolher entre as opções como os botões de rádio;
- Fazer medições em seus experimentos com vários instrumentos – réguas, cronômetros, voltímetros e termômetros.

Isso permite que os professores e alunos explorem as relações de causa e efeito e respondam a perguntas científicas através da exploração da simulação.

2.3.2 Alcance das Simulações

As simulações não representam a totalidade de eventos ocorridos, podem mascarar a objetividade, isso ocorre quando tratamos com uma simulação que incorpore todas as propriedades do mundo real. As simulações são projetadas em modelos simplificados, deixando de considerar aspectos de grande influência no ato da experiência real. Logo, a

simulação computacional deve ser usada pelo professor como ferramenta de ensino, e deve-se assegurar que os aprendizes saibam que a simulação de forma virtual de um fenômeno não supre todas as propriedades de uma experiência no mundo real (MEDEIROS, 2002).

Visualizar um fenômeno usando simulações computacionais é sem dúvidas superior às imagens estáticas de livros ou diagramas feitos no quadro de sala de aula, mas deve-se estabelecer o pensamento de que a simulação é uma representação simplificada do real. Essa orientação é dada por Medeiros, quando afirma que:

“Uma animação não é, jamais, uma cópia fiel do real. Toda animação, toda simulação está baseada em uma modelagem do real. Se essa modelagem não estiver clara para professor e educando, se os limites de validade do modelo não forem tornados explícitos, os danos potenciais que podem ser causados por tais simulações são enormes”.(MEDEIROS, 2002, p.81).

O professor deve orientar o estudante a construir, mesmo de forma simplificada, o modelo físico que ilustra um fenômeno ou situação, que permitirá ao aluno compreender o raciocínio e as relações que devem ser feitas para compreender o fenômeno (FIOLHAIS; TRINDADE, 2003).

Capítulo 3

Materiais e Método

3.1 Materiais

Para realizar o experimento proposto é preciso dos seguintes materiais de base: um microcomputador, placa Arduino UNO, *protoboard* de 400 furos, kit de cabos *Jumpers* macho/macho e macho/fêmea, utilizados para as devidas conexões entre os terminais elétricos, dois *shields*, sendo um módulo LCD e o outro um módulo serial I2C.

Um ambiente bem iluminado é indispensável, bem como de uma superfície plana para dispor seus componentes. Lembre-se de que vamos trabalhar com eletricidade (ainda que uma corrente contínua de baixa voltagem); portanto, uma superfície de metal deverá ser coberta por um material não condutivo, como uma toalha, madeira ou papel, antes que você posicione seus materiais. Essa superfície deve estar próxima de seu PC ou laptop para permitir a conexão USB e o carregamento do código para o Arduino, após passar o código para o Arduino é opcional o uso de um PC.

Da mesma forma, será interessante ter, ainda que não seja essencial, algumas ferramentas necessárias dependendo de suas montagens, caso opte por fazer seus próprios *jumpers*, vai precisar de fios, alicate de corte, um ferro de solda, que pode ser utilizado no reaproveitamento de componentes eletrônicos dos aparelhos velhos. Serão necessários, caso deseje aproveitar componentes eletrônicos de sucatas, como sensores e atuadores, chave de *Fenda/Phillips* conforme a necessidade, um adaptador de 7V a 12V para alimentar o Arduino sem o uso do PC.

Na tabela 3.1 vemos a relação dos materiais de que utilizei e seus respectivos valores¹. O Arduino utilizado foi o Nano, uma versão totalmente compatível com o UNO e mais barato.

¹Os produtos foram comprados na loja virtual <http://www.mercadolivre.com.br/>.

Tabela 3.1: Material de base

Itens	Valor R\$
Arduino Nano	14,80
Modulo LCD	12,80
Modulo Serial I2C	13,50
Protoboard 400 furos	14,90
Frete	17,60
Total	73,60

3.2 Metodologia

O Guia Pedagógico foi elaborado com o intuito de proporcionar uma aprendizagem significativa e tem como objetivo dar suporte ao professor durante a aplicação da proposta didática. No guia estão descritos o objetivo e a forma de usar o material, a montagem e configuração do experimento, e a simulação correspondente. O Guia de Atividade é composto de um questionário com a metodologia PIE que é baseado em uma situação modelo.

As atividades propostas procuram levar o aluno a refletir sobre o conteúdo, ultrapassando o pensamento empirista simplista da resposta única e correta, fazendo da atividade investigativa um instrumento de aprendizagem.

Os guias são caracterizados pela presença de uma situação problema que favorece os alunos a explorar e testar ideias, sendo-lhes dada a oportunidade para discutirem em grupo e compararem pontos de vista, exercitando o pensamento crítico. Para alcançar tais objetivos elaborou-se questões cujas respostas poderiam ser comparadas com os resultados obtidos após a realização das atividades práticas, o que caracteriza a metodologia PIE (Predizer, Interagir e Explicar) discutida em Dorneles (2010, p. 101).

Nesse exemplo citado observa-se que o vínculo entre a teoria e a prática é reforçado pelo uso da metodologia PIE nas atividades, o que torna o material instrucional potencialmente significativo. O professor poderá ter acesso ao *sketch* consultando o Apêndice A deste trabalho.

Capítulo 4

Roteiro Experimental

Neste Capítulo apresentamos o produto educacional desenvolvido para as aulas experimentais, que envolve o conceito de óptica, mas poderíamos trabalhar qualquer outro conteúdo da física. Para a introdução do experimento na aula foi elaborado um Guia Pedagógico que orienta o professor na atividade experimental e um Guia de Atividade que avalia o conhecimento prévio e posterior á prática.

Para que o usuário possa utilizar de forma adequada este material e usufruir de toda a sua potencialidade, descrevemos separadamente cada um dos passos. Desta forma, acreditamos que o professor poderá compreender em detalhes a montagem de todo o equipamento e sua aplicação no ensino de Física.

O produto educacional desenvolvido para a consolidação deste trabalho de conclusão de curso é composto por:

- Guia Pedagógico desenvolvido para auxiliar os professores durante a aplicação das atividades;
- Guia de Atividades com atividades experimentais dirigidas aos alunos;
- Sketch desenvolvido na linguagem C/C++ para serem executados na plataforma Arduino, utilizado no gerenciamento do experimento.

Todas as linhas do *sketch* deste trabalho são comentadas, o que facilita a compreensão.

4.1 Óptica

4.1.1 Guia Pedagógico - Formação das Cores

Este guia Pedagógico apresenta a atividade prática envolvendo o conceito de formação das cores, através do controle de um LED RGB gerenciado pelo Arduino. O experimento

relaciona a formação de imagens e cores em um visor LCD. O professor deve realizar antes do experimento, o Guia de atividade composto por questionário sugerido e montado com base na metodologia PIE (Predizer, Interagir e Explicar). Deve realizar, também, a simulação, e transmitir o conteúdo correspondente. O questionário tem a função de gerar uma auto avaliação no conhecimento do aluno antes e depois da transmissão do conteúdo, da simulação e da aula experimental.

4.1.2 Conceitos teóricos

Sabemos que a luz ou luz visível como é fisicamente caracterizada, é uma forma de energia radiante. É o agente físico que, atuando nos órgãos visuais, produz a sensação da visão. Energia radiante é aquela que se propaga na forma de ondas eletromagnéticas, dentre as quais se pode destacar as ondas de rádio, TV, microondas, raios X, raios gama, radar, raios infravermelho, radiação ultravioleta e luz visível. Uma das características das ondas eletromagnéticas é a sua velocidade de propagação, que no vácuo tem o valor de aproximadamente 300 mil quilômetros por segundo, ou seja, $c \approx 3 \cdot 10^8 \text{m/s}$.

A luz que percebemos tem como característica sua frequência que vai da faixa de $4 \cdot 10^{14} \text{HZ}$ (vermelho) até $8 \cdot 10^{14} \text{HZ}$ (violeta). Esta faixa é a de maior emissão do Sol, por isso os órgãos visuais de todos os seres vivos estão adaptados a ela, e não podem ver além desta, como por exemplo, a radiação ultravioleta e infravermelha. A cor é a sensação que nossos olhos correspondem a frequência emitida pela fonte de luz. que pode ser fontes primárias ou secundárias. As fontes primárias são chamadas de corpos luminosos, são corpos que emitem luz própria, como por exemplo, o Sol, as estrelas, a chama de uma vela, uma lâmpada acesa, etc. Já as fontes secundárias, também chamadas de corpos iluminados, são os corpos que enviam a luz que recebem de outras fontes, como por exemplo, a Lua, os planetas, as nuvens, os objetos visíveis que não têm luz própria, etc.

Em 1666, o cientista inglês Isaac Newton verificou que a luz branca proveniente do Sol é, na realidade, composta por luzes de várias cores. Isso pode ser percebido quando a luz branca passa por um prisma de vidro. Nessas condições ocorre a decomposição da luz branca nas várias cores que formam o arco-íris.

De acordo com sua cor a luz pode ser classificada como Monocromática ou Policromática. Chama-se luz monocromática aquela composta de apenas uma cor, como por exemplo, a luz amarela emitida por lâmpadas de sódio. Chama-se luz policromática aquela composta por uma combinação de duas ou mais cores monocromáticas, como por exemplo, a luz branca emitida pelo sol ou por lâmpadas comuns.

Ao nosso redor é possível distinguir várias cores, mesmo quando estamos sob a luz do Sol, que é branca. Esse fenômeno acontece pois quando é incidida luz branca sobre um

corpo de cor verde, por exemplo, este absorve todas as outras cores do espectro visível, refletido de forma difusa apenas o verde, o que torna possível distinguir sua cor. Por isso, um corpo de cor branca é aquele que reflete todas as cores, sem absorver nenhuma, enquanto um corpo de cor preta absorve todas as cores sobre ele incididas, sem refletir nenhuma, o que causa aquecimento.

Embora popularmente se diga que o arco-íris tem sete cores – vermelho, alaranjado, amarelo, verde, azul, anil e violeta – na realidade ele tem inúmeras cores distintas, que incluem muitos tons de vermelho, de alaranjado, de amarelo, de verde, de azul (incluindo o que costumava ser chamado de anil) e de violeta. Quando todas essas cores atingem simultaneamente o olho humano, elas provocam a sensação visual da luz branca.

Evidências experimentais mostraram que para que o olho humano tenha a sensação de branco, não é necessário que todas as cores do arco-íris o atinjam. Se luzes de cores vermelha, azul e verde atingirem simultaneamente nossos olhos, isso já será suficiente para causar a sensação visual de luz branca.

As telas dos monitores apresentam características semelhantes mas baseadas nas cores primária, secundária e terciária derivadas de um sistema aditivo de cores, o sistema RGB (Vermelho, verde e azul), existem outros sistemas de cores como o sistema subtrativo (CMYK) utilizado nas impressoras jato de tinta. Cada pixel é formado por três elementos emissores de luz e cada um emite uma cor primária deste sistema de cores, e por estarem muito próximas em relação aos nossos olhos elas se misturam entre si formando uma enorme quantidade de tonalidades em cada pixel.

Dessa forma, o objetivo é colocar os alunos em contato com atividades que proporcionam uma aprendizagem mais concreta a respeito da formação de cores e de imagens.

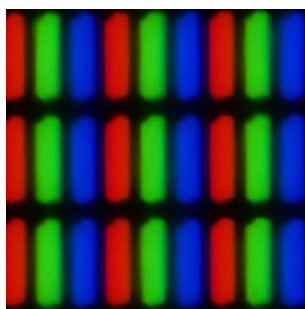


Figura 4.1: Sistema aditivo RGB em monitor LCD, cada pixel é uma sequência das três cores base do sistema.

[Fonte: https://en.wikipedia.org/wiki/Pixel#/media/File:Pixel_geometry_01_Pengo.jpg]

Essa é uma atividade experimental de baixo custo e que pode ser realizada por grupos de alunos. É interessante que o professor crie uma situação-problema para que o próprio aluno faça as combinações de luz a fim de chegar à cor pedida no problema.

O olho humano possui três tipos de receptores, chamados cones. As cores não são

uma propriedade da luz, mas um conceito biológico, baseado na resposta fisiológica do olho humano à luz. (TIPLER, 2000, p.342). O sistema aditivo é aquele formado pelas três cores primárias da luz (Azul, vermelho e verde) as cores primárias do sistema aditivo RGB foram estipuladas por motivos comerciais.

Cor primária é uma cor que não pode ser decomposta em outras cores, ou seja, só é formada por um único comprimento de onda. E nunca é obtida a partir da mistura de outras cores ou de outros comprimentos de onda, sendo impossível ser produzido ou encontrado na natureza, então foi definido onde as três cores do sistema aditivo tem seus maiores picos de intensidade e definidos como cores primárias a faixa de comprimentos de onda circunvizinhas aos respectivos picos. Quando duas cores primárias são mescladas, em partes iguais, duas a duas, elas geram outras três cores: ciano, magenta e amarelo, chamadas de cores secundárias. Existem também as cores terciárias que são o resultado de misturas de uma cor secundária com a outra cor primária.

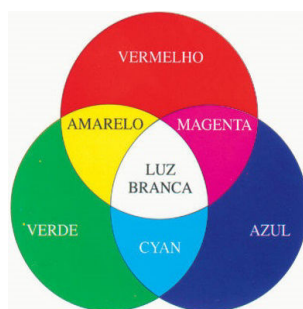


Figura 4.2: Síntese do sistema aditivo.

[Fonte: <http://www.invivo.fiocruz.br/media/rgbcmky3.gif>]

No sistema RGB a formação de outras cores faz-se pela combinação do vermelho, verde e azul, respeitando uma representação numérica. A mais usual é uma escala de valores entre 0 e 255. A escala mínima (zero) é a tonalidade mais escura e acontece quando os três canais estão na escala 0; a escala máxima, ou seja quando os canais têm todos o valor 255, resulta no branco. A tabela 4.1, lista algumas das cores com diferentes Tonalidades:

Tabela 4.1: Intensidades das cores formadas no sistema aditivo de cores (R,G,B)

Cor	Intensidades	Classificação
Preto	(0, 0, 0)	Ausência de cor
Vermelho	(255, 0, 0)	Primária
Verde	(0, 255, 0)	Primária
Azul	(0, 0, 255)	Primária
Amarelo	(255, 255, 0)	Secundária
Magenta	(255, 0, 255)	Secundária
Cyan	(0, 255, 255)	Secundária
Branco	(255, 255, 255)	Somatórias de todas as cores

O físico inglês Isaac Newton demonstrou, utilizando um prisma de três faces, que a luz branca podia ser decomposta em sete cores: violeta, azul, ciano, verde, amarelo, laranja, e vermelho. Na natureza, essa decomposição origina o arco-íris. Ao incidir nas gotas de água da chuva, os raios da luz solar que atravessam sob as nuvens se decompõem nas várias cores.

A cor é relacionada com esses diferentes comprimentos de onda do espectro eletromagnético, do qual apenas uma parte, o espectro visível, pode ser percebida como uma sensação, pelas pessoas e por alguns animais, por meio dos órgãos de visão. Essa faixa varia entre 400 e 700 nanômetros. (TIPLER, 2000 p.342).

Segundo CEPSRM¹, a cor de um objeto é determinado pela média da frequência dos pacotes de onda refletidos por suas moléculas, quando a luz branca incide nesse objeto. É a diferença entre as respostas dos três tipos de cones que permite interpretar diferentes comprimentos de onda como correspondendo a cores diferentes. Esta interpretação é conhecida como teoria dos três estímulos. Esta teoria postula que qualquer cor (comprimento de onda) do espectro visível pode ser reproduzida através da adição dos resultados obtidos pelo estímulo dos três tipos de cones de forma diferentes. A cor percebida depende unicamente da relação entre os três estímulos. Portanto, um corpo terá determinada cor se não absorver os raios correspondentes à frequência daquela cor.

A tonalidade ou matiz é uma medida de claro e escuro, é o que caracteriza a intensidade de uma cor. O tom define o quanto de luz chega realmente aos nossos olhos, é a medida do comprimento de onda médio da luz que ele reflete ou emite e define a cor do objeto. Atinge valores de 0 a 100%.

Enquanto as outras duas características da cor (saturação e intensidade) referem-se à natureza da luz refletida pelo objeto, a saturação também chamado de “pureza”. Quanto menor esse valor, mais com tom de cinza aparecerá a imagem. Quanto maior o valor, mais “pura” é a imagem. Expressa o intervalo de comprimentos de onda ao redor do comprimento de onda médio no qual a energia é refletida ou transmitida. Atinge valores de 0 a 100% (ver figura 4.3).

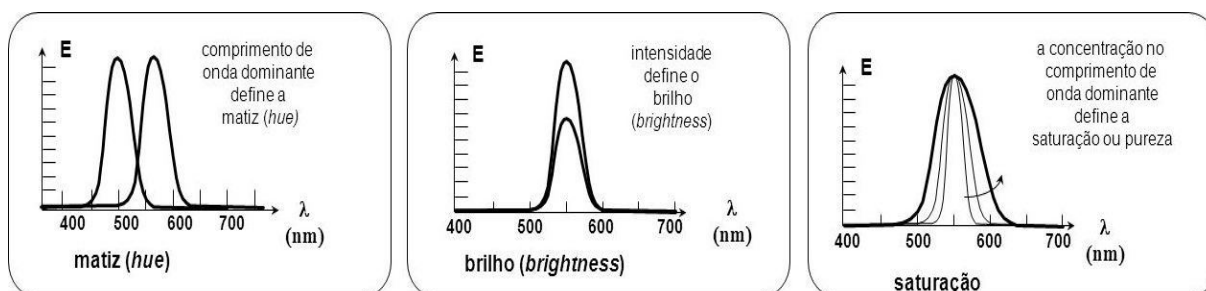


Figura 4.3: Características das fontes luminosas.

[Fonte: <http://images.slideplayer.com.br/3/391798/slides/slide-7.jpg>]

¹Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia.

A Energia (E) define o brilho da cor (ver figura 4.3). Atinge valores de 0 a 100% é comum utilizarmos um *byte* para cada componente, um *byte* permite $2^8 = 256$ possibilidades, assumindo valores de 0 a 255. O que nos dá a faixa de 256 possíveis valores para cada uma das cores. Com este sistema, mais de 16 milhões ($16.777.216$ ou 256^3) diferentes combinações de tons, saturação e brilho podem ser especificados, mesmo que não sejam facilmente distinguidos.

4.1.3 Simulação

Os conceitos teóricos podem ser trabalhados de forma virtual nas aulas ou como atividades que precede o experimento que podem ser feitos em casa previamente ou durante a exposição do conteúdo.

A página² do PhET disponibiliza uma simulação sobre a visão das cores, podemos usar uma fonte de luz branca (policromática) ou com um comprimento de onda definido pelo usuário (monocromática), bem como os filtros limitadores. A simulação é de fácil manejo, e possui uma variedade de opções. Segue exemplo do uso desta simulação.

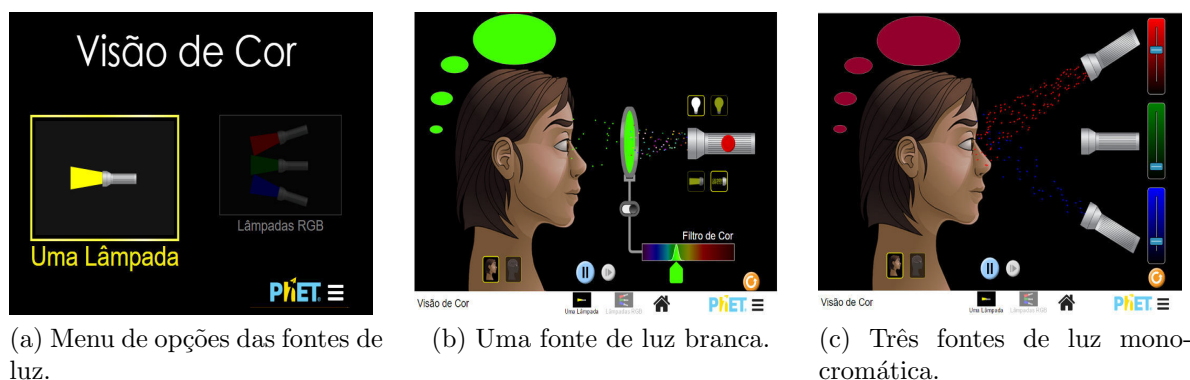


Figura 4.4: Formação das cores, fontes de luz e o uso de filtros.

Os comandos podem variar tanto a intensidade como a frequência, é possível também usar uma representação contínua ou a representação de fótons. São inúmeras as possibilidades que podem ser analisadas, diversos fenômenos podem ser trabalhados virtualmente e posteriormente comprovados com o experimento real. Dentre os fenômenos que podem ser simulados estão: absorvância e transmitância da luz, frequência, comprimento de onda, velocidade da onda, intensidade, espectro visível, fontes de luz policromática e monocromática etc.

²https://phet.colorado.edu/pt_BR/simulation/color-vision

4.1.4 Prática

Para montarmos os experimentos vamos precisar dos seguintes materiais da tabela 4.2. Os materiais devem ser utilizados conforme o esquema de montagem.

Tabela 4.2: Materiais utilizados na prática de óptica.

Quantidade	Item	Valor R\$
1	Kit base	73,60
3	Potenciômetros	11,50
3	Resistores de 220 Ohms, 0,25W	0,90
vários	Jumper	-
1	LED RGB de alto brilho	4,50
Total		90,40

O Kit base pode ser encontrados na tabela 3.1, o LED utilizado possui quatro terminais que são respectivamente da esquerda para a direita, vermelho, GND (o terminal GND possui a maior perna), verde e o azul (para maiores informações conferir o *Datasheet* do fabricante), e com a mistura formará também Amarelo (Vermelho e Verde), Ciano (Verde e Azul), Magenta (Azul e Vermelho) e Branco (Vermelho, Azul e Verde).

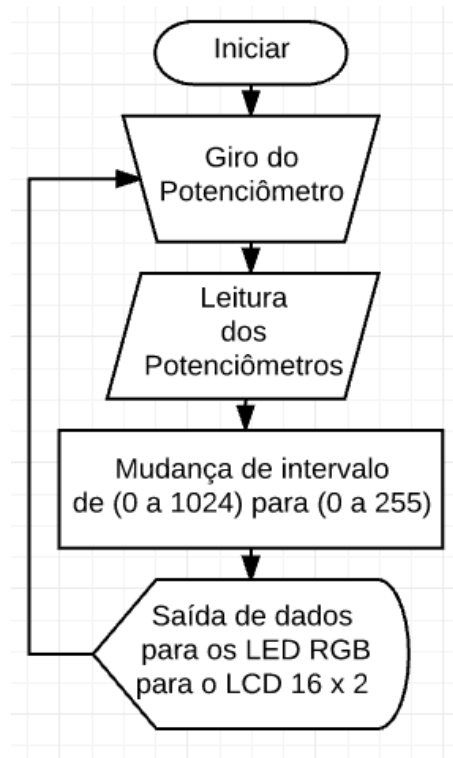
O LED representa um pixel no monitor, e o monitor por sua vez é formado por uma matriz destes dispositivos, para se formar a imagem o processador de vídeo acessa cada um dos dispositivos por sua linha e coluna e caracteriza-os com intensidade e cor, dessa forma a imagem é gerada.

Na montagem do experimento real utilizamos os pinos 9 (para o vermelho), 10 (para o verde) e 11 (para o azul), pois esses são os pinos que possuem PWM no Arduino. É necessário utilizar os pinos com PWM porque, segundo o site do Arduino, ao utilizar o sinal PWM, é possível controlar o potencial que vai para os pinos do LED RGB. Essa técnica permite ter um sinal aparentemente analógico através de um sinal digital. Dessa maneira, é possível criar muitas cores, fornecendo diferentes valores de tensão para o LED.

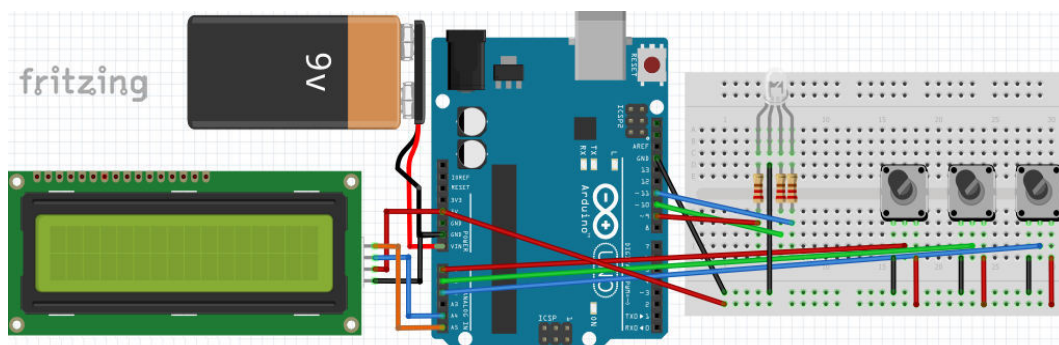
São necessários também três resistores 220 Ohms a 0,25W para limitar a passagem de corrente em cada pino de entrada do LED, pois caso contrário, o mesmo poderá queimar. Um *jumper* que ligue o catodo do LED ao GND do Arduino deve ser providenciado para fechar o circuito.

O resultado da montagem seguindo o esquema (ver figura 4.5b) e baseado no fluxograma (ver figura 4.5a), o fluxograma simplifica o experimento para melhor entendermos o processo de leitura e caracterização das intensidades e cores no LED. A compilação (*Sketch: Formação das Cores - Óptica*) no Arduino deve gerar os resultados demonstrados na figura 4.5c já era esperado e discutido. O *sketch* pode ser encontrado no apêndice A.

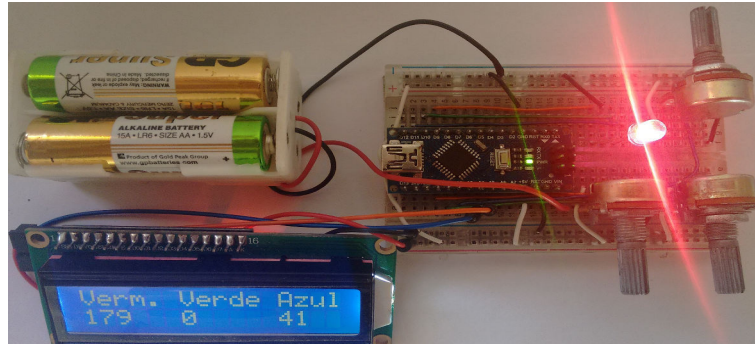
O experimento pode ser incrementado com a utilização de um LDC 16x2, que tem como objetivo apresentar a intensidade em uma escala de 0 a 255. Esse por sua vez é opcional, sua utilização se mostra necessário quando queremos relacionar a intensidade que o olho humano está recebendo de cada cor, com a cor percebida. É possível também detectar a cor preta (0,0,0) e a cor branca (255,255,255), esses são os valores que o processador de vídeo manda para cada um dos LEDs em um monitor.



(a) Fluxograma.



(b) Esquema elétrico do experimento com LCD 16x2.



(c) Experimento.

Figura 4.5: Esquema de montagem do experimento e o resultado esperado na montagem.

4.1.5 Guia de Atividade - Formação das Cores

Este guia da atividade envolve o conceito de formação das cores. Usando seus conhecimentos responda ao questionário individualmente antes de realizar o experimento, fazendo uma predição teórica com base nos seus conhecimentos. Em seguida, você interagirá com o experimento. Compare suas respostas. Neste experimento será utilizado um Arduino como gerenciador de três fontes luminosas utilizando um LED RGB e três potenciômetros.

Prof. Cleiton Malvessi

Aluno(a):..... Turma:..... Data:...../...../.....

Imagine a seguinte situação modelo, um monitor de TV gera as cores de uma imagem basicamente utilizando um componente eletrônico, o LED RGB, que por sua vez emite somente três cores base, Vermelho, Verde e Azul (RGB). O componente é disposto em uma matriz e a formação de imagens se dá pela sequência de cores em cada um dos LEDs (*pixel*) que podemos ver quando se tem uma gota de água na tela. Baseado nessa situação modelo responda o questionário seguinte. Justifique todas as suas respostas.

Questionário: (tempo previsto: 10 minutos)

I. O que é luz branca?

II. Como se dá a formação de outras cores além do Vermelho, Verde e Azul?

III. Qual cor um objeto vermelho emite ao incidir luz branca?

IV. Qual cor um objeto vermelho emite ao incidir luz verde?

V. O que é um tom de cor, e como se gera um tom diferente?

VI. Se tivermos 3 cores e cada cor gerar 256 tons diferentes, quantos novos tons podemos gerar?

VII. Se o arranjo dos LEDs é uma matriz, como se formação de imagem na TV?

Capítulo 5

Considerações Finais

Uma das principais metas do presente trabalho consistiu na busca por uma alternativa para reduzir as dificuldades encontradas pelos professores de Física em transmitir os conteúdos. Para este fim, desenvolvemos um material instrucional de baixo custo com a utilização de tecnologias livres, voltado para o ensino de Física no Ensino Médio, de posse da placa de prototipagem Arduino-UNO. O material foi desenvolvido baseado no conceito de Recursos Educacionais Abertos (REA) que permite que todo ele possa ser utilizado livremente, estudado, adaptado, copiado e distribuído às condições e interesses de professores e usuários a qualquer momento.

Neste trabalho optamos por desenvolver um material instrucional que proporcionasse a contextualização, compreensão e envolvimento dos alunos no estudo da disciplina de Física. Já no primeiro ano do Ensino Médio, foi uma alternativa encontrada para a aplicação deste tipo de recurso didático.

Por acreditarmos que os alunos possam avaliar suas próprias ideias e concepções, optamos pelo uso da metodologia PIE, que permite a discussão em pequenos grupos, e com a turma toda. Acreditamos que adoção de tal metodologia possibilitará a realização de aulas mais dinâmicas, que contará com a participação efetiva dos alunos durante a realização das aulas.

Os resultados de aprendizagem que esperamos neste trabalho indicam que o uso de experimentos com demonstração de dados em tempo real nas aulas de Física em conjunto com uma metodologia adequada de ensino deve contribuir para o aprendizado dos alunos, pois propiciam uma contextualização dos conteúdos e permitem que as aulas sejam centrados no aluno e não apenas no professor.

Este trabalho proporciona aos professores de Física do Ensino Médio trabalharem tanto a Física, Programação, como também a Matemática no ensino médio. Acredito que se pretendemos que nossos alunos tenham um aprendizado significativo, devemos desenvolver materiais que permitam uma boa inter-relação entre teoria e prática, através

da discussão e reflexão sobre os conceitos e suas aplicações no mundo real. Também é a chance de mostrarmos ao aluno que a Física não é um “bicho de sete cabeças”, nem uma série de verdades inquestionáveis, e que pressupõe um processo de construção e avaliação permanente de modelos científicos.

Durante a realização deste trabalho, muitos dos conceitos que imaginava ter domínio, foram refeitos, o que me proporcionaram uma nova aprendizagem que considero de maior significado, pois saber os conceitos e não compreender as relações e as influências dos mesmos com o mundo é um desperdício de recurso que está a disposição para que compreendamos com maior clareza a Física.

Referências

- [1] ARDUINO **Home**. 2016. Disponível em: <<https://www.arduino.cc/>>. Acesso em: 30 Abril de 2016.
- [2] CAVALCANTE, M. A.; BONIZZIA, A.; GOMES, L. C. P. Aquisição de dados em laboratórios de física: um método simples, fácil e de baixo custo para experimentos em mecânica. **Revista Brasileira de Ensino de Física** v33, n.4, p.1-9, 2008.
- [3] CEPSRM; Centro Estadual de Pesquisas em Sensoriamento Remoto e Meteorologia; UFRGS; **Dinâmica para Aprendizado do Sensoriamento Remoto. Formação das Cores**. 2016. Disponível em: <<http://www.ufrgs.br/engcart/PDASR/formcor.html>>. Acesso em: 24 junho de 2016.
- [4] AUSUBEL, D. P., NOVAK, J.D., HANESIAN H. **Psicologia Educacional**. Rio de Janeiro: Interamericana, 1980.
- [5] DEITEL, H. M. **C++: Como programar**. v.5, São Paulo: Pearson Prentice Hall, 2006.
- [6] DORNELES, P. F. T. **Integração entre atividades computacionais e experimentais como recurso instrucional no ensino de electromagnetismo em física geral**. 2010. 184 f. Tese de doutorado – Instituto de Física, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010.
- [7] EXPERIMENTORIA; **Arduino: você conhece sua história?** Disponível em: <<http://www.experimentoria.com.br/arduino-voce-conhece-sua-historia/>>. Acesso em: 30 Abril de 2016.
- [8] FETZNER, G. F.; **Experimentos de baixo custo para o ensino de Física em Nível Médio usando a placa Arduino-UNO**. Porto Alegre: UFRGS, 2015. Dissertação (Mestrado Profissional em Ensino de Física) - Programa de Pós-Graduação em Ensino de Física, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2015.

- [9] FIOLHAIS, C.; TRINDADE, J.; **Física no Computador: O Computador como uma Ferramenta no Ensino e na Aprendizagem das Ciências Físicas.** Revista Brasileira de Ensino de Física, v.25, n.3, p. 259 – 272. Set. de 2003.
- [10] GOHN, M. G. **Educação Não Formal e Cultura Política.** São Paulo: Cortez, 2001.
- [11] IGOE, Tom. and O’SULLIVAN Dan. **Physical Computing – Sensing and Controlling the Physical World with Computers.** Boston, MA: Thomson Course Technology. 2004.
- [12] KIRIANAKI, N. et al.; **Data Acquisition and Signal Processing for Smart Sensors.** Baffins Lane: John Wiley & Sons Ltd, 2002.
- [13] MARANDINO, M.; Silveira, R. V. M.; Chelini, M. J. E. ; Fernandes, A. B.; Garcia, V. A. R.; Martins, L. C.; Lourenço, M. F.; Fernandes, J. A.; Florentino, H. A.; **Educação Não Formal e a Divulgação Científica: o que pensa quem faz?** In: Atas do IV Encontro Nacional de Pesquisa em Ensino de Ciências. ENPEC, p. 1-11, 2004.
- [14] MEDEIROS, A.; MEDEIROS, C. F.; **Possibilidades e Limitações das Simulações Computacionais no Ensino da Física.** v.24, n.2, São Paulo, Junho de 2002.
- [15] MOREIRA, M.A.; MASINI, E.F.S. **Aprendizagem Significativa: a teoria de David Ausubel.** São Paulo: Moraes, 1982.
- [16] TAO, P. K.; Gunstone, R. F. **The process of conceptual change in force and motion during computer-supported physics instruction** Journal of Research in Science Teaching, New York , v. 36, n. 7, p. 859-882, Sept. 1999.
- [17] TIPLER, P.; **Física para cientistas e engenheiros.** 4ª Ed. Editora livros técnicos e científicos S.A. Rio de Janeiro. 2000.
- [18] VIEIRA, V., M. Lucia Bianconi, Monique D. **Espaços não formais de ensino e o currículo de Ciências.** 2005. Disponível em: <http://cienciaecultura.bvs.br/pdf/cic/v57n4/a14v57n4.pdf>. Acesso em: 15 de janeiro de 2016.
- [19] SILVA, O. Q. **Estrutura de dados e algoritmos usando C - fundamentos e aplicações.** Rio de Janeiro: Editora Ciência Moderna Ltda., 2007.

- [20] SOARES, R. R.; Borges, P.F. **O Plano Inclinado de Galileu: uma medida manual e uma medida com aquisição automática de dados.** Revista Brasileira de Ensino de Física, v. 32, n. 2, 2501, 2010.

Apêndice - A

Sketch: Formação das Cores - Óptica

```
1  /*****
2   Autor      : Cleiton Malvessi, 2016.
3   Descrição  : Espectro Visível, formações de cores com um LED RGB.
4   *****/
5   #include <Wire.h>
6   #include <LiquidCrystal_I2C.h>
7
8   // Inicializa o display no endereço 0x27
9   LiquidCrystal_I2C lcd(0x3F, 2, 1, 0, 4, 5, 6, 7, 3, POSITIVE);
10
11  int redPin = 9;    // Variável redPin, Pino 9 para cor vermelha.
12  int greenPin = 10; // Variável greenPin, Pino 10 para cor verde.
13  int bluePin = 11;  // Variável bluePin, Pino 11 para cor azul.
14
15  //Pinos de leitura dos valores no potenciômetro para cada uma das cores.
16  int redIn = A0;    // Vermelho.
17  int greenIn = A1; // Verde.
18  int blueIn = A2;  // Azul.
19
20  //Variaveis que armazenam o valor lido no Potenciômetro para as cores.
21  int redVal;       // Vermelho.
22  int greenVal;     // Verde.
23  int blueVal;      // Azul.
24
25  void setup() {
26    lcd.begin(16, 2); // Inicializar o LCD
27    Serial.begin(9600); // Inicializar o Monitor Serial
28  }
29
30  void loop() {
31    // Atribui os valores das leituras nas respectivas variáveis.
32    redVal = analogRead(redIn);    // Vermelho.
33    greenVal = analogRead(greenIn); // Verde.
34    blueVal = analogRead(blueIn);  // Azul.
35
36    // O ciclo de trabalho para a função "(analogWrite( , );" é
37    // entre 0 (sempre desligado) e 255 (sempre ligado).
38    // Os valores lidos no potenciômetro é de 0 a 1204.
39    // A função (map) se encarrega de fazer a mudança de intervalo.
40    redVal = map(redVal, 0, 1023, 0, 255);    // Vermelho.
41    greenVal = map(greenVal, 0, 1023, 0, 255); // Verde.
42    blueVal = map(blueVal, 0, 1023, 0, 255);  // Azul.
43
44    // Atribui o valor para cada pino
45    analogWrite(redPin, redVal);    // Intensidade da cor vermelha.
46    analogWrite(greenPin, greenVal); // Intensidade da cor verde.
47    analogWrite(bluePin, blueVal);  // Intensidade da cor azul.
48  {
49    // Adicionar o Código opcional aqui.
50    // Para o LCD 16X2 ou para Monitor serial.
51  }
52 }
```

Sketch Opcional: Formação das Cores - Óptica

```
{
  // os comando de Imprimir é opcional.
  // Imprime as cores e o valor do tom no LCD 16X2.
  lcd.setCursor(0, 0);      // Ajusta o cursor, Coluna 1 linha 1.
  lcd.print("Verm.");      // Imprime a palavra Vermelho.
  lcd.setCursor(0, 1);     // Ajusta o cursor, Coluna 1 linha 2.
  lcd.print(redVal);       // Imprime o valor atribuído a variável redVal.

  lcd.setCursor(6, 0);     // Ajusta o cursor, Coluna 6 linha 1.
  lcd.print("Verde");      // Imprime a palavra Verde.
  lcd.setCursor(6, 1);     // Ajusta o cursor, Coluna 6 linha 2.
  lcd.print(greenVal);     // Imprime o valor atribuído a variável redVal.

  lcd.setCursor(12, 0);    // Ajusta o cursor, Coluna 12 linha 1.
  lcd.print("Azul");       // Imprime a palavra Azul.
  lcd.setCursor(12, 1);    // Ajusta o cursor, Coluna 12 linha 2.
  lcd.print(blueVal);      // Imprime o valor atribuído a variável redVal.
  lcd.clear();

  // Imprime as cores e o valor do tom no Monitor Serial.
  Serial.print("vermelho:"); // Imprime a palavra Vermelho.
  Serial.print(redVal);      // Imprime o valor atribuído a variável redVal.
  Serial.print(" ");        // Imprime um espaço.

  Serial.print("verde:");   // Imprime a palavra Verde.
  Serial.print(greenVal);   // Imprime o valor atribuído a variável greenVal.
  Serial.print(" ");        // Imprime um espaço.

  Serial.print("Azul:");    // Imprime a palavra Azul.
  Serial.println(blueVal);  // Imprime o valor atribuído a variável blueVal.
}
```