

PPGTER/DES.20.2020.MAT

Oficinas de Arduíno: Manual para Professores de Lógica de Programação

Autores

Anderson José Lauer
lauer_anderson@hotmail.com

Andre Zanki Cordenonsi
andrezc@inf.ufsm.br



Versão 1.0
Status: Final
Distribuição: Externa
FEVEREIRO 2020



2020 PPGTER – Programa de Pós-Graduação em Tecnologias Educacionais em Rede

Atribuição-Não Comercial 4.0 Internacional (CC BY-NC 4.0)

Você tem o direito de compartilhar, copiar e redistribuir o material em qualquer suporte ou formato; adaptar, remixar, transformar, e criar a partir do material, de acordo com o seguinte: você deve dar o crédito apropriado, prover um link para a licença e indicar se mudanças forem feitas. Você deve fazê-lo em qualquer circunstância razoável, mas de nenhuma maneira que sugira que o licenciante apoia você ou seu uso. Você não pode usar o material para fins comerciais.

PPGTER

PROGRAMA DE PÓS-GRADUAÇÃO EM TECNOLOGIAS EDUCACIONAIS EM REDE
UNIVERSIDADE FEDERAL DE SANTA MARIA - CENTRO DE EDUCAÇÃO

Editoria Técnica do PPGTER

Universidade Federal de Santa Maria

Av. Roraima n. 1000

Centro de Educação, Prédio 16, sala 3146

Santa Maria – RS – CEP 97105-900

Fone / FAX: 55 3220 9414

ppgter@ufsm.br

edtec.ppgter@gmail.com

ISSN: 2675-0309

Relatórios Técnicos do Programa de Pós-Graduação em Tecnologias Educacionais em Rede / Programa de Pós-Graduação em Tecnologias Educacionais em Rede, Universidade Federal de Santa Maria. – Vol. 2. n. 1 (2020) Jan/Jun. – Santa Maria: PPGTER/UFSM, 2020.

Periodicidade semestral.

1. Tecnologia Educacional. 2. Desenvolvimento de Tecnologias Educacionais. 3. Gestão de Tecnologias Educacionais. I. Universidade Federal de Santa Maria. Programa de Pós-Graduação em Tecnologias Educacionais em Rede.

Como citar este relatório:

LAUER, A.J.; CORDENONSI, A.Z. **Oficinas de Arduino**: Manual para Professores de Lógica de Programação. Santa Maria: 2020. Relatórios Técnicos do Programa de Pós-Graduação em Tecnologias Educacionais em Rede, v. 2., n.1. Disponível em: <https://www.ufsm.br/cursos/pos-graduacao/santa-maria/ppgter/ppgter-des-20-2020-mat/>

RESUMO

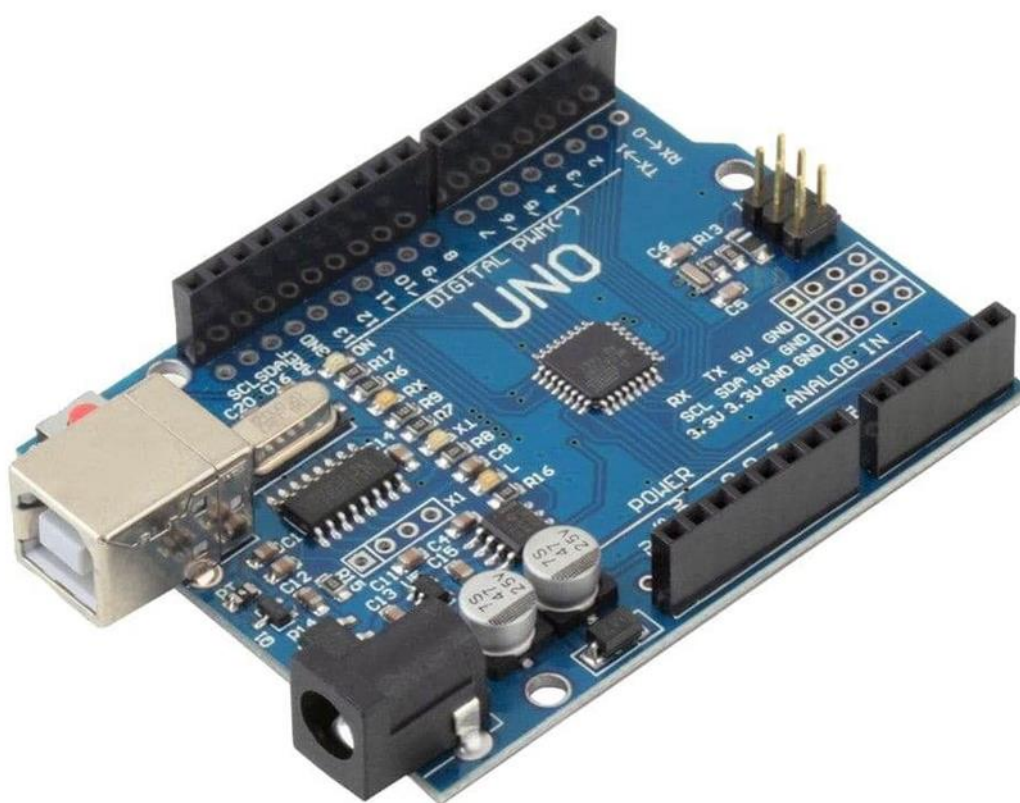
Este relatório técnico apresenta uma série de projetos no formato de oficinas para o ensino de lógica de programação com o uso da plataforma Arduino. Buscou-se verificar como o Arduino pode contribuir no processo de ensino e aprendizagem de programação de computadores na construção de protótipos eletrônicos por meio dos três momentos dialógicos problematizadores. Esse material foi elaborado como produto final da dissertação de mestrado de Anderson José Lauer no Programa de Pós-Graduação em Tecnologias Educacionais em Rede da Universidade Federal de Santa Maria-UFSM.

APÊNDICE A

OFICINAS DE ARDUÍNO

OFICINAS DE ARDUINO

Manual para professores



ANDERSON JOSÉ LAUER
Andre Zanki Cordenonsi

Versão 1.0 - Fevereiro 2020

Anderson José Lauer

Anderson José Lauer possui Curso Técnico em Informática pelo Instituto Federal de Educação, Ciência e Tecnologia Farroupilha – Campus Santo Augusto, graduação em Licenciatura em Computação pelo Instituto Federal de Educação, Ciência e Tecnologia Farroupilha – Campus Santo Augusto, especialização em Informática Aplicada na Educação pelo Instituto Federal de Educação, Ciência e Tecnologia Farroupilha – Campus Santo Augusto. Atualmente cursa o Mestrado Profissional em Tecnologias Educacionais em Rede pela Universidade Federal de Santa Maria. É Técnico em Informática, atuando na área de manutenção de computadores e sistemas na Prefeitura Municipal de Coronel Bicaco.

Contato: lauer_anderson@hotmail.com

Andre Zanki Cordenonsi

Andre Zanki Cordenonsi possui graduação em Informática pela Universidade Federal de Santa Maria (1996), mestrado em Ciências da Computação pela Universidade Federal do Rio Grande do Sul (2000) e doutorado em Informática na Educação pela Universidade Federal do Rio Grande do Sul (2008). Atualmente, é professor associado da Universidade Federal de Santa Maria, atuando nos Cursos de Graduação em Arquivologia e Sistemas de Informação. Também é pesquisador do Mestrado Profissional em Patrimônio Cultural e do Mestrado Profissional em Tecnologias Educacionais em Rede, ambos da UFSM, atuando na área de gestão de documentos eletrônicos e desenvolvimento de tecnologias educacionais para o ensino superior.

Contato: andrezc@inf.ufsm.br

Limites do Artefato

Este documento é resultado de um projeto de pesquisa realizada ao longo do curso de Mestrado Profissional em Tecnologias Educacionais em Rede na Universidade Federal de Santa Maria (2018-2020), de autoria de Anderson José Lauer sob a orientação do Professor Doutor André Zanki Cordenonsi. O objetivo do artefato foi desenvolver oficinas de Arduino para alunos do Curso Técnico em Informática do Instituto Federal Farroupilha – Campus Santo Augusto, no entanto, ressalta-se que este artefato não foi aprovado ou referendado pelo curso ou instituição supracitada.



Esta obra está licenciada com uma Licença Creative Commons - Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional.

LISTA DE FIGURAS

Figura 1 – Folha com o circuito do Desafio Inicial para o Projeto 01.....	9
Figura 2 – Folha com o circuito do Desafio Inicial para o Projeto 02.....	10
Figura 3 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 02	11
Figura 4 – Folha com o circuito do Desafio Inicial para o Projeto 03.....	13
Figura 5 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 03	14
Figura 6 – Folha com o circuito do Desafio Inicial para o Projeto 04.....	16
Figura 7 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 04	17
Figura 8 – Folha com o circuito do Desafio Inicial para o Projeto 05.....	19
Figura 9 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 05	20
Figura 10 – Folha com o circuito do Desafio Inicial para o Projeto 06.....	22
Figura 11 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 06	23
Figura 12 – Folha com o circuito da MSEM do Projeto Final	25

SUMÁRIO

PREFÁCIO	5
1 PLATAFORMA ARDUINO	6
2 OFICINA 1 : APRESENTAÇÃO E PROJETOS 01 E 02	8
2.1 Apresentação	8
2.2 Projeto 1: Funcionamento do LED e Temporização	8
2.3 Projeto 2: Sinalizador de Garagem	10
3 OFICINA 2: PROJETO 3	13
3.1 Projeto 03: Controlando LED utilizando potenciômetro	13
4 OFICINA 3: PROJETO 4	16
4.1 Projeto 04: Controlando um motor utilizando potenciômetro	16
5 OFICINA 4: PROJETO 5	18
5.1 Projeto 05: Detectando luminosidade com sensor de luz	18
6 OFICINA 5: PROJETO 6	21
6.1 Projeto 06: Detecção de presença com sensor infravermelho	21
7 OFICINA 6: PROJETO FINAL	23
REFERÊNCIAS	27

PREFÁCIO

Este material foi elaborado como produto final da dissertação intitulada “Plataforma Arduino: uma alternativa para potencializar a aprendizagem de programação de computadores” do Programa de Pós-Graduação em Tecnologias Educacionais em Rede da Universidade Federal de Santa Maria-UFSM.

Esta pesquisa buscou verificar como a Plataforma Arduino pode contribuir no processo de ensino e aprendizagem de programação de computadores na construção de protótipos eletrônicos por meio dos três momentos dialógicos problematizadores com o objetivo de analisar as potencialidades desta ferramenta.

Para fazer essa análise e posteriormente a avaliação foram desenvolvidas atividades práticas em formato de oficinas, a qual culminou na criação desse manual que pode ser utilizado por professores que atuam nas áreas de programação de computadores, constituindo-se como uma metodologia para ser aplicada no ensino e aprendizagem de programação.

Este conteúdo foi construído no formato de manual para professores, apresentando os objetivos de cada oficina, os momentos pedagógicos, os materiais utilizados e as respostas aos desafios que foram apresentados aos alunos. É importante salientar que as respostas apresentadas não são únicas ou invariáveis; é possível que outras soluções sejam apresentadas pelos alunos. Os professores que utilizarem este manual podem fazer alterações nas oficinas para poderem adaptar ao contexto de aplicação.

1 PLATAFORMA ARDUINO

O Arduino teve seu surgimento na Itália em 2005 com o objetivo de ser utilizado em projetos escolares para prototipagem de sistemas, buscando uma ferramenta de menor custo. Esta plataforma consiste em “[...] um computador minúsculo que você, pode programar para processar entradas e saídas entre o dispositivo e os componentes externos que conectar a ele.” (MCROBERTS 2015, p. 24).

Para entrada das informações, o Arduino pode utilizar alguns dispositivos como sensores, dentre eles: sensores de temperatura, luminosidade, distância, aceleração, inclinação, pressão, movimento, umidade. Para saída das informações, são utilizados dispositivos chamados atuadores, que são responsáveis por receberem as informações do Arduino e gerar um comando para ativar um elemento de controle, como exemplos: atuadores eletrônicos, hidráulicos, elétricos, motores, bombas que geram uma determinada ação, etc.

O processamento dessas informações de entrada, que processadas geram uma informação como resultado de saída, acontece na placa Arduino, pois esta possui um processador e uma memória capaz de manipular estas informações. Existem diversas placas de Arduino no mercado, o que as diferem são suas capacidades de processamento, memória e a quantidade de pinos de entrada e saída para a conexão dos dispositivos.

Em paralelo, existe uma infinidade de placas que também podem ser acopladas ao Arduino permitindo novas funcionalidades e expandindo suas capacidades. Estas placas são conhecidas como escudos (*shields*). Como exemplos temos *shields ethernet* para desenvolver projetos conectados, *shields* para controle de motores, utilização de sons, entre outras infinidades.

Nas oficinas propostas neste artefato, utilizamos os Três Momentos Pedagógicos propostos por Abegg, De Bastos e Mallmann (2001 apud DE BASTOS E MAZZARDO, 2005), que tem origem na educação dialógico-problematizadora (Freire, 1987), propondo uma visão de educação que tem por objetivo dar voz a educandos e educadores em sala de aula através do diálogo com base no contexto vivido por essas pessoas e de suas visões de conhecimento. Essa educação busca

envolver os alunos fazendo com que busquem respostas para a resolução de problemas.

Esta prática se organiza da seguinte forma:

- **Desafio Inicial - DI:** Nesse momento, o professor apresenta aos alunos o desafio, no qual são instigados e motivados a buscar informações referentes ao problema e ficar envolvido na resolução deste;
- **Melhor Solução Educacional no Momento - MSEM:** Caracterizado como o principal momento, aonde o professor conduz os estudos acerca do desafio, buscando trazer conhecimentos científicos e relacionando com conhecimentos já desenvolvidos pelos alunos, para propor a solução ao desafio e posteriormente propor um desafio maior;
- **Desafio mais Amplo - DA:** Neste momento, é proposto um desafio maior, que possui relação com o desafio inicial buscando colocar em pratica a solução do desafio e através de complexidade do desafio inicial fazer com que o aluno evolua em seus conhecimentos a respeito do tema.

As oficinas propostas aqui podem ser utilizadas para alunos de cursos técnicos que já tenham tido alguma disciplina vinculada à programação, o que não impede que as mesmas possam ser utilizadas em disciplinas de introdução à programação, desde que realizadas as adaptações necessárias. Propõe-se que as mesmas sejam realizadas em seis encontros de três horas cada, mas é possível realizar as mesmas com diferentes tempos e cargas didáticas.

2 OFICINA 1 : APRESENTAÇÃO E PROJETOS 01 E 02

Esta oficina é dividida em três momentos: Apresentação, Projeto 01 e Projeto 02.

2.1 APRESENTAÇÃO

Objetivos: apresentação da plataforma Arduino, com o intuito de familiarizar os estudantes com seus componentes. Também foi explicada a linguagem de programação utilizada pelo sistema, sua sintaxe, variáveis, funções e comandos básicos da linguagem. Essa linguagem segue uma sequência lógica aonde primeiramente são declaradas as variáveis de acordo com as entradas digitais e analógicas em que estão conectados os componentes, em um segundo momento são declaradas se essas variáveis vão receber ou enviar determinada informação ao Arduino e, posteriormente, é realizada toda a programação para o funcionamento de um determinado projeto, aonde são definidas as ações que serão executadas.

Materiais:

1. um kit da plataforma Arduino;
2. um notebook/desktop para cada aluno ou grupo de alunos.

2.2 PROJETO 1: FUNCIONAMENTO DO LED E TEMPORIZAÇÃO

Objetivos: apresentação do funcionamento de um LED que liga e desliga em um intervalo de tempo.

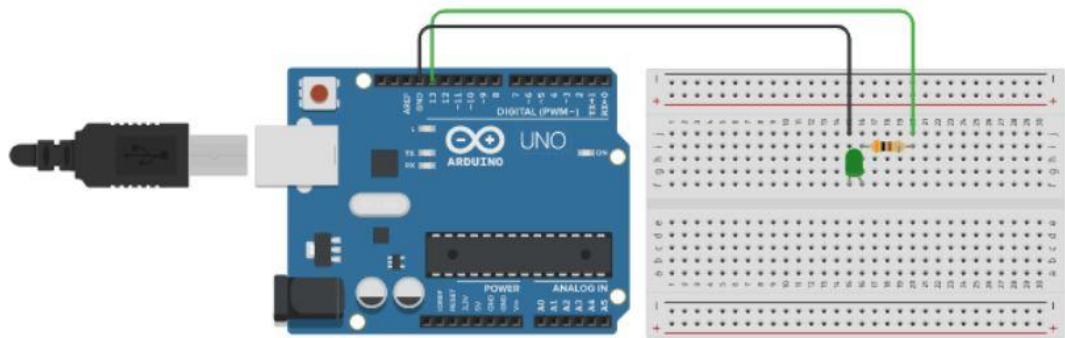
Materiais:

1. um kit da plataforma Arduino;
2. um notebook/desktop para cada aluno ou grupo de alunos;
3. uma protoboard;
4. um cabo USB;
5. fios jumper;
6. um resistor;

7. um LED.

Materiais didáticos:

Figura 1 – Folha com o circuito do Desafio Inicial para o Projeto 01



Fonte: do autor.

Momentos didáticos:

DI: Como ligar e desligar um LED durante um determinado período de tempo?

MSEM: Conceituação e resolução de como ligar e desligar um LED

DA: Discussão sobre aplicações práticas do circuito.

MSEM do Projeto 01:

```
const int pinoLed = 13; //PINO DIGITAL UTILIZADO PELO LED
void setup() {
    pinMode(pinoLed, OUTPUT); //DEFINE O PINO COMO SAÍDA
}
void loop() {
    digitalWrite(pinoLed, HIGH); //LIGA O LED
    delay(1000); //DEFINE O TEMPO DE ESPERA
    digitalWrite(pinoLed, LOW); //DESLIGA O LED
    delay(1000); //DEFINE O TEMPO DE ESPERA
}
```

DA do Projeto 01:

Entre as aplicações práticas do Projeto 01, pode-se citar interruptores de emergência que, ao serem acionados, emitem uma luz piscante indicando emergência ou atenção.

2.3 PROJETO 2: SINALIZADOR DE GARAGEM

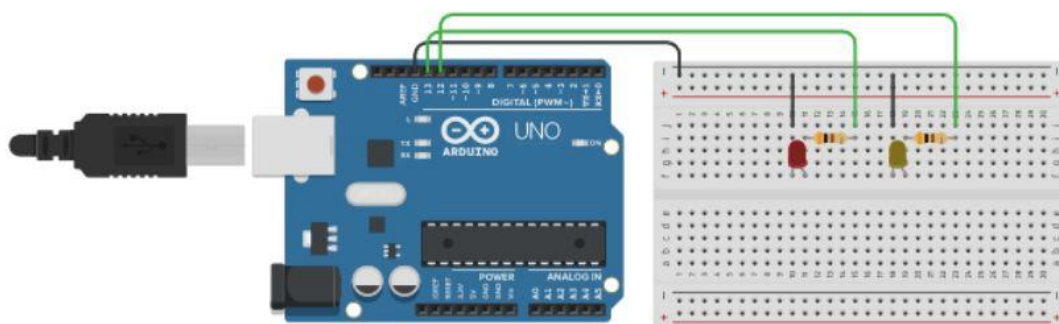
Objetivos: apresentação do funcionamento de dois LEDs simultâneos, simulando um Sinalizador de Garagem.

Materiais:

1. um kit da plataforma Arduino;
2. um notebook/desktop para cada aluno ou grupo de alunos;
3. uma protoboard;
4. um cabo USB;
5. fios jumper;
6. dois resistores;
7. dois LEDs.

Materiais didáticos:

Figura 2 – Folha com o circuito do Desafio Inicial para o Projeto 02



Fonte: do autor.

Momentos didáticos:

DI: Como desenvolver um sinalizador de garagem?

MSEM: Conceituação e resolução de um sinalizador de garagem.

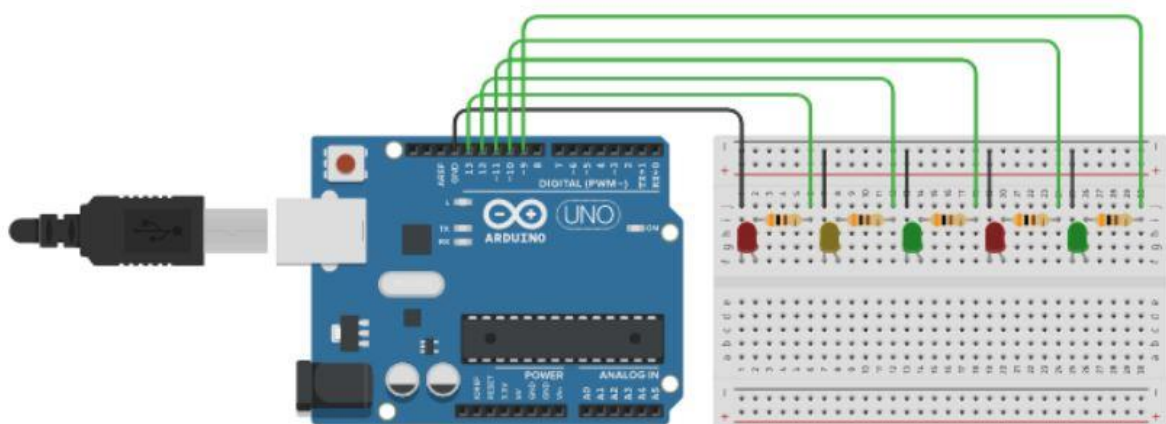
DA: Como construir um semáforo de carros e pedestres utilizando LEDs?

MSEM do Projeto 02:

```
const int pinoLedV = 13; //DEFINE PINO LED VERMELHO
const int pinoLedA = 12; //DEFINE PINO LED AMARELO
void setup() {
    pinMode(pinoLedV, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinoLedA, OUTPUT); //DEFINE O PINO COMO SAÍDA
}
void loop() {
    digitalWrite(pinoLedV, HIGH); //LIGA O LED VERMELHO
    digitalWrite(pinoLedA, LOW); //DESLIGA O LED AMARELO
    delay(500); //DEFINE O TEMPO PARA QUE UM LED APAGUE E OUTRA ACENDA
    digitalWrite(pinoLedA, HIGH); //LIGA O LED AMARELO
    digitalWrite(pinoLedV, LOW); //DESLIGA O LED VERMELHO
    delay(500); //DEFINE O TEMPO PARA QUE UM LED APAGUE E OUTRA ACENDA
}
```

DA do Projeto 02:

Figura 3 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 02



Fonte: do autor.

Código com a resposta do Desafio Mais Amplo do Projeto 02:

```
//DEFINE AS VARIÁVEIS DOS PINOS QUE ESTÃO CONECTADOS AOS LEDS
const int carroVermelho = 13;
const int carroAmarelo = 12;
const int carroVerde = 11;
const int pedestreVermelho = 10;
const int pedestreVerde = 9;
void setup() {
    //DEFINE AS VARIÁVEIS DOS LEDS COMO SAÍDA
    pinMode(carroVermelho, OUTPUT);
    pinMode(carroAmarelo, OUTPUT);
    pinMode(carroVerde, OUTPUT);
    pinMode(pedestreVermelho, OUTPUT);
    pinMode(pedestreVerde, OUTPUT);
}
void loop() {
    //DEFINE A SEQUÊNCIA, SITUAÇÃO DOS LEDS E TEMPO PARA QUE FUNCIONE
    COMO UM SEMÁFORO
    digitalWrite(carroVerde, HIGH);
    digitalWrite(pedestreVermelho, HIGH);
    delay(5000);
    digitalWrite(carroVerde, LOW);
    digitalWrite(carroAmarelo, HIGH);
    delay(3000);
    digitalWrite(carroAmarelo, LOW);
    digitalWrite(carroVermelho, HIGH);
    digitalWrite(pedestreVermelho, LOW);
    digitalWrite(pedestreVerde, HIGH);
    delay(5000);
    digitalWrite(carroVermelho, LOW);
    digitalWrite(pedestreVerde, LOW);
}
```


3 OFICINA 2: PROJETO 3

Esta oficina apresenta o Projeto 03, que utiliza o Arduino com um potenciômetro.

3.1 PROJETO 03: CONTROLANDO LED UTILIZANDO POTENCIÔMETRO

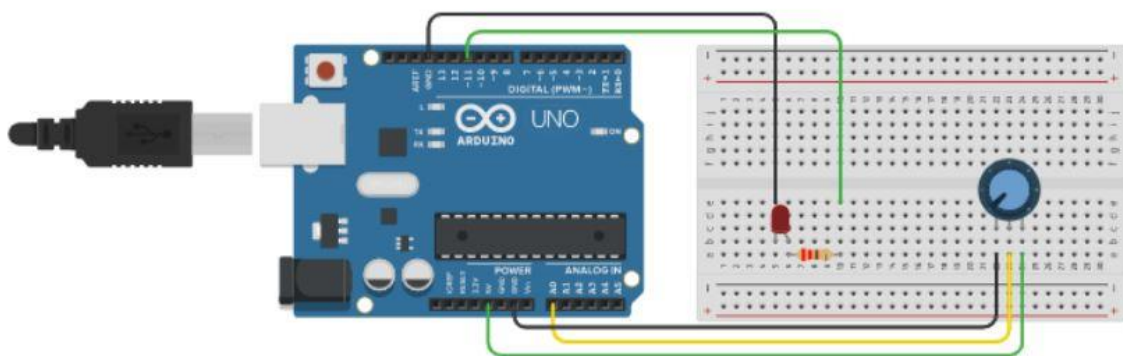
Objetivos: Apresentação e funcionamento de um LED cuja intensidade é controlada através de um potenciômetro.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;
4. fios jumper;
5. um resistor;
6. um LED;
7. um potenciômetro.

Materiais didáticos:

Figura 4 – Folha com o circuito do Desafio Inicial para o Projeto 03



Fonte: do autor.

Momentos didáticos:

DI: Como controlar a potência de um LED utilizando potenciômetro?

MSEM: Conceituação e resolução de como controlar a potência de um LED utilizando potenciômetro.

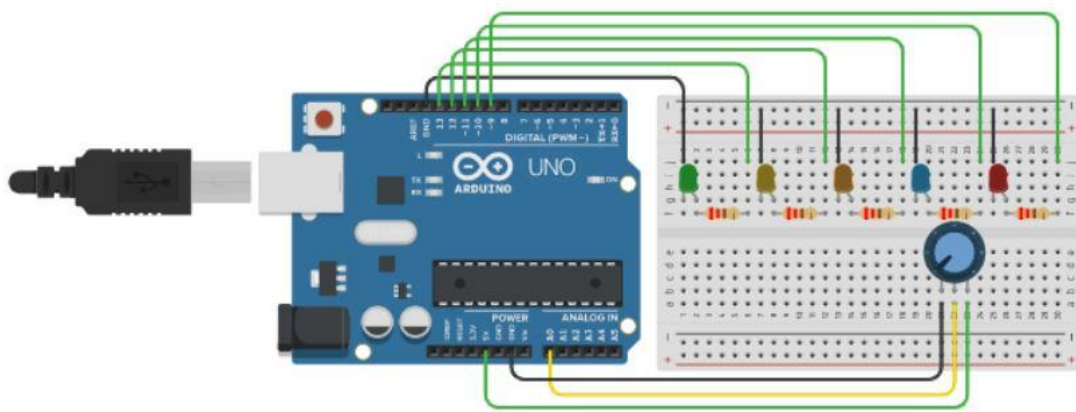
DA: Como ligar LEDs em sequência utilizando potenciômetro?

MSEM do Projeto 03:

```
const int potPin = A0; //DEFINE O PINO ANALÓGICO DO POTENCIÔMETRO
const int ledPin = 11; //DEFINE O PINO DO LED
float valPot = 0; //VARIÁVEL ARMAZENA O VALOR DO POTENCIÔMETRO
void setup() {
    pinMode(ledPin, OUTPUT); //DEFINE O PINO COMO SAIDA
    pinMode(potPin, INPUT); //DEFINE O PINO COMO ENTRADA
}
void loop() {
    valPot = analogRead(potPin);
    //FAZ A LEITURA ANALÓGICA DO POTENCIÔMETRO
    valPot = map(valPot, 0, 1023, 0, 255);
    //FAZ O MAPEAMENTO DO POTENCIÔMETRO PARA TRANSFORMAR EM VALORES
    DIGITAIS PWM
    analogWrite(ledPin, valPot);
    //ACIONA O LED CONFORME VALOR DO POTENCIÔMETRO
}
```

DA do Projeto 03:

Figura 5 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 03



Fonte: do autor.

Código com a resposta do Desafio Mais Amplo do Projeto 03:

```
const int potPin = A0;//DEFINE O PINO ANALÓGICO DO POTENCIÔMETRO
const int quantLeds = 5;//DEFINE A QUANTIDADE DE LEDS
int ledPins[] = {13,12,11,10,9};
    //DEFINE EM UM VETOR OS PINOS QUE OS LEDS ESTÃO LIGADOS
void setup()
{
    for(int i = 0;i<quantLeds;i++)//DEFINE OS PINOS DOS LEDS COMO SAIDA
    {
        pinMode(ledPins[i],OUTPUT);
    }
}
void loop () {
    int valPoten = analogRead(potPin);//FAZ LEITURA DO POTENCIÔMETRO
    int ledNivel = map(valPoten,0,1023,0,quantLeds);
    //FAZ O MAPEAMENTO DOS LEDS PARA O POTENCIÔMETRO
    for(int i = 0;i<quantLeds;i++)//PERCORRE O VETOR DE LEDS
    {
        if(i<ledNivel)
        {
            digitalWrite(ledPins[i],HIGH);//ACENDE O LED
        }else{
            digitalWrite(ledPins[i],LOW);//DESLIGA O LED
        }
    }
}
```

4 OFICINA 3: PROJETO 4

Esta oficina apresenta o Projeto 04, que utiliza o Arduino com um potenciômetro controlando um motor.

4.1 PROJETO 04: CONTROLANDO UM MOTOR UTILIZANDO POTENCIÔMETRO

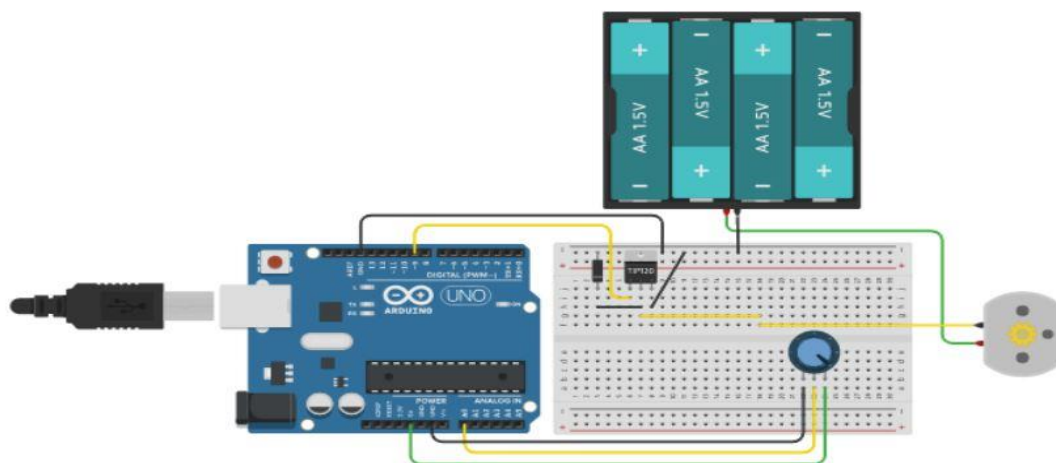
Objetivos: Apresentação e funcionamento de um motor cuja velocidade é controlada através de um potenciômetro.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;
4. fios jumper;
5. um motor;
6. um transistor;
7. um diodo;
8. um potenciômetro;
9. uma fonte de energia externa.

Materiais didáticos:

Figura 6 – Folha com o circuito do Desafio Inicial para o Projeto 04



Fonte: do autor.

Momentos didáticos:

DI: Como controlar a velocidade de um motor utilizando potenciômetro?

MSEM: Conceituação e resolução de como controlar a velocidade de um motor utilizando potenciômetro.

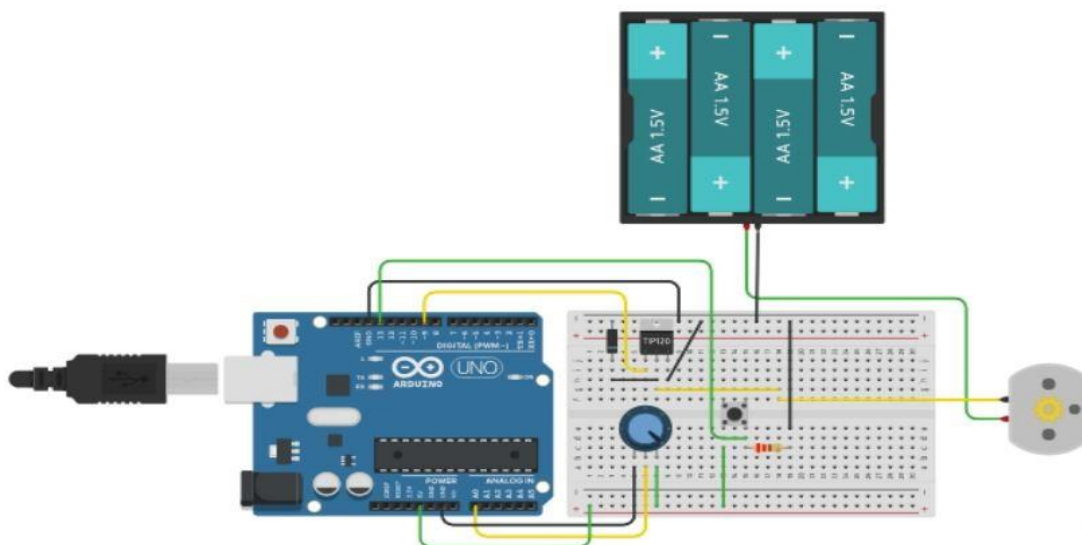
DA: Como controlar a velocidade, ligar e desligar um motor através de um botão?

MSEM do Projeto 04:

```
const int pot = A0; //DEFINE PINO QUE ESTÁ CONECTADO O POTENCIÔMETRO
const int transistor = 9; //DEFINE PINO QUE ESTÁ CONECTADO O TRANSISTOR
int valPot = 0; //ARMAZENA O VALOR DO POTENCIÔMETRO
void setup() {
    pinMode(pot, INPUT); //DEFINE O PINO COMO ENTRADA
    pinMode(transistor, OUTPUT); //DEFINE O PINO COMO SAÍDA
}
void loop() {
    valPot = analogRead(pot); //VARIÁVEL RECEBE VALOR DO POTENCIÔMETRO
    valPot = map(valPot, 0, 1023, 0, 255); //CONVERTE O VALOR DO POTENCIÔMETRO
    //EM VALORES PWM
    analogWrite(transistor, valPot); //ENVIA O VALOR DO POTENCIÔMETRO AO
    //MOTOR
}
```

DA do Projeto 04:

Figura 7 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 04



Fonte: do autor.

Código com a resposta do Desafio Mais Amplo do Projeto 04

```
const int pot = A0; //DEFINE PINO QUE ESTÁ CONECTADO O POTENCIÔMETRO
const int transistor = 9; //DEFINE PINO QUE ESTÁ CONECTADO O TRANSISTOR
int valPot = 0; //ARMAZENA O VALOR DO POTENCIÔMETRO
const int botao = 13;

void setup() {
    pinMode(botao, INPUT); //DEFINE O PINO COMO ENTRADA
    pinMode(pot, INPUT); //DEFINE O PINO COMO ENTRADA
    pinMode(transistor, OUTPUT); //DEFINE O PINO COMO SAÍDA
}

void loop() {
    if(digitalRead(botao) == HIGH) { //ATIVA O VALOR DO POTENCIÔMETRO
        valPot = analogRead(pot);
        //VARIÁVEL RECEBE VALOR DO POTENCIÔMETRO
        valPot = map(valPot, 0, 1023, 0, 255);
        //CONVERTE O VALOR DO POTENCIÔMETRO EM VALORES PWM
        analogWrite(transistor, valPot);
        //ENVIA O VALOR DO POTENCIÔMETRO AO MOTOR
    }
}
```

5 OFICINA 4: PROJETO 5

Esta oficina apresenta o Projeto 05, que utiliza o Arduino para detectar luminosidade com sensores de luz.

5.1 PROJETO 05: DETECTANDO LUMINOSIDADE COM SENSOR DE LUZ

Objetivos: apresentação e funcionamento de um fotoresistor que, ao não identificar a luz, o LED acende.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;


```

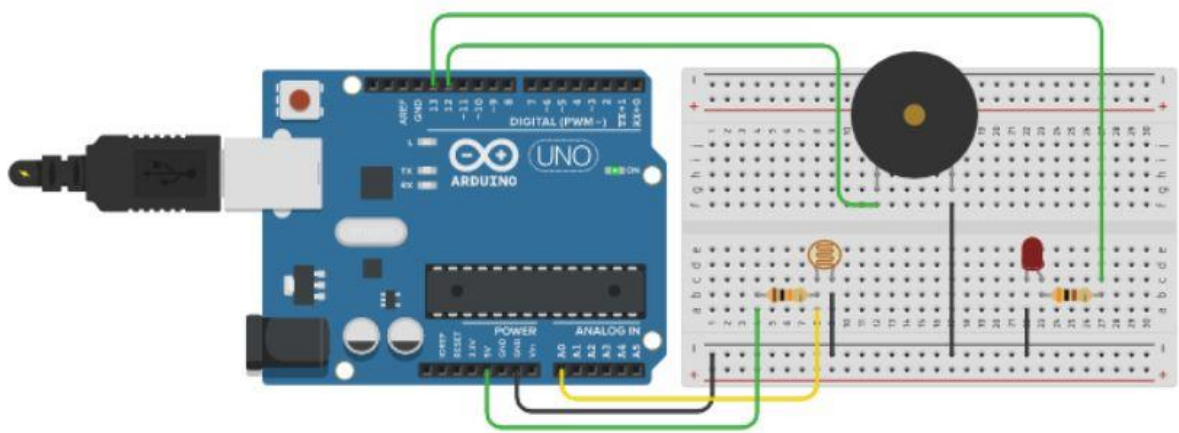
    }
Else {
    digitalWrite(ledPin,LOW);//DEFINE O LED COMO DESLIGADO
}

Serial.println(ldrValor);//INFORMA O VALOR DO SENSOR NA TELA
delay(100);
}

```

DA do Projeto 05:

Figura 9 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 05



Fonte: do autor.

Código com a resposta do Desafio Mais Amplo do Projeto 05:

```

int ledPin = 13; //DEFINE PINO DIGITAL DO LED
int buzzerPin = 12;//DEFINE PINO DIGITAL DO BUZZER
int ldrPin = A0; //DEFINE PINO ANALOGICO DO LED
int ldrValor = 0; //VARIÁVEL QUE ARMAZENA VALOR DO SENSOR
void setup() {
    pinMode(ledPin,OUTPUT); //DEFINE O PINO DO LED COMO SAIDA
    pinMode(buzzerPin,OUTPUT); //DEFINE O PINO DO BUZZER COMO SAIDA
    Serial.begin(9600); //INICIALIZA A COMUNICAÇÃO SERIAL
}
void loop() {
    ldrValor = analogRead(ldrPin); //ARMAZENA VALOR DO SENSOR NA VARIÁVEL
    if (ldrValor>= 800){
        //SE O VALOR DO SENSOR FOR MAIOR QUE 800 LIGA O LED
        digitalWrite(ledPin,HIGH); //DEFINE O LED COMO LIGADO
        digitalWrite(buzzerPin,HIGH); //DEFINE O BUZZER COMO LIGADO
    }
}

```



```
    }  
    else {  
        digitalWrite(ledPin, LOW); //DEFINE O LED COMO DESLIGADO  
        digitalWrite(buzzerPin, LOW); //DEFINE O BUZZER COMO DESLIGADO  
    }  
    Serial.println(ldrValor); //INFORMA O VALOR DO SENSOR NA TELA  
}
```

6 OFICINA 5: PROJETO 6

Esta oficina apresenta o Projeto 06, que utiliza o Arduino detectando presença com sensor infravermelho.

6.1 PROJETO 06: DETECÇÃO DE PRESENÇA COM SENSOR INFRAVERMELHO

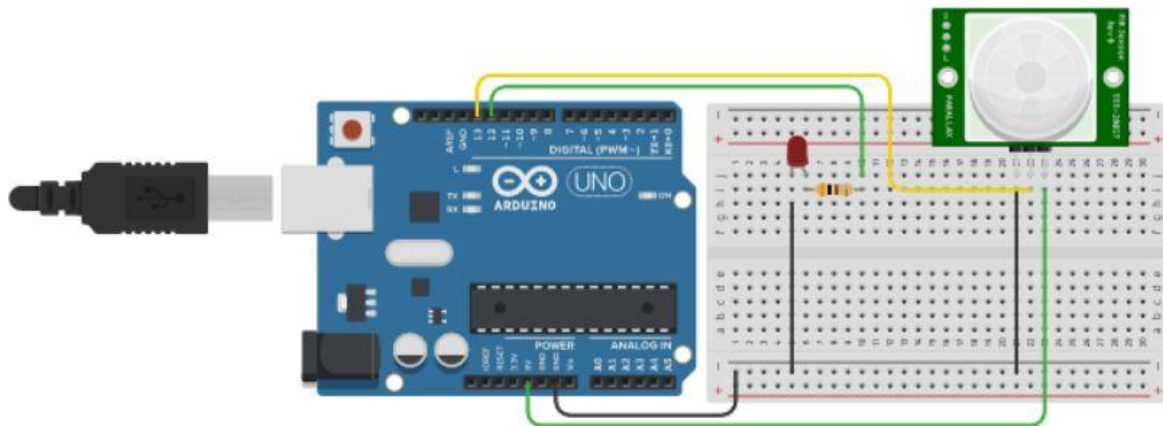
Objetivos: apresentação e funcionamento de um sensor de movimento que ao identificar um movimento no ambiente um LED acende.

Materiais:

1. um Arduino;
2. uma protoboard;
3. um cabo USB;
4. fios jumper;
5. um resistor;
6. um LED;
7. um sensor de movimento.

Materiais didáticos:

Figura 10 – Folha com o circuito do Desafio Inicial para o Projeto 06



Fonte: do autor.

Momentos didáticos:

DI: Como detectar presença com sensor infravermelho e informar situação através de LED?

MSEM: Conceituação e resolução de como detectar presença com sensor infravermelho.

DA: Como detectar presença com sensor infravermelho e informar presença através de som?

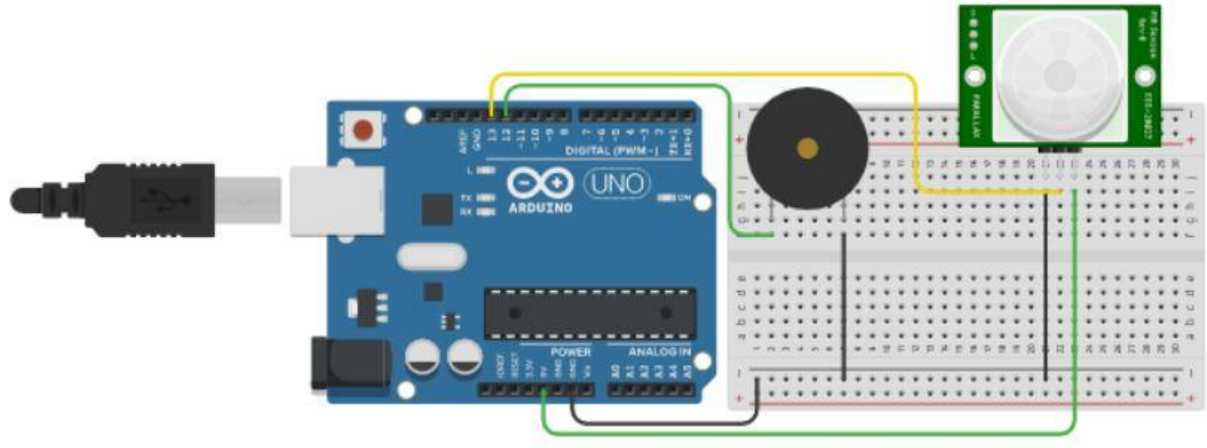
MSEM do Projeto 06:

```
const int pinoPIR = 13; //PINO DIGITAL UTILIZADO PELO SENSOR DE PRESENÇA
const int pinoLED = 12; //PINO DIGITAL UTILIZADO PELO LED
void setup() {
    pinMode(pinoLED, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinoPIR, INPUT); //DEFINE O PINO COMO ENTRADA
}
void loop() {
    if(digitalRead(pinoPIR) == HIGH) {
        //SE A LEITURA DO PINO FOR IGUAL A HIGH, FAZ
        digitalWrite(pinoLED, HIGH); //ACENDE O LED
    }
    else {
```

```
digitalWrite(pinoLED, LOW);}} //APAGA O LED
```

DA do Projeto 06:

Figura 11 – Folha com o circuito com a resposta do Desafio Mais Amplo do Projeto 06



Fonte: do autor.

Código com a resposta do Desafio Mais Amplo do Projeto 06:

```
const int pinoPIR = 13; //PINO DIGITAL UTILIZADO PELO SENSOR DE PRESENÇA
const int pinoBUZZER = 12; //PINO DIGITAL UTILIZADO PELO BUZZER
void setup() {
    pinMode(pinoBUZZER, OUTPUT); //DEFINE O PINO COMO SAÍDA
    pinMode(pinoPIR, INPUT); //DEFINE O PINO COMO ENTRADA
}
void loop() {
    if(digitalRead(pinoPIR) == HIGH){
        //SE A LEITURA DO PINO FOR IGUAL A HIGH, FAZ
        digitalWrite(pinoBUZZER, HIGH); //LIGA O BUZZER
    }
    else {
        //SENÃO, FAZ
        digitalWrite(pinoBUZZER, LOW); //DESLIGA O BUZZER
    }
}
```

7 OFICINA 6: PROJETO FINAL

Esta oficina tem como objetivo desenvolver um projeto mais amplo com base nos projetos colocados em prática, aplicando todo o conhecimento adquirido ao longo das oficinas.

Objetivos:

Desenvolvimento de um protótipo de um carro com funções autônomas.

Materiais:

1. Arduino;
2. protoboard;
3. cabo USB;
4. fios jumper;
5. LEDs;
6. buzzer;
7. diodos;
8. resistores;
9. transistores;
10. sensor de luminosidade;
11. sensor de distância;
12. motores;
13. baterias.

Materiais Didáticos:

Os materiais didáticos utilizados pelos alunos nesse projeto foram as folhas com os circuitos desenvolvidos nos projetos anteriores.

Momentos Didáticos:

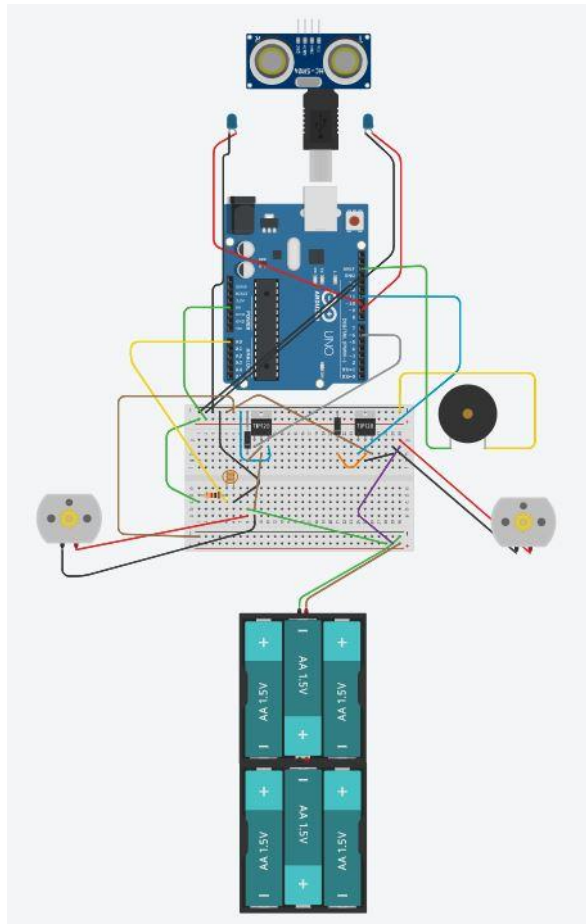
DI: Como desenvolver o protótipo de um carro com funções autônomas?

MSEM: Conceituação de como construir um protótipo de carro com funções autônomas.

DA: Resolução de como construir e colocar em funcionamento o protótipo de um carro com funções autônomas.

MSEM do Projeto Final:

Figura 12 – Folha com o circuito da MSEM do Projeto Final



Fonte: do autor.

Código da MSEM do Projeto Final:

```
#include <Ultrasonic.h>
const int ledPin1 = 9;
const int ledPin2 = 10;
const int LDR = A0;
const int motord = 11;
const int motore = 6;
const int buzzer = 13;
#define pino_trigger 4
#define pino_echo 5
Ultrasonic ultrasonic(pino_trigger, pino_echo);
float valLDR = 0;
void setup() {
```

```

    pinMode( ledPin1, OUTPUT);
    pinMode( ledPin2, OUTPUT);
    pinMode( LDR, INPUT);
    pinMode( motord, OUTPUT);
    pinMode( motore, OUTPUT);
    pinMode( buzzer, OUTPUT);
    Serial.begin( 9600);
}
void loop() {
    valLDR = analogRead( LDR);
    valLDR = map( valLDR, 0,1023,0,255);
    analogWrite(ledPin1,valLDR);
    analogWrite(ledPin2, valLDR);
    analogWrite( motord, valLDR);
    analogWrite(motore, valLDR);
    float cmMsec, inMsec;
    long microsec = ultrasonic.timing();
    cmMsec = ultrasonic.convert(microsec, Ultrasonic::CM);
    Serial.print("Distancia em cm: ");
    Serial.println(cmMsec);
    delay(1000);
    int distancia = 15;
    if( cmMsec< distancia){
        digitalWrite(buzzer, HIGH);
    }else{
        digitalWrite(buzzer, LOW);
    }
}
}

```

REFERÊNCIAS

DE BASTOS, F. P.; MAZZARDO, M. D. Prática escolar dialógico-problematizadora mediada por tecnologia informática livre. **Linguagens & Cidadania**, Santa Maria, v. 7, n. 1, jan./jun. 2005. Disponível em: <<https://periodicos.ufsm.br/LeC/article/view/28532>>. Acesso em: 15 abr. 2019.

FREIRE, P. **Pedagogia do Oprimido**. Rio de Janeiro: Paz e Terra, 2005.

MCROBERTS, M. **Arduino básico**. 2. ed. São Paulo: Ed. Novatec, 2015. 512 p.

PORTES, W. A. O. **Utilização de Arduino e eletrônica na automação residencial com acessibilidade a pessoa portadora de deficiência**. 2014. 45 p. Trabalho de Conclusão de Curso (Graduação em Análise e Desenvolvimento de Sistemas)- Instituto Municipal de Ensino Superior de Assis, Assis, SP, 2014.

REAL, M. P. C. et al. Dialogicidade e tema gerador: problematização teórico-metodológica para o ensino superior em três momentos pedagógicos. **Revista Signos**, Lageado, v. 39, n. 1, jun. 2018. Disponível em: <http://www.univates.br/revistas/index.php/signos/article/view/1803>. Acesso em: 15 abr. 2018.

