

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ - CAMPO MOURÃO
PROGRAMA DE MESTRADO PROFISSIONAL EM ENSINO DE FÍSICA

JANAINA DE OLIVEIRA REIS TOUREIRO

Produto Educacional

**KIT PARA PROGRAMAÇÃO BÁSICA DE ARDUINO EM RESISTORES
NÃO LINEARES NO 9º ANO DO ENSINO FUNDAMENTAL**

Produto Educacional apresentado ao Programa de Mestrado Profissional em Ensino de Física da Universidade Tecnológica Federal do Paraná (Polo 32 MNPEF), campus Campo Mourão, como requisito parcial à obtenção do título de Mestre em Ensino de Física.

Orientador: Prof. Dr. Gilson Junior Schiavon
Coorientador: Prof. Dr. Michel Corci Batista

Campo Mourão
2019

1 Apresentação

Pensando no cenário educacional e o momento tecnológico onde a sociedade está inserida é que nos motivou a buscar por técnicas e estratégias de ensino que desperte e promova uma aprendizagem significativa. Por muitos anos a transmissão de conhecimentos tem se pautado somente em teorias, a experimentação não encontra lugar nessa aprendizagem passiva, essas abordagens não podem ser tratadas dicotomicamente, uma é alicerce da outra, dando sentido e significado aos conhecimentos adquiridos.

Muitas vezes a falta de material pedagógico e o despreparo do docente na formação acadêmica, acabam limitando o educador na preparação de suas aulas.

Nessa perspectiva, nosso produto educacional se constitui como um material de apoio, que tem o intuito de auxiliar o professor na apresentação do tema Resistores Não Lineares, de forma a promover uma participação mais efetiva dos alunos durante as aulas de Física. Nossa proposta de intervenção está embasada em uma metodologia que prioriza a atividade experimental e a utilização das TICs no ensino de Física, bem como valoriza o trabalho em equipe, criando assim condições para que os alunos instaurem um clima de parceria e de colaboração entre eles.

Para a execução do roteiro de ensino, estão previstas 6 h/a, podendo esse número ser alterado, caso haja necessidade.

O roteiro de ensino está organizado em três módulos, cujo objetivo é estabelecer um diálogo entre os diferentes saberes sociais e fazer com que o cotidiano possa ser objeto de estudo.

2 UMA INTRODUÇÃO AO ESTUDO DOS RESISTORES

Vamos iniciar buscando entender o que é a grandeza física resistência. Resistência é uma propriedade que todos os materiais oferecem. Nos bons condutores, essa resistência é baixa. Nos isolantes, a resistência é tão grande que praticamente não há passagem de corrente elétrica² por entre suas moléculas. Assim a resistência pode ser definida como sendo a medida da dificuldade que um material oferece à passagem da corrente.

Fique ligado!!!

Quanto maior o valor da resistência, menor o valor da corrente que circula pelo circuito.

Nos circuitos eletrônicos, utiliza-se um dispositivo chamado resistor para controlar a intensidade da corrente elétrica que percorre o circuito.

Um efeito sempre presente nos resistores é o seu aquecimento quando é percorrido pela corrente elétrica, chamado de efeito Joule. Devido a esse efeito, podemos afirmar que a energia elétrica é transformada em energia térmica.

Assim, resistor deve ser entendido como um dispositivo eletrônico que apresenta como funções principais: transformar energia elétrica em energia térmica (Efeito Joule) e dificultar a passagem de corrente elétrica. Entende-se à dificuldade que o resistor apresenta a essa passagem de corrente como sendo sua resistência elétrica (Batista e Batista, 2016, p.126).

² Corrente elétrica: é o movimento ordenado de cargas elétricas negativas no interior de um condutor devido a uma diferença de potencial elétrico.

Fique ligado!!!

Em alguns casos o aquecimento do resistor é desejável, por exemplo, nos equipamentos que aquecem água ou ar, como chuveiro, secadores, ferro de passar. Em outros, é totalmente indesejável, como nos circuitos de computadores, porque deixa seu processamento mais lento e pode danificar alguns componentes.

Os resistores podem ser classificados em dois tipos: os fixos e os variáveis.

2.1 RESISTORES FIXOS

Os resistores fixos, figura 1, são aqueles cujo o valor da resistência não pode ser alterada, ou seja, após sua fabricação sua resistência se mantém constante.

Figura 1: Resistor fixo



Fonte: Batista e Fusinato, 2014, p.15

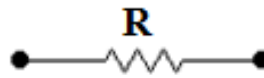
Os resistores fixos são especificados por três parâmetros: o valor nominal da resistência elétrica, a tolerância e máxima potência elétrica dissipada.

O valor de um resistor fixo pode ser facilmente identificado de acordo com as cores que apresenta na cápsula que envolve o material resistivo, perceba na figura 1, ou então usando um ohmímetro, instrumento que efetua a medida da grandeza resistência elétrica.

Simbologia

A figura 2 mostra o símbolo utilizado para a representação de um resistor, as características específicas dos resistores aparecem acima do símbolo:

figura 2: Representação de um resistor



Fonte: Batista e Fusinato, 2014, p.16

Os resistores de acordo com a sua constituição podem ser classificados de diferentes maneiras:

- a) Resistores de filme de carbono;
- b) Resistores de carvão;
- c) Resistores de fio.

Cada um dos tipos tem em sua constituição características que o torna mais adequado que outros tipos em sua classe de aplicação.

a) Resistor de filme de carbono - (baixa potência)

O resistor de filme de carbono, também conhecido como resistor de película, é constituído por um corpo cilíndrico de cerâmica que serve de base para a fabricação do componente. Sobre esse corpo é enrolado em forma de espiral, uma fina camada de material resistivo (conhecido como filme de carbono) que determina o valor ôhmico do resistor. Após conectado os terminais (de conexão) às extremidades do fino material resistivo, o corpo do resistor recebe uma camada de revestimento que dá acabamento na fabricação e isola o filme de carbono da ação da umidade.

As características fundamentais do resistor de filme de carbono são a precisão e a estabilidade do valor da resistência.

b) Resistor de carvão - (média potência)

O resistor de carvão é constituído por um corpo cilíndrico de porcelana. No interior da porcelana são comprimidas partículas de carvão que definem a resistência do componente e o valor da resistência depende da concentração de partículas de carvão. Quanto maior a concentração, menor a resistência.

Esses resistores apresentam pequenas dimensões (tamanho físico reduzido). Os valores de dissipação e resistência não são precisos.

c) Resistor de fio - (alta potência)

O resistor de fio é constituído por um corpo cilíndrico de porcelana ou cerâmica. Sobre esse corpo é enrolado um fio especial (por exemplo: níquel cromo) cujo comprimento e seção transversal determinam o valor da resistência do resistor. Esses resistores não possuem anéis coloridos e o valor da resistência e da potência, normalmente, vem escrito no corpo do componente.

Os resistores de fio tem capacidade para trabalhar com correntes maiores. Esse tipo de resistor dissipa grande potência produzindo assim grande quantidade de calor quando em funcionamento.

Esse tipo de resistor pelo fato de produzir grande quantidade de calor é construído sobre um tubo oco de porcelana para facilitar o resfriamento.

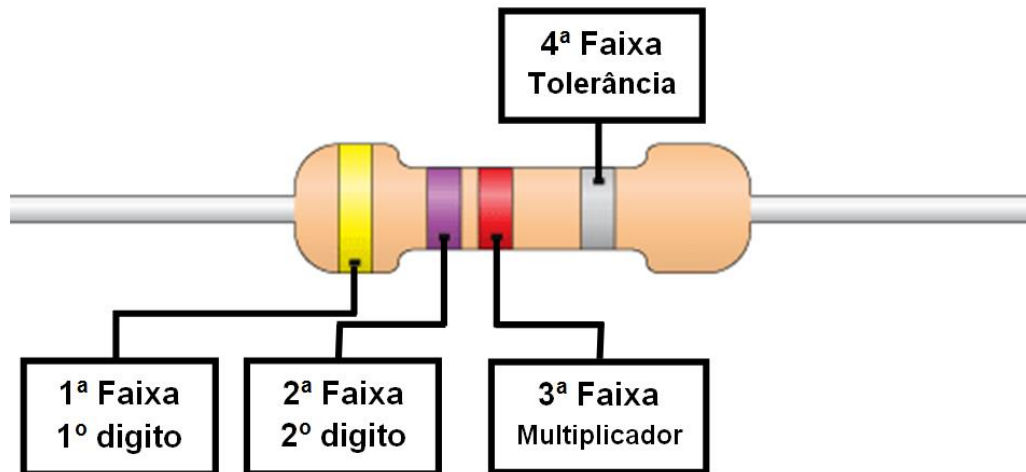
Código de cores

Os valores das resistências dos resistores podem ser reconhecidos pelas cores das faixas em suas superfícies, cada cor e sua posição no corpo do resistor representa um número, de acordo com o seguinte esquema:

Importante!

O resistor deve ser posicionado com as faixas coloridas da esquerda para a direita, para que a leitura seja feita corretamente.

Figura 3: A partir do código de cores



Fonte: Batista e Fusinato, 2014, p.18

Identificando a resistência do resistor

O valor da resistência é obtido de acordo com a ordem que aparece as cores das faixas. Os dois primeiros algarismos são representados pelas cores da primeira e segunda faixa, a cor da terceira faixa indica o expoente da potência de 10 que deve ser multiplicado pelo número formado pelos algarismos citados anteriormente (ou seja, é a quantidade de zeros) e a cor da quarta faixa informa a tolerância.

Tabela de Cores

Tabela 1: Código de cores para resistores fixos

Cor	Código
Preto	0
Marrom	1
Vermelho	2
Laranja	3
Amarelo	4
Verde	5
Azul	6
Violeta	7
Cinza	8
Branco	9
Cor	Tolerância
Ouro	+/- 5%
Prata	+/- 10%
Sem cor	+/- 20%

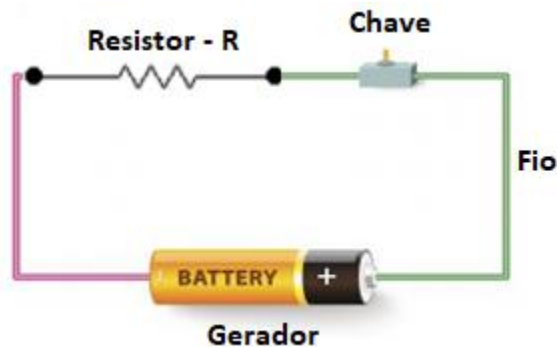
Fonte: Batista e Fusinato, 2014, p.19

Praticando			
Vamos fazer a leitura de resistor que apresenta suas faixas coloridas nessa sequência:			
Amarelo	Violeta	Laranja	Prata
4	7	000	+/- 10%

1ª Lei de Ohm

Devemos inicialmente pensar em um circuito elétrico simples, constituído por um gerador (que pode ser uma pilha ou uma bateria), um resistor fixo, pedaços de fios e uma chave (ou interruptor), como representado na figura 4.

Figura 4: Circuito elétrico simples



Fonte: Os autores (2019)

Quando fornecemos uma diferença de potencial para um circuito elétrico surge nesse circuito uma corrente elétrica e é esta que permite o equipamento entrar em funcionamento, por exemplo, uma lâmpada acende, um ferro de passar aquece.

Em 1826, George Simon Ohm descobriu que para condutores metálicos a tensão varia linearmente com a corrente elétrica, ou seja, a razão entre a tensão e a corrente elétrica é constante e esta é denominada resistência R do condutor que é dada pela equação:

$$R = \frac{U}{i}$$

Em que U é a diferença de potencial nas extremidades do condutor e i é a intensidade da corrente elétrica que o atravessa. A unidade de resistência elétrica no SI é dada pela razão $\frac{\text{Volt}}{\text{Ampère}}$ e recebe o nome de ohm (Ω).

Ele também observou que mantida a temperatura constante, a tensão e a intensidade da corrente são diretamente proporcionais.

$$\frac{U_1}{i_1} = \frac{U_2}{i_2} = \frac{U_n}{i_n} = \textit{constante}$$

E formulou a equação abaixo conhecida como a 1ª Lei de Ohm.

$$U = R \cdot i$$

Logo essa equação é usada para definir o conceito de resistência e se aplicam a todos os dispositivos que conduzem corrente elétricas. Desse modo temos dois tipos de resistores: os ôhmicos e os não ôhmicos. Os resistores ôhmicos são aqueles cuja resistência elétrica é constante.

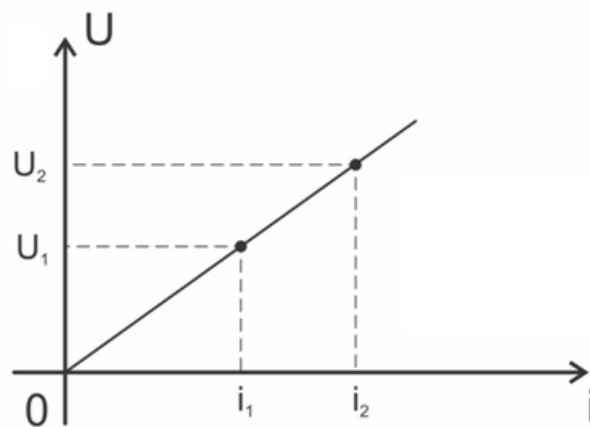
Nos casos em que a resistência elétrica não é constante, ou seja, em que a razão entre U e i não são constantes e sua variação depende da tensão U , dizemos que o resistor não é ôhmico.

O gráfico $U \times i$ permite saber de forma prática quando um resistor é ôhmico ou não ôhmico.

Curva característica dos resistores

Se a resistência elétrica (R) for constante, o gráfico de U em função de i é uma reta, cujo o coeficiente angular é o valor da resistência elétrica desse resistor, figura 5.

Figura 5: Curva característica de um resistor ôhmico



Com isso podemos concluir que:

$$tg\theta = \frac{U_1}{i_1} = \frac{U_2}{i_2} = \frac{U_n}{i_n} = R$$

Fique ligado!

Os resistores fixos que obedecem a 1ª lei de Ohm são chamados de resistores ôhmicos.

Unidade de Medida

O ohm é a unidade de medida da resistência elétrica, padronizada pelo Sistema Internacional de Unidades (S.I.). Corresponde à relação entre a tensão de um volt e uma corrente de um ampère sobre um elemento, seja ele um condutor ou isolante. Assim, um condutor que tenha uma resistência elétrica de 1 ohm, causará uma queda de tensão de 1 volt a cada 1 ampère de corrente que passar por ele.

O ohm é simbolizado pela letra grega ômega maiúsculo (Ω) e como muitas vezes é insuficiente utilizar somente a medida de referência para quantificar algo, teremos que usar **múltiplos** e **submúltiplos**, onde é utilizado o sistema decimal (base 10).

Tabela 2 – Múltiplos e Submúltiplos da unidade de medida ohm

	Múltiplos				Unidade de Referência	Submúltiplos			
Prefixo	Tera	Giga	Mega	Kilo	Ohm	Mili	Micro	Nano	Pico
Símbolo	T	G	M	k	Ω	m	μ	n	p
Potência	10^{12}	10^9	10^6	10^3	10^0	10^{-3}	10^{-6}	10^{-9}	10^{-12}

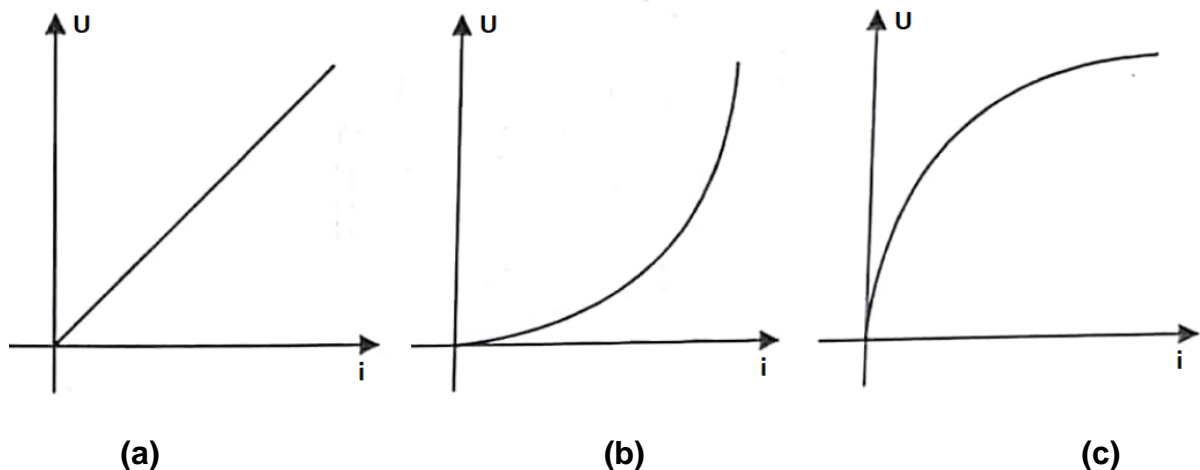
2.2 RESISTORES COM RESISTÊNCIAS VARIÁVEIS

A primeira Lei de Ohm não pode ser aplicada integralmente a todos os materiais, uma vez que alguns não têm uma relação linear entre tensão e corrente, ou seja, suas resistências não se mantêm constantes com a variação da tensão aplicada. Por isso, esses materiais são chamados de não ôhmicos.

Em geral, esses materiais são mais sensíveis à temperatura, que altera tanto suas resistividades quanto suas dimensões.

A figura 6 representa três gráficos $U \times i$, sendo o gráfico (a) correspondente a um material ôhmico e os gráficos (b) e (c) correspondentes a materiais não ôhmicos.

Figura 6: Gráficos de $U \times i$ para diferentes resistores



Fonte: Os autores (2019)

Um exemplo típico de material não ôhmico é o filamento de uma lâmpada incandescente, cuja resistência aumenta com o aumento da tensão aplicada, ou seja, sua curva característica tem o aspecto do gráfico 6 (b).

2.2.1 Sensores de luz e temperatura

São chamados de sensores, dispositivos que são sensíveis a fenômenos físicos, como, por exemplo: luz, temperatura, impedância elétrica e outros. Estes transmitem um sinal para outro dispositivo capaz de medir e informar a variação do valor do fenômeno observado.

Sensores de luminosidade ou LDR (*Light Dependent Resistor*), figura 7, são resistores variáveis. Mediante a incidência da radiação eletromagnética do espectro visível a resistência apresenta variações.

Figura 7 - Sensor de Luminosidade LDR 5mm



Fonte: <https://www.ime.usp.br/~adao/sensores.pdf>

Sensores de temperatura ou Termistores NTC (*Negative Temperature Coefficient*), figura 8, são resistores termicamente sensíveis. Por isso, exibem variação de resistência ôhmica em relação a temperatura submetida. Assim, conforme a temperatura aumenta, a resistência térmica diminui.

Figura 8 - Sensor de Temperatura NTC 10K 5mm



Fonte: <https://www.ime.usp.br/~adao/sensores>.

ATIVIDADE 1: Conhecendo e Programando o Arduino

O Arduino é baseado em hardware e software livre apresentando vários modelos. Um deles é o Arduino UNO formado a partir de uma pequena placa de circuitos. Por ser uma plataforma aberta os usuários podem aproveitar a versão básica ou desenvolver extensões aprimoradas. Neste trabalho, será utilizado o Arduino UNO.

Esta plataforma surgiu na Itália em 2005, facilitando tarefas complicadas de programação, assim qualquer usuário com o mínimo de conhecimento em programação é capaz de dominar os códigos abertos em linguagem simples, conhecido como ambiente integrado de desenvolvimento (IDE).

O Arduino foi criado com o objetivo de fornecer uma plataforma de fácil prototipação de projetos interativos, que são características da computação gráfica. A computação gráfica é uma área da computação na qual o software se comunica diretamente com o hardware, controlando componentes eletrônicos, como sensores e atuadores, permitindo a construção de sistemas (códigos) que consigam perceber e interagir com ambientes reais.

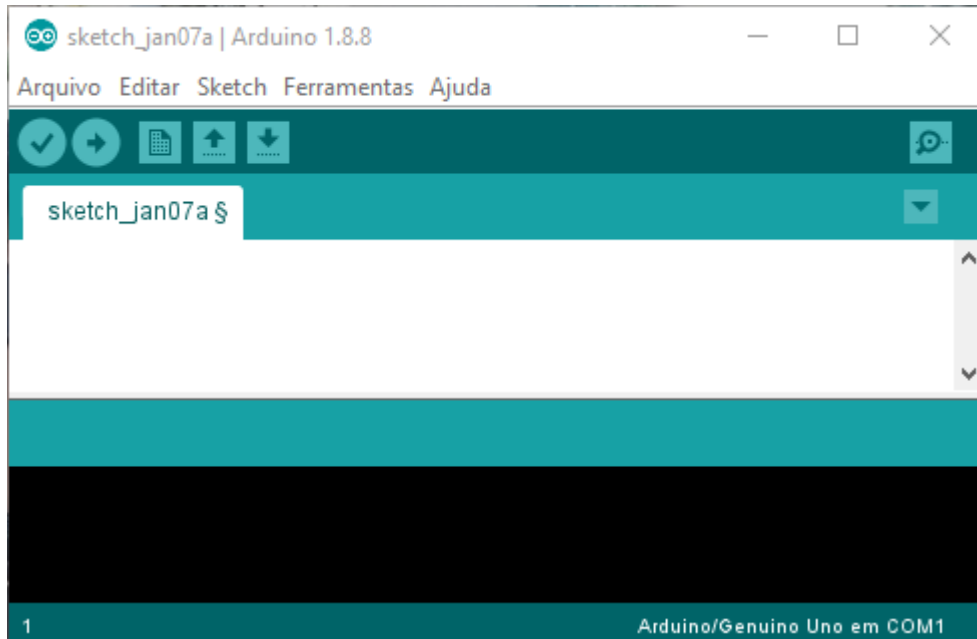
O Arduino pode ser entendido como uma plataforma eletrônica de prototipagem rápida, fundamentada sob licença de código aberto e sendo constituído de duas partes, figuras 9 e 10:

Figura 9. Hardware Arduino UNO



Fonte: <https://www.ime.usp.br/~adao/sensores.pdf>

Figura 10. Software IDE



Fonte: Print da tela inicial do programa

Conforme visto na figura 9, a placa Arduino UNO é formada por diversos conectores que atuam como interface para o ambiente externo. Na figura 10, temos a imagem do ambiente de programação (software IDE), quando aberto para programar o Arduino. Este permite a criação de Sketches para a placa Arduino, por meio de códigos escritos em linguagem C, que são gravados no Arduino, de forma que o mesmo pode ser usado como um computador ao ser ligado a uma fonte de energia, executando o Sketch criado de forma independente.

Após, a instalação do IDE no computador, o Arduino é conectado por meio de um cabo USB próprio do aparelho. Logo, o IDE possibilita a utilização de muitos recursos, a programação pode ser escrita ou simplesmente baixada para o Arduino que executará os comandos programados. O IDE está disponível na internet e pode ser baixado diretamente da página do Arduino, ele possui versões adaptadas para vários tipos de sistemas operacionais, tais como Windows, Macintosh OSX e Linux.

A placa é de fácil manuseio e vem com vários sensores prontos para adicioná-los nas portas de entrada ou saída, facilitando a construção de experimentos físicos. Ele pode fazer leituras ou controles de sinais analógicos e digitais relacionados aos sensores, motores ou equipamentos com circuitos elétricos simples.

Pensando pelo lado financeiro e analisando a viabilidade de aplicação de um projeto educacional robótico nas escolas, este material tecnológico apresenta peças

de baixo custo e também é conhecido como robótica livre ou *hardware* livre que em inglês é conhecido como *Open Source Hardware*.

Este equipamento é muito versátil devido às várias possibilidades de combinações de componentes e em diversas áreas de ensino, como por exemplo, estudo da aceleração da gravidade, coordenar uma rede trifásica, análise de dados ambientais e sísmicos, fonte de pulsos, ultrassom, e outros.

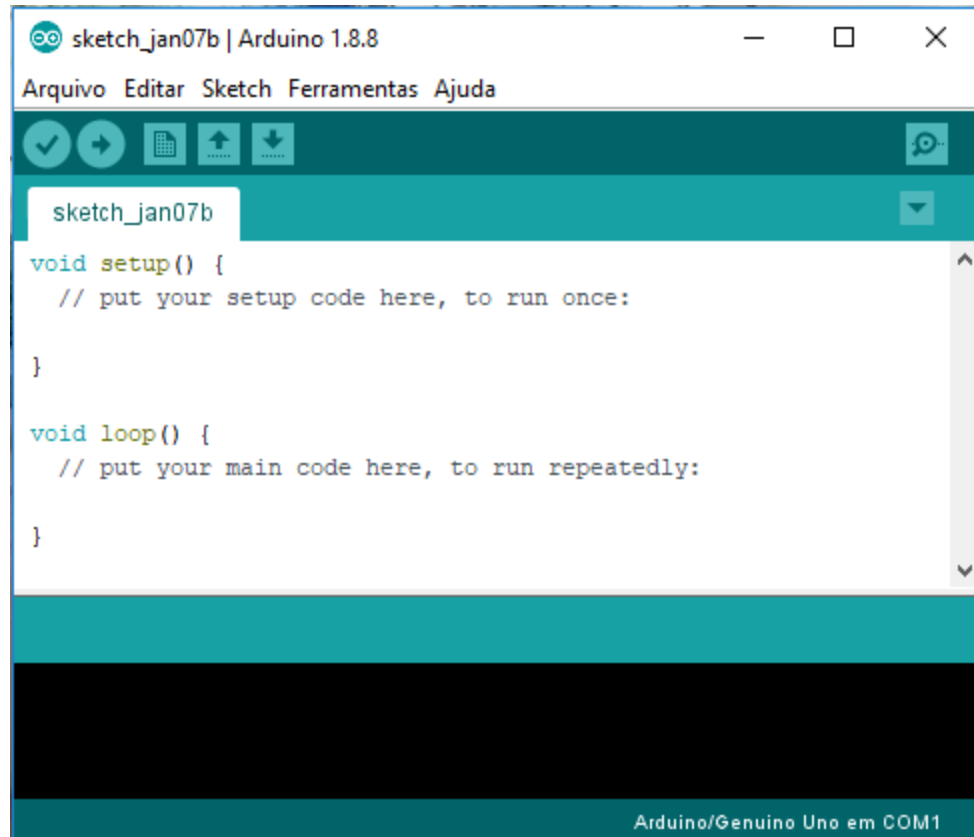
A estrutura básica da linguagem de programação do arduino é bem simples, formada por dois blocos de funções que carregam outros blocos de funções escritas em linguagem C ou C++. Funções em linguagem de programação são como sub-rotinas ou procedimentos, são pequenos blocos de programas usados para montar um programa principal, podem ser escritas por um programador ou importadas prontas para o programa em forma de biblioteca.

Toda função deve ser declarada antes de ser chamada atribuindo-lhe um tipo e um nome seguido de parênteses, onde serão colocados os parâmetros de passagem da função. Depois do nome são definidos entre chaves os procedimentos que a função vai executar.

Os dois blocos são as funções descritas a seguir:

- 1) **setup()**: essa é a primeira função a ser chamada quando o programa inicia, e é executada apenas nessa primeira vez. Esta é uma função de preparação, ela dita o comportamento dos pinos do arduino e inicializa a porta serial.
- 2) **Loop()**: essa função é chamada logo a seguir e todas as funções embarcadas nela são repetidamente executadas, ela fica lendo os pinos de entrada do arduino e comandando os pinos de saída e a porta serial.

A figura 11 apresenta a página do programa quando é dado o comando de iniciar uma nova programação, clicando em Arquivo → Novo. Repare que as duas funções já são inseridas automaticamente.

Figura 11. Iniciando um novo programa

```
sketch_jan07b | Arduino 1.8.8
Arquivo Editar Sketch Ferramentas Ajuda

sketch_jan07b
void setup() {
  // put your setup code here, to run once:
}

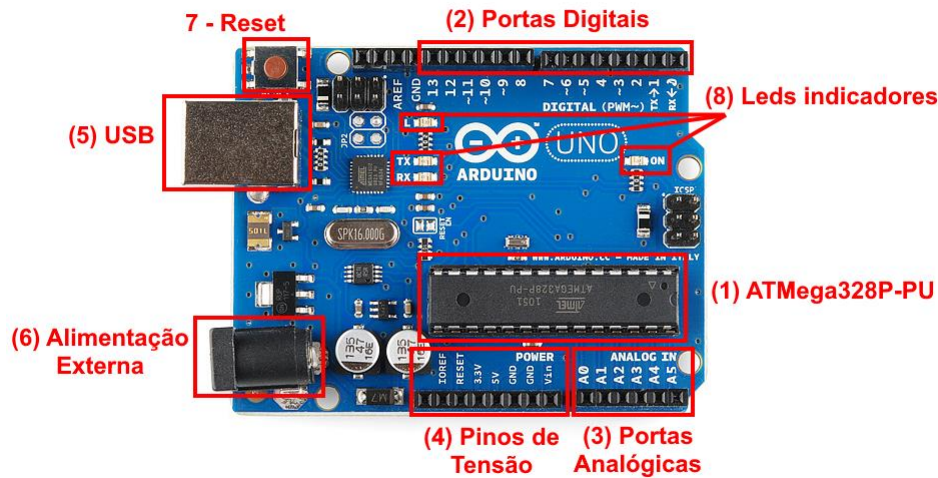
void loop() {
  // put your main code here, to run repeatedly:
}

Arduino/Genuino Uno em COM1
```

Fonte: Print da tela do programa

O microcontrolador presente na versão UNO do Arduino é o ATMEGA 328P-PU, de 8 bits da família AVR com arquitetura RISC avançada e com encapsulamento DIP28. Ele conta com 32 kB de memória Flash (mas 512 Bytes são utilizados pro bootloader), 2 kB de RAM e 1 kB de EEPROM. Pode operar a até 20 MHz, porém na placa Arduino UNO opera em 16 MHz, valor do cristal externo que está conectada aos pinos 9 e 10 do microcontrolador. Possui 14 pinos de entrada e saída (I/O) digitais e 6 pinos de entradas analógicas. Sua alimentação pode ser realizada pelo conector de fonte externa, pela conexão USB ou pelo pino Vin, essas informações podem ser observadas na figura 4 (EMBARCADOS, 2013).

Figura 12. Identificação dos conectores



Fonte: Aprendendo física

A placa pode ser alimentada pela conexão USB (item (5) da figura 12) ou por uma fonte de alimentação externa (item (6) da figura 12). A alimentação externa é feita através do conector *jack* com positivo no centro, onde o valor de tensão da fonte externa deve estar entre os limites de 6 V a 20 V, porém se alimentada com uma tensão abaixo de 7 V, a tensão de funcionamento da placa, que no Arduino Uno é 5 V, pode ficar instável e quando alimentada com tensão acima de 12 V, o regulador de tensão da placa, que é de 5 V, pode sobreaquecer e danificar a placa. Dessa forma, são recomendadas fontes com valores de tensão entre 7 V e 12 V.

Quando o cabo USB é plugado a um PC, por exemplo, a tensão não precisa ser estabilizada pelo regulador de tensão. Dessa forma a placa é alimentada diretamente pela USB. O circuito da USB apresenta alguns componentes que protegem a porta USB do computador em caso de alguma anormalidade.

O conector indicado como, (4) Pinos de Tensão, na figura 12, é responsável em fornecer tensão de alimentação para placas (*shields*) e circuitos externos a placa do arduino, como segue:

IOREF - Fornece uma tensão de referência para que *shields* possam selecionar o tipo de interface apropriada, dessa forma *shields* que funcionam com a placas Arduino que são alimentadas com 3,3 V podem se adaptar para ser utilizados em 5 V e vice-versa.

RESET - pino conectado ao botão de *RESET* do microcontrolador. Pode ser utilizado para um reset externo da placa Arduino.

3,3 V - Fornece tensão de 3,3 V para alimentação de *shield* e módulos externos, com fornecimento de corrente máxima de 50 mA.

5 V - Fornece tensão de 5 V para alimentação de *shields* e circuitos externos.

GND - pinos de referência, terra.

VIN - pino para alimentar a placa através de *shield* ou bateria externa. Quando a placa é alimentada através do conector *jack*, a tensão da fonte estará nesse pino.

Conforme exibido na figura 12 (item (2) Portas Digitais), a placa Arduino UNO possui 14 pinos que podem ser programados como entrada ou saída digitais. Estes pinos operam em 5 V, onde cada pino pode fornecer ou receber uma corrente máxima de 40 mA. Cada pino possui resistor de pull-up interno que pode ser habilitado por software.

Para interface com o mundo analógico, a placa Arduino UNO possui 6 entradas (item (3) Portas Analógicas da figura 12), onde cada uma tem a resolução de 10 bits. Por padrão a referência do conversor AD está ligada internamente a 5 V, ou seja, quando a entrada estiver com 5 V (tensão máxima admitida) o valor da conversão analógica digital será 1023. O valor da referência pode ser mudado através do pino AREF.

Para a execução deste trabalho precisamos programar 2 entradas analógicas para monitorar as variações da temperatura por meio de um sensor NTC10k e da luz incidente em um sensor de luz do tipo LDR, ambos sensores resistivos com comportamento não linear, programaremos as entradas analógicas A0 e A1, respectivamente. Em seguida, precisamos programar 2 portas digitais para operarem como saídas digitais, onde programaremos as saídas 2 e 3, desta forma quando a temperatura e/ou a luminosidade atingirem os valores programados, estas saídas serão acionadas mantendo-se em 5 V até que as condições de temperatura e luminosidade fiquem abaixo dos valores programados. Nota: poderiam ter sido escolhidas qualquer uma das 6 entradas analógicas, assim como qualquer uma das 14 portas digitais para operarem como saída digital.

Na saída digital 2 será ligado um cooler, o qual será acionado quando a temperatura atingir o valor programado, enquanto na saída digital 3 será ligado um conjunto Led/resistor que será acionado quando a luminosidade incidente no LDR atingir o valor programado, simulando um sistema de iluminação automática.

Para iniciarmos a programação, abrimos o software IDE previamente instalado no computador, selecionamos na barra superior o item Arquivo, em seguida Novo. Será apresentada uma tela semelhante a da figura 11.

Antes da função “void setup ()” devemos fazer as declarações das portas digitais a serem utilizadas, onde daremos os nomes de “saida1” para a porta 2 e “saida2” para a porta 3, como segue:

```
int saida1 = 2;
```

```
int saida2 = 3;
```

Na sequência entramos na função “void setup()”, onde definimos os pinos 2 e 3 declarados anteriormente como saídas, assim como inicializamos a porta serial, como segue:

```
pinMode(saida1, OUTPUT);
```

```
pinMode(saida2, OUTPUT);
```

```
Serial.begin(9600);
```

Para finalizar a programação entramos na função “void loop()”, onde serão programadas as leituras das tensões nas entradas analógicas, as quais serão convertidas em um valor digital por um conversor Analógico/Digital (A/D), que na sequência serão comparados com um valor de referência adotado como 2,5 V na programação. Como resultado dessa comparação teremos as saídas digitais 2 e 3, sendo ligadas ou não de acordo com o programado, resumindo, se as tensões nas entradas analógicas forem inferiores a 2,5 V as saídas digitais permanecem desligadas, caso estejam maiores ou iguais a 2,5 V as saídas digitais serão acionadas, como segue:

```
int sensor1 = analogRead(A0);
```

```
int sensor2 = analogRead(A1);
```

```
float valorentrada1 = sensor1 * (5.0 / 1023.0);
```

```
float valorentrada2 = sensor2 * (5.0 / 1023.0);
```

```
Serial.println(valorentrada1);
```

```
Serial.println(valorentrada2);
```

```
if (valorentrada1 >= 2.5) {digitalWrite(saida1, HIGH);}
```

```
else {digitalWrite(saida1, LOW);}
```

```
if (valorentrada2 >= 2.5) {digitalWrite(saida2, HIGH);}
```

```
else {digitalWrite(saida2, LOW);}
```

Na sequência é apresentado o programa completo, o qual pode ser copiado e colado no software IDE:

```
//Declaração das portas digitais a serem utilizadas;
int saida1 = 2;
int saida2 = 3;

void setup()
{
  //Define os pinos dos LED's referente a cada tensão aplicada na entrada analógica
  pinMode(saida1, OUTPUT);
  pinMode(saida2, OUTPUT);

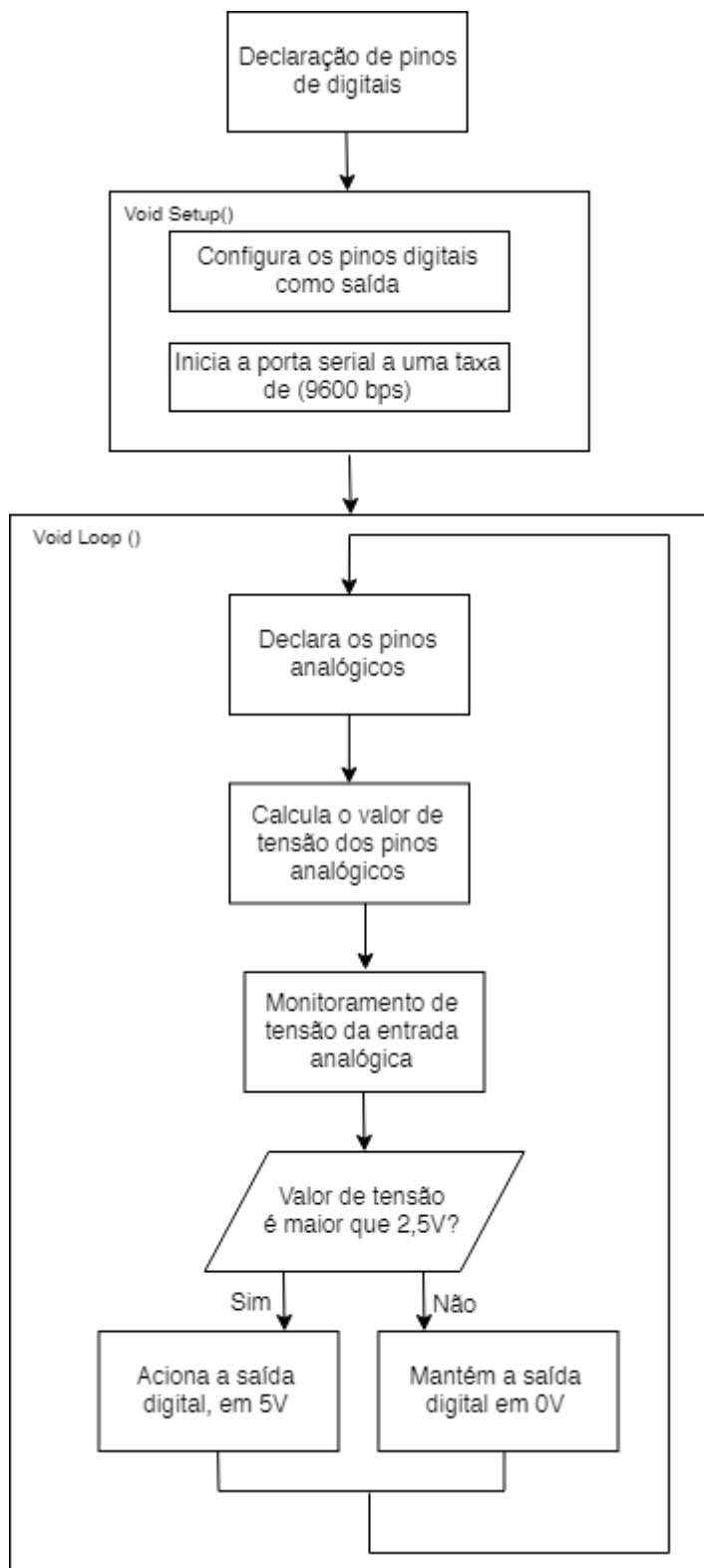
  //Inicializa a porta serial
  Serial.begin(9600);
}

void loop()
{
  //Faz a leitura do valor de tensão aplicado na entrada analógica e armazena em uma
  variável
  int sensor1 = analogRead(A0);
  int sensor2 = analogRead(A1);
  // 5 é o valor de tensão maximo do sensor, 1023 é referente aos 10 bits do conversor
  A/D ((2^10)-1).
  float valorentrada1 = sensor1 * (5.0 / 1023.0);
  float valorentrada2 = sensor2 * (5.0 / 1023.0);
  //Imprime na tela os valores de tensão de entrada
  Serial.println(valorentrada1);
  Serial.println(valorentrada2);
  //Efetua as comparações e aciona cada LED de acordo com sua respectiva tensão
  if (valorentrada1 >= 2.5) {
    digitalWrite(saida1, HIGH);
  }
}
```

```
else {  
digitalWrite(saida1, LOW);  
}  
if (valorentrada2 >= 2.5) {  
digitalWrite(saida2, HIGH);  
}  
else {  
digitalWrite(saida2, LOW);  
}  
}
```

A figura 13, apresenta um fluxograma completo da programação.

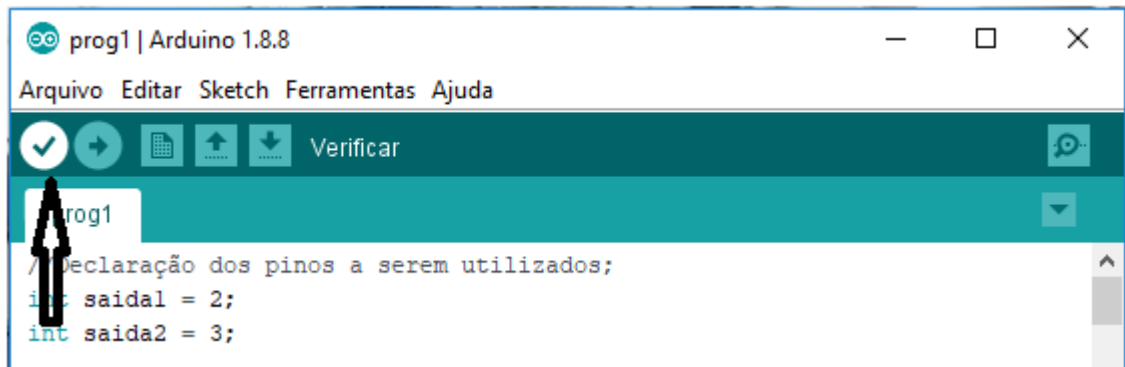
Figura 13. Fluxograma da programação



Fonte: Autoria própria

Com a programação finalizada podemos fazer uma verificação (compilação) no próprio software IDE com o objetivo de verificar se há algum erro no programa, para isto basta acionar o botão mostrado na figura 14.

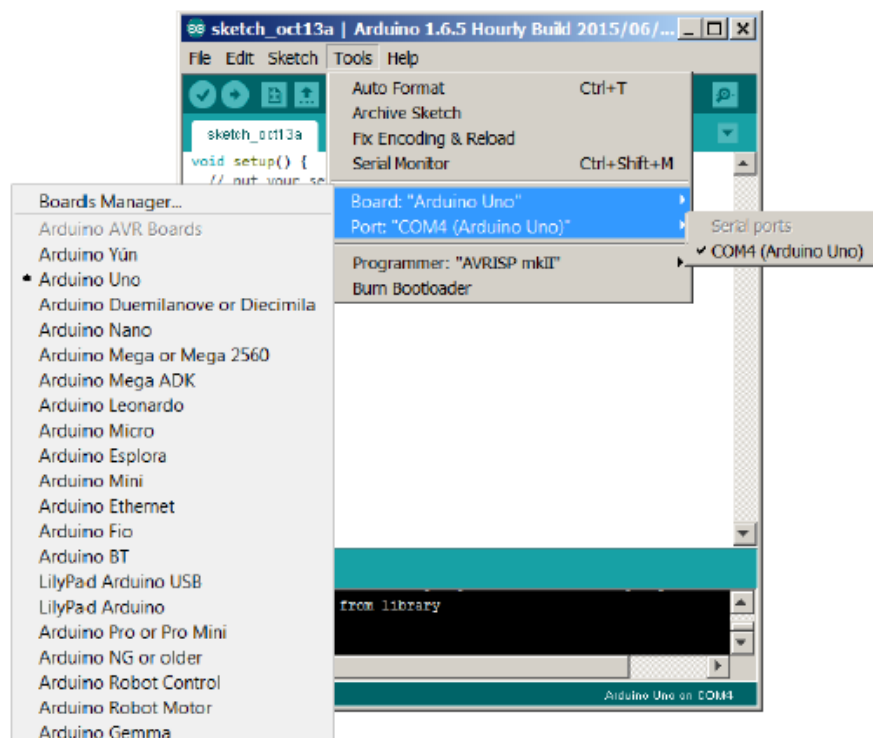
Figura 14. Verificação da programação



Fonte: Print da tela do programa

Para a gravação do programa desenvolvido no software IDE na placa do arduino, conectamos o cabo USB no pc e na placa arduino, clica-se no menu TOOLS/FERRAMENTAS selecionando o item Arduino UNO e verifica-se a porta designada com o termo COM#, conforme figura 15.

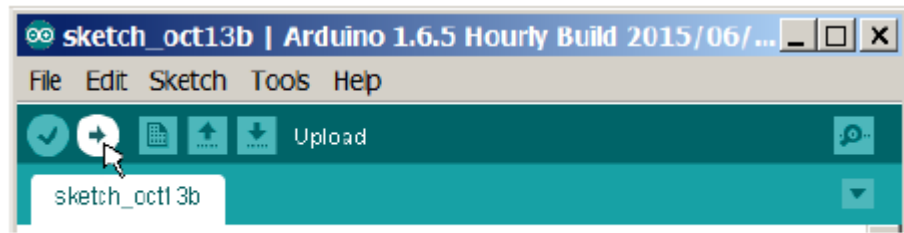
Figura 15. Verificação da porta de comunicação



Fonte: Print da tela do programa

Para enviar o código/programa para o arduino executá-lo, basta clicar no comando indicado na figura 16.

Figura 16. Gravação do código



Fonte: *Print Screen* da janela do programa Arduino.

Fonte: Print da tela do programa

Neste momento os led's da placa arduino irão piscar, sinalizando que a placa está recebendo a programação, após o término da gravação a sua placa arduino está pronta para ser utilizada, agora basta montar os circuitos externos a placa (sensores e dispositivos de saídas).

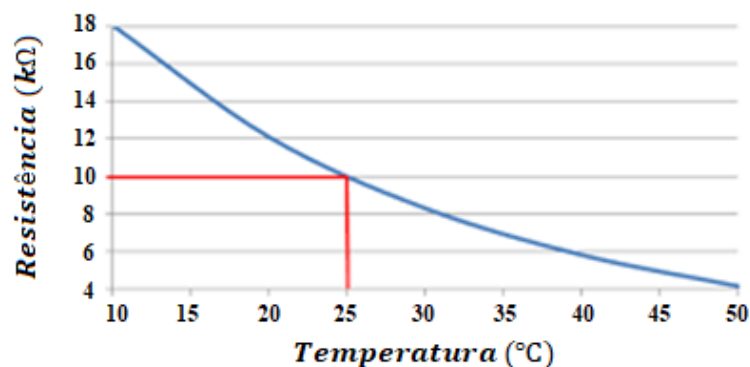
ATIVIDADE 2: Fazendo os cálculos para a utilização do sensor de temperatura

O primeiro sensor que utilizaremos é o sensor de temperatura (NTC), também chamado de Termistor NTC (coeficiente negativo de temperatura) é um resistor sensível a temperatura. Sua resistência elétrica diminui à medida que a temperatura aumenta. Sua curva de temperatura em relação à resistência não é linear e tem um comportamento exponencial, de acordo com a figura 17.

Figura 17: Relação entre resistência e temperatura para o sensor NTC

Temperatura	RESISTÊNCIA (k Ohms)		
25	9,800	10,000	10,200
30	7,874	8,066	8,228
35	6,371	6,545	6,685
40	5,187	5,343	5,463
45	32,185	4,386	32,835
50	25,016	3,620	25,624
55	19,582	3,003	20,138
60	15,429	2,504	15,931
65	12,255	2,098	12,705
70	9,800	1,766	10,200
75	7,874	1,493	8,228
80	6,371	1,268	6,685
85	5,187	1,081	5,463

(a)



(b)

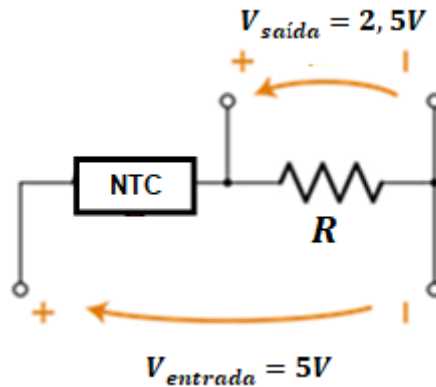
Fonte: (a) Web 1³; (b) Web 2⁴

Em nosso circuito programamos o acionamento para quando a temperatura do ambiente (aumentando) atingir 30°C. Para isso programamos a tensão de saída em 2,5V. A representação do esquema está disposto na figura 18.

³ <http://frioclimatizado.blogspot.com/2016/04/medicao-de-sensores-doscondicionador.html>

⁴ <http://squids.com.br/arduino/index.php/projetos-arduino/projetos-squids/basico/153-projeto-41-usando-o-sensor-de-temperatura-termistor-ntc-10k-com-lcd>

Figura 18: Divisor de tensão com o NTC



Fonte: Os autores

Analise a figura 17 e verifique qual o valor de resistência que corresponde à temperatura de 30°C, registre o valor no quadro 1, abaixo.

Temperatura (°C)	Resistência (kΩ)

IMPORTANTE:

O múltiplo *k* equivale a ordem de grandeza 1000. Assim $1k\Omega = 1000\Omega$.

Agora que você já possui a resistência do sensor, aplique a equação do divisor de tensão e descubra qual o valor do resistor fixo que deve ser ligado em série com o sensor para que ele funcione como projetado.

$$V_{saida} = \left(\frac{R}{R_{NTC} + R} \right) \cdot V_{entrada}$$

Esse é o valor do resistor que será utilizado na montagem do circuito. É importante ressaltar que não existem todos os valores de resistores para comprar, os valores encontrados são chamados de valores comerciais, estes estão dispostos na

figura 19. Quando o resistor calculado para um projeto simples não tem valor comercial podemos utilizar o valor superior mais próximo.

Figura 19: Valores comerciais de resistores fixos

5% Resistores de Filme de Carbono							
Tabela de Valores Comerciais							
1,0 Ω	10 Ω	100 Ω	1 K Ω	10 K Ω	100 K Ω	1 M Ω	10 M Ω
1,1 Ω	11 Ω	110 Ω	1,1 K Ω	11 K Ω	110 K Ω	1,1 M Ω	15 M Ω
1,2 Ω	12 Ω	120 Ω	1,2 K Ω	12 K Ω	120 K Ω	1,2 M Ω	22 M Ω
1,3 Ω	13 Ω	130 Ω	1,3 K Ω	13 K Ω	130 K Ω	1,3 M Ω	
1,5 Ω	15 Ω	150 Ω	1,5 K Ω	15 K Ω	150 K Ω	1,5 M Ω	
1,6 Ω	16 Ω	160 Ω	1,6 K Ω	16 K Ω	160 K Ω	1,6 M Ω	
1,8 Ω	18 Ω	180 Ω	1,8 K Ω	18 K Ω	180 K Ω	1,8 M Ω	
2,0 Ω	20 Ω	200 Ω	2 K Ω	20 K Ω	200 K Ω	2 M Ω	
2,2 Ω	22 Ω	220 Ω	2,2 K Ω	22 K Ω	220 K Ω	2,2 M Ω	
2,4 Ω	24 Ω	240 Ω	2,4 K Ω	24 K Ω	240 K Ω	2,4 M Ω	
2,7 Ω	27 Ω	270 Ω	2,7 K Ω	27 K Ω	270 K Ω	2,7 M Ω	
3,0 Ω	30 Ω	300 Ω	3 K Ω	30 K Ω	300 K Ω	3 M Ω	
3,3 Ω	33 Ω	330 Ω	3,3 K Ω	33 K Ω	330 K Ω	3,3 M Ω	
3,6 Ω	36 Ω	360 Ω	3,6 K Ω	36 K Ω	360 K Ω	3,6 M Ω	
3,9 Ω	39 Ω	390 Ω	3,9 K Ω	39 K Ω	390 K Ω	3,9 M Ω	
4,3 Ω	43 Ω	430 Ω	4,3 K Ω	43 K Ω	430 K Ω	4,3 M Ω	
4,7 Ω	47 Ω	470 Ω	4,7 K Ω	47 K Ω	470 K Ω	4,7 M Ω	
5,1 Ω	51 Ω	510 Ω	5,1 K Ω	51 K Ω	510 K Ω	5,1 M Ω	
5,6 Ω	56 Ω	560 Ω	5,6 K Ω	56 K Ω	560 K Ω	5,6 M Ω	
6,2 Ω	62 Ω	620 Ω	6,2 K Ω	62 K Ω	620 K Ω	6,2 M Ω	
6,8 Ω	68 Ω	680 Ω	6,8 K Ω	68 K Ω	680 K Ω	6,8 M Ω	
7,5 Ω	75 Ω	750 Ω	7,5 K Ω	75 K Ω	750 K Ω	7,5 M Ω	
8,2 Ω	82 Ω	820 Ω	8,2 K Ω	82 K Ω	820 K Ω	8,2 M Ω	
9,1 Ω	91 Ω	910 Ω	9,1 K Ω	91 K Ω	910 K Ω	9,1 M Ω	

Fonte: Web 3⁵

Utilize os valores apresentados na figura 19 e escolha o resistor comercial para ser utilizado em seu projeto.

$$R = \text{_____} \Omega$$

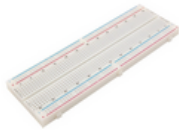





⁵ <https://universoca.wordpress.com/2017/04/16/eletronica-analogica-resistencia-e-resistores/>

ATIVIDADE 3: Montando o circuito com o NTC utilizando Arduino

Objetivo da prática:

Utilizar o Termistor NTC 10k (Sensor de Temperatura) em conjunto com o Arduino a fim de acionar um cooler quando a temperatura atingir 30°C.

Materiais:

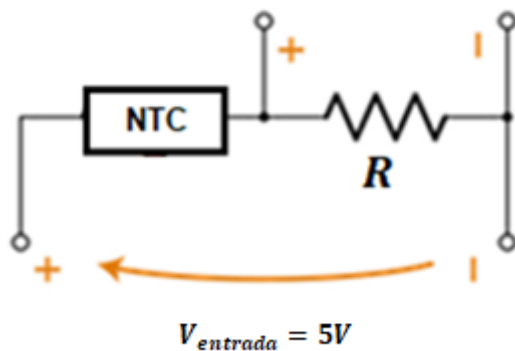
1 - Protoboard 830 pontos	
1 - Kit cabos ligação macho/macho	
1 - Sensor de Temperatura Termistor NTC 10k	
1 - Resistor de 8,2kΩ (Valor encontrado na atividade 2)	
1 - Arduino UNO Original (ou similar)	
1 - 1 Cooler Pc 2510S 5V - 25*10mm	

Procedimento:

Monte o termistor NTC como um divisor de tensão de acordo com o esquema da figura 20:

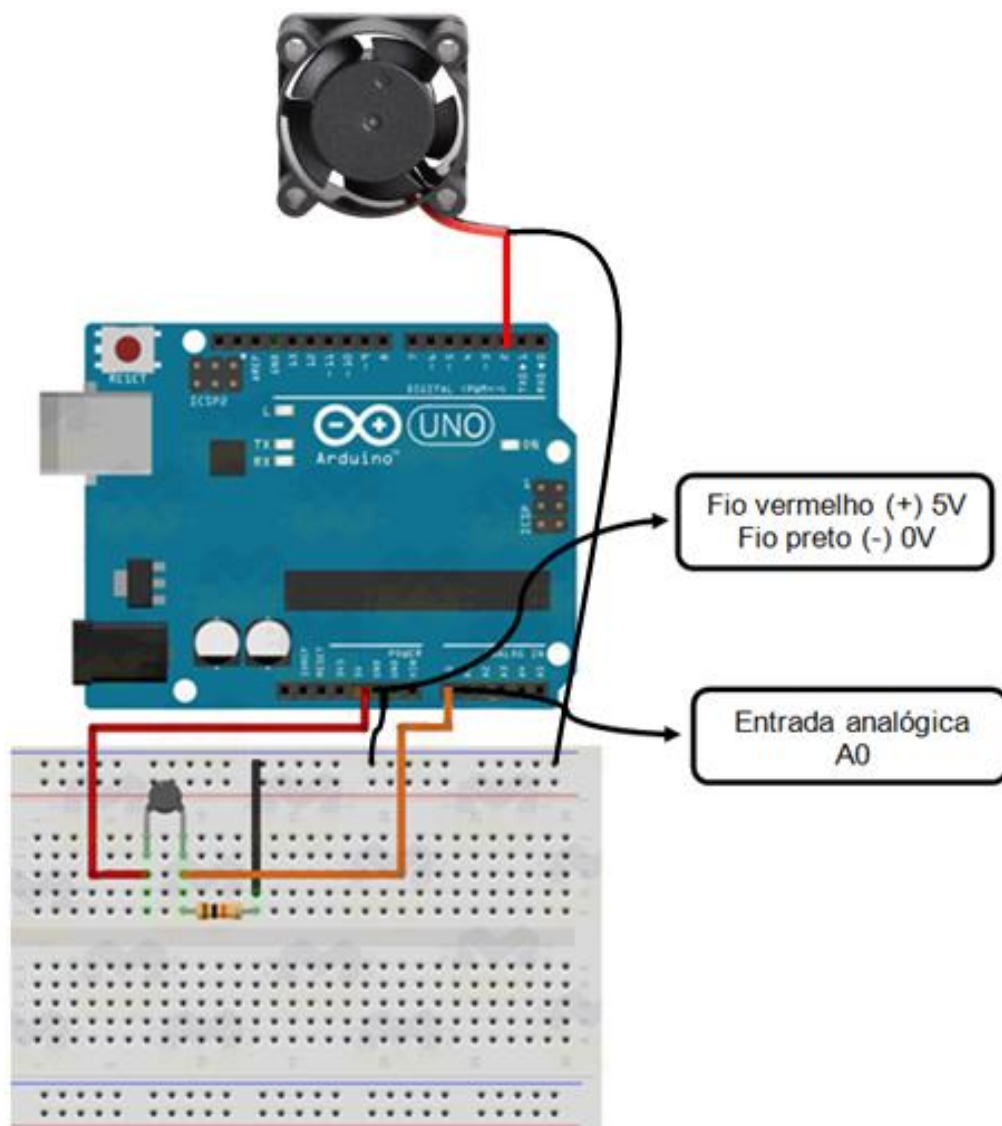
Figura 20: Esquema de montagem para o sensor de temperatura

Entrada analógica A0



Fonte: Os autores

Esquema de montagem do circuito elétrico com Arduino



Fonte: Os autores

Com o circuito montado conecte o cabo do Arduino ao computador. Em seguida aperte levemente o termistor NTC com a ponta dos dedos, apenas para aquecer o componente. Registre o que você observou.

Pesquise e responda

- 1 - O que é um termistor?
- 2 - Que tipos de termistores existem? Em que eles se diferenciam?
- 3 - Para que se utilizam termistores?
- 4 - O que acontece com um termistor NTC se aplicarmos sobre ele um ar quente (com um secador de cabelo por exemplo)?
- 5 - Explique utilizando suas palavras o que acontece no experimento realizado.

ATIVIDADE 4: Fazendo os cálculos para a utilização do sensor de Luz

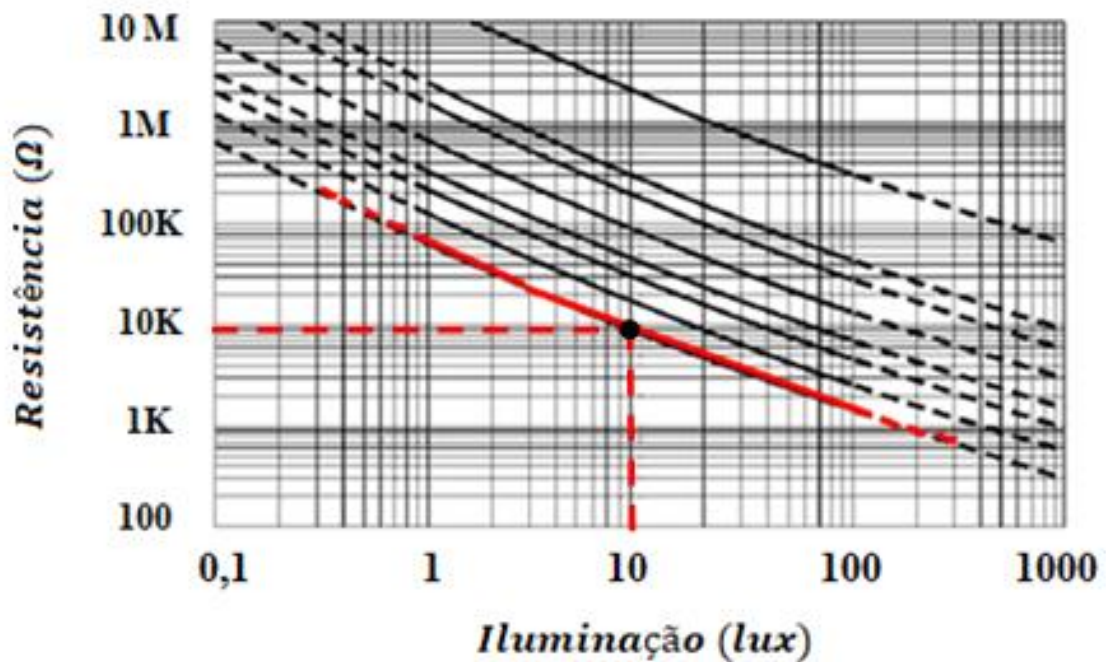
O segundo sensor que utilizaremos é o sensor de Luz conhecido como LDR. LDR, do inglês Light Dependent Resistor ou Resistor Dependente de Luz, é um sensor que varia sua resistência conforme a intensidade de luz. Quanto mais luz, menor sua resistência.

Atenção:

O LDR não fornece uma leitura da quantidade de LUX em um ambiente, somente consegue fornecer a informação se está claro ou escuro.

Sua curva de Lux em relação à resistência não é linear e tem seu comportamento apresentado na figura 21.

Figura 21: Relação entre resistência e luminosidade para o sensor LDR

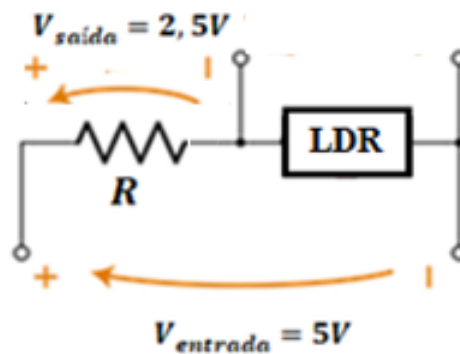


Fonte: Web⁶

⁶ <https://arduinobymyself.blogspot.com/2012/03/arduino-sensoreamento-de-luz-ldr.html>

Em nosso circuito programamos o acionamento para quando a luminosidade atingir 10 lux. Para isso programamos a tensão de saída em 2,5V. A representação do esquema está disposto na figura 22.

Figura 22: Divisor de tensão com o LDR



Fonte: Os autores

Analise a figura 6 e verifique qual o valor de resistência que corresponde à luminosidade de 10lux, registre o valor no quadro 2, abaixo.

Luminosidade (lux)	Resistência (k Ω)

IMPORTANTE:

O múltiplo k equivale a ordem de grandeza 1000. Assim $1k\Omega = 1000\Omega$.

Agora que você já possui a resistência do sensor, aplique a equação do divisor de tensão e descubra qual o valor do resistor fixo **comercial** que deve ser ligado em série com o sensor para que ele funcione como projetado.

$$V_{saida} = \left(\frac{R}{R_{LDR} + R} \right) \cdot V_{entrada}$$

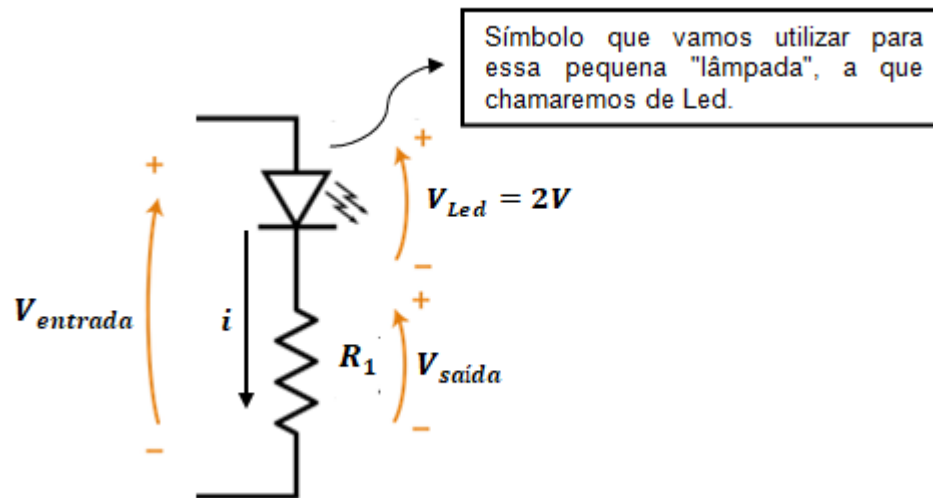
ATIVIDADE 5: Montando o circuito com o LDR utilizando Arduino

Como o LDR varia sua resistência de acordo com a intensidade da luz, iremos utilizar uma porta analógica do Arduino para ler esta variação. Para fazermos isso, montaremos um divisor de tensão com um resistor comercial encontrado na atividade 4.

Nessa atividade, acionaremos um Led quando diminuirmos a luminosidade no sensor LDR, no entanto o Led funciona com 2V e a corrente máxima que ele suporta é de 10mA (especificações do fabricante), se ligarmos 5V da saída digital do arduino direto nele podemos danificá-lo. Então devemos projetar um resistor para ligar em série com o Led e permitir que ele funcione de acordo com as especificações do fabricante.

Observe o desenho esquemático na figura 23.

Figura 23: desenho esquemático do divisor de tensão resistor fixo - Led



Fonte: Os autores

- Qual o valor da corrente máxima que pode circular pelo Led de acordo com o fabricante?

- Qual o valor da corrente que circula pelo resistor fixo?

- Se a tensão de entrada é igual a 5V e a tensão máxima sobre o Led deve ser 2V, quanto de tensão deverá ter sobre o resistor fixo?

- De posse de todos esses dados calcule a resistência fixa que deverá ser utilizada em série com o Led para que o mesmo não sofra nenhum dano.

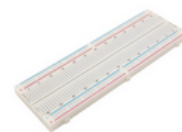
- Verifique se o valor calculado é um valor comercial, caso não seja indique o valor comercial que deve ser utilizado.







Objetivo da prática:

Utilizar o LDR em conjunto com o Arduino a fim de acionar um Led quando a luminosidade ambiente chegar em 10lux..

Materiais:

1 - Protoboard 830 pontos

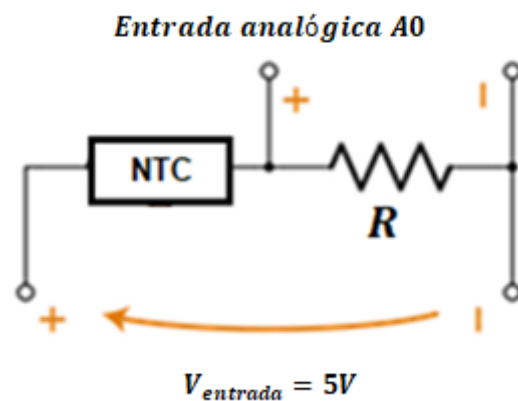


1 - Kit cabos ligação macho / macho	
1 - Led difuso 5mm vermelho	
1 - Resistor de 10kΩ (Valor encontrado na atividade 4)	
1 - Resistor de 330Ω (Valor encontrado no início da atividade 5)	
1 - Arduino UNO Original (ou similar)	
1 - Sensor de luminosidade LDR 10mm	

Procedimento:

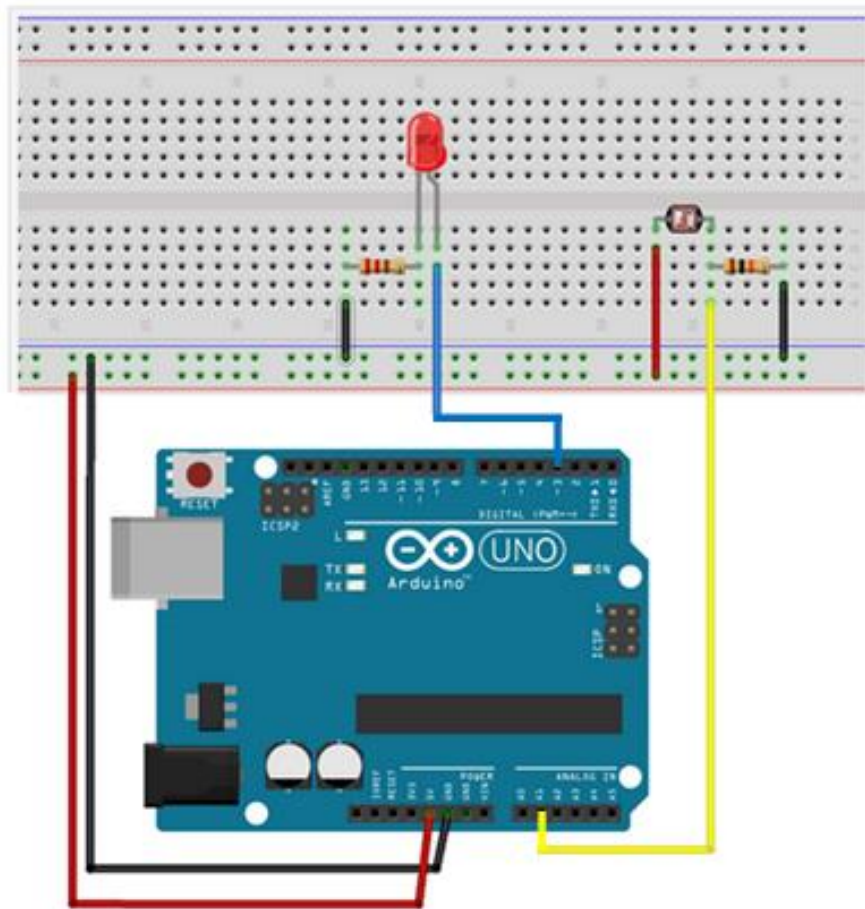
Monte o LDR como um divisor de tensão de acordo com o esquema da figura 24:

Figura 24: Divisor de tensão



Fonte: Os autores

Figura 25: Esquema de montagem do circuito elétrico com Arduino



Fonte: Os autores

Com o circuito montado conecte o cabo do Arduino ao computador. Em seguida com a ponta do dedo tampe o sensor LDR. Registre o que você observou.

Pesquise e responda

- 1 - O que é LDR?
- 2 - O que significa a sigla LDR?
- 3 - Onde se utiliza o LDR?
- 4 - Explique utilizando suas palavras o que acontece no experimento.

CONSIDERAÇÕES SOBRE O PRODUTO

A construção desse produto se resume em uma tentativa de um novo olhar para as práticas pedagógicas, entendendo o aprendiz como um ser integrado ao processo de ensino e aprendizagem, considerando suas vivências e experiências. Nesse viés, o presente trabalho utilizou a Robótica Educacional para promover um ambiente de aprendizagem diferente do tradicional, visto que as TIC apresentam um grande potencial junto ao público adolescente.

Espera-se que o roteiro de ensino aqui apresentado provoque um encurtamento das distâncias entre professor e aluno, oportunizando o diálogo, a socialização e o levantamento de hipóteses. Pode-se destacar também a importância da experimentação no ensino de Física aqui demonstrado que, devidamente trabalhado, mostra-se um poderoso aliado na construção do entendimento do aluno, dando sentido e significado à teoria que, muitas vezes, carregadas de abstração, impedem o aprendiz de relacioná-la ao seu dia-a-dia.

A aplicação desse produto em uma turma do 9º ano do Ensino Fundamental II mostrou um resultado muito positivo, tanto com relação à motivação e participação dos alunos nas atividades propostas, quanto nas relações estabelecidas por eles com situações cotidianas.

REFERÊNCIAS

ARDUINO. Disponível em <https://www.arduino.cc/> acesso em: nov. 2018.

BATISTA, M. C., BATISTA, D. C., **Física experimental**. Maringá: UniCesumar,.168 p. “Graduação - EaD”. 2016.

BATISTA, M. C., FUSINATO P. A. **Eletricidade básica: caderno de atividades experimentais**. Ed. Massoni, Maringá 2014.

CAVALCANTE, M. A.; TAVOLARO, C. R. C.; MOLISANI, E. **Física com Arduino para iniciantes**. Revista Brasileira de Ensino de Física, São Paulo, v. 33, n. 4, p. 4503, 2011.

GROOVER, M. P. **Robótica: Tecnologia e Programação**. McGraw-Hill, 1.998.

HALLIDAY, D.; RESNICK, R. **Fundamentos de Física. Vol. 3: Eletromagnetismo**. Rio de Janeiro: Livros Técnicos e Científicos, 1991.

MOREIRA, M. A., MASINI, E. F. S. **Aprendizagem Significativa: A Teoria de David Ausubel**. São Paulo: Centauro, 2001.