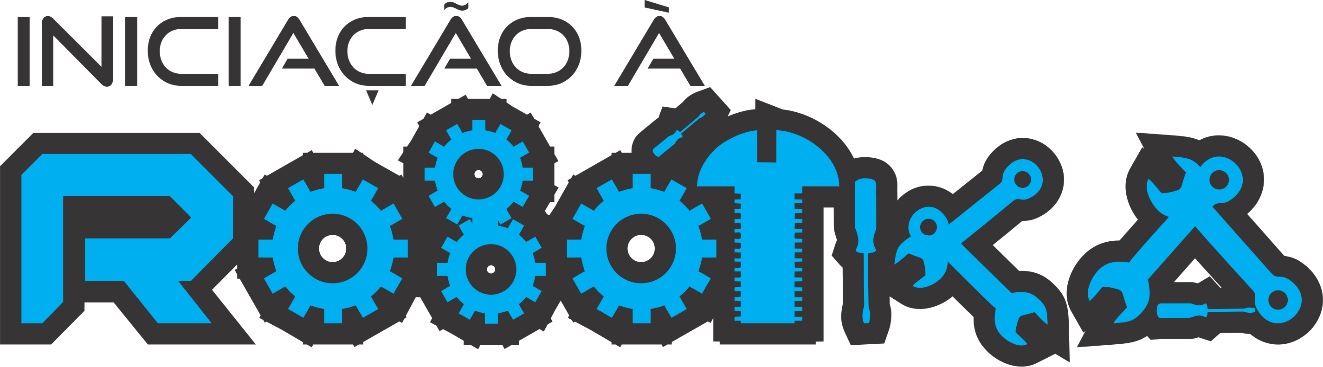
****

**Prof. Me. Mustafa Gonçalves Sahid**

**Prof. Hélio Evangelista da Silva**

Organizadores

**Prof.ª Ma. Joseane Gabriela A. Mezerhane Correia**

Revisora

Rio Branco / 2019 – 4ª Ed.

**“Quem perde tempo, eterna perda chora. ”**

Olavo Bilac

**APRESENTAÇÃO**

Esse material foi organizado pelo Centro de Matemática, Ciências e Filosofia – CMCF da Secretaria de Estado da Educação, Cultura e Esporte.

Os principais objetivos desse curso são conhecer a robótica, robôs e suas aplicações, identificar as características e funções dos componentes de robôs de aprendizagem, a Plataforma Arduino e aprender a trabalhar com softwares de programação visual visando desenvolver, nos alunos da educação básica do Acre, o raciocínio lógico e hipotético dedutivo, a capacidade de concentração e o interesse pela robótica.

Bom proveito!

**Prof. Me. Mustafa Gonçalves Sahid**

**Prof. Hélio Evangelista da Silva**

**SUMÁRIO**

|  |
| --- |
| Aulas 1 a 4 – Robótica, Robô, Aplicações e Conhecendo o robô DUDE: características e funções........04  1.O que é Robótica? ................................................................................................................................04  2.O que é Robô? E quais suas aplicações? .............................................................................................04  3.Conhecendo o Robô DUDE: características e funções..........................................................................04  3.1. O Design.............................................................................................................................................04  3.2. Componentes......................................................................................................................................05  3.2.1 Shields..............................................................................................................................................05  3.2.2 Placa Arduino....................................................................................................................................06  3.2.3 Sensores...........................................................................................................................................06  3.2.3.1 Sensor ultrassônico.......................................................................................................................06  3.2.3.2 Sensor seguidor de linha...............................................................................................................07  3.2.3.3 Sensor de temperatura..................................................................................................................07  3.2.3.4 Sensor de luz.................................................................................................................................08  3.2.3.5 Sensor infravermelho.....................................................................................................................08  3.2.4 Led RGB...........................................................................................................................................09  3.2.5 Buzzer...............................................................................................................................................09  3.2.6 Botão ou push buttons......................................................................................................................10  3.2.7 Servomotor.......................................................................................................................................10  Ativ. A1 a A4 – Robótica, Robô, Aplicações e Conhecendo o robô DUDE: características e funções …10  Aulas 5 e 6 – Plataforma Arduino, Linguagem C e Algumas Aplicações .................................................11  1. O Hardware............................................................................................................................................11  2. O Software.............................................................................................................................................12  3. Linguagem de Programação .................................................................................................................12  4. Linguagem C .........................................................................................................................................12  5. Etapas para a construção de um protótipo............................................................................................13  Atividades A5 e A6 – Plataforma Arduino, Linguagem C e Algumas Aplicações .....................................13  Aulas 07 a 10 – Ardublock - Programação Gráfica para Arduino..............................................................15  1. Programando com o Ardublock.............................................................................................................16  1.1 Led piscando, com uma cor, usando o setar pin digital.......................................................................16  1.2 Led piscando, com cores diferentes, utilizando o led RGB.................................................................18  1.3 Sirene...................................................................................................................................................19  1.4 Sensor de luz de topo..........................................................................................................................20  1.5 Sensor ultrassônico.............................................................................................................................21  1.6 Sensor de luz de topo azul e vermelha, senão verde..........................................................................22  1.7 Dog Inteligente com LEDs...................................................................................................................23  Atividades A07 a A10 – Ardublock - Programação Gráfica para Arduino..................................................23  REFERÊNCIAS..........................................................................................................................................24 |

Aulas 1 a 4 – Robótica, Robô, Aplicações e Conhecendo o robô DUDE: características e funções

1. **O que é Robótica?**

**Robótica** é um ramo educacional e [tecnológico](https://pt.wikipedia.org/wiki/Tecnologia_da_informa%C3%A7%C3%A3o) que engloba [computadores](https://pt.wikipedia.org/wiki/Computadores), [robôs](https://pt.wikipedia.org/wiki/Rob%C3%B4s) e [computação](https://pt.wikipedia.org/wiki/Computa%C3%A7%C3%A3o), que trata de sistemas compostos por partes mecânicas automáticas e controladas por [circuitos integrados](https://pt.wikipedia.org/wiki/Circuito_integrado), tornando sistemas mecânicos motorizados, controlados manualmente ou automaticamente por [circuitos elétricos](https://pt.wikipedia.org/wiki/Circuito_el%C3%A9ctrico).

1. **O que é Robô? E quais suas aplicações?**

Um **robô** (ou **robot**) é um dispositivo, ou grupo de dispositivos, eletromecânicos ou biomecânicos capazes de realizar trabalhos de maneira autônoma ou pré-programada. Os robôs são comumente utilizados na realização de tarefas em locais mal iluminados, ou na realização de tarefas sujas ou perigosas para os [seres humanos](https://pt.wikipedia.org/wiki/Ser_humano).

[](https://www.google.com.br/url?sa=i&rct=j&q=&esrc=s&source=images&cd=&cad=rja&uact=8&ved=0ahUKEwjluNrHsqXNAhUBQZAKHQxlCiMQjRwIBw&url=http://fernandotecnobox.blogspot.com/2011/02/robos-sao-cobaias-para-soldados-do.html&bvm=bv.124272578,d.Y2I&psig=AFQjCNGTq-OONgf6TWH50qanHohAvsv_xQ&ust=1465920754800061)

A idéia de construir robôs começou no inicio do século XX com George Devol, o pai da Robótica Industrial, que através de grande inovação, proporcionou a introdução de robôs nas fabricas. Hoje em dia, a robótica vem sendo aplicada cada vez mais nas fábricas para a construção de robôs que visam reduzir custos e aumentar a produtividade substituindo a mão humana. Seu emprego é lucrativo para as empresas, pois o robô trabalha 24 horas por dia, não descansa, não falta e não precisa pagar salário ou direitos trabalhistas.

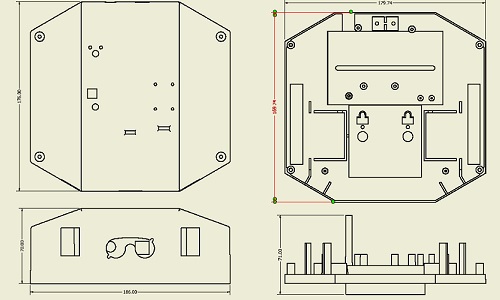
Os [robôs industriais](https://pt.wikipedia.org/wiki/Rob%C3%B4_industrial) utilizados nas linhas de produção são a forma mais comum de robôs, uma situação que está mudando recentemente com a popularização dos robôs comerciais limpadores de pisos e cortadores de gramas. Outras aplicações são: tratamento de lixo tóxico, exploração subaquática e espacial, [cirurgias](https://pt.wikipedia.org/wiki/Cirurgia_rob%C3%B3tica), mineração, busca e resgate, e localização de minas terrestres. Os robôs também aparecem nas áreas do entretenimento e tarefas caseiras.

1. **Conhecendo o Robô DUDE: características e funções**

3.1 ****O Design****

O design do robô foi constituído basicamente para comportar a interface de controle Arduino e o sistema de locomoção é baseado em servo motores. Pensando na utilização por crianças do Ensino Fundamental, o chassi/carcaça do robô foi construído em plástico injetado, a fim de não preocupar os professores com quebras devido a fragilidade do material.



 Design do robô DUDE

* 1. **Componentes**

O robô é capaz de realizar algumas atividades pré-estabelecidas, como: seguir linhas, desviar de obstáculos com o seu sistema de infravermelho ou com o sonar. Também pode ser utilizado para medir variáveis ambientais como temperatura, luminosidade ou som. Pode emitir sons e sinais luminosos através de Leds. Se movimenta através de servo motores programáveis. Outros sensores podem ser acoplados através de uma porta de comunicação na sua parte superior. O robô é alimentado por um conjunto de 6 pilhas AA e programado através de um cabo do tipo USB.







* + 1. **Shields**

"Shields" é o nome dado às placas de expansão de hardware que encaixam na placa Arduino principal. Através dos Shields, uma placa Arduino pode ser melhorada ainda mais.

Os circuitos contidos nos diversos Shields contêm uma eletrônica que adiciona funções que a placa principal não possui. Como exemplos temos Shields para controle de motores, Shields para comunicação Wi-Fi, Shields para acoplamento de sensores, e muito mais.

* + 1. **Placa Arduino**

**O Arduino é uma placa** de prototipagem eletrônica de código aberto **que permite a automação de projetos eletrônicos e robóticos por profissionais e amadores.** Uma placa Arduino é composta, basicamente, por um controlador Atmel AVR de 8 bits, uma interface serial ou USB e alguns pinos digitais e analógicos. Assim, a partir desses componentes, ela pode servir tanto para o desenvolvimento de projetos interativos como ser conectada a um outro computador. Em suma, o Arduino é uma plataforma sobre a qual serão construídos outros equipamentos.

****

* + 1. **Sensores**

Um sensor é um dispositivo que detecta um estímulo físico (calor, luz, som, pressão, campo magnético, movimento) e transmite um impulso (mensurável ou operante) correspondente. Em outras palavras são artefatos que permitem obter informação do meio e interagir com o mesmo.

Alguns sensores respondem com sinal elétrico a um estímulo, isto é, convertem a energia recebida em um sinal elétrico. Nesse caso, podem ser chamados de transdutores. O transdutor converte um tipo de energia em outro. É geralmente composto por um elemento sensor e uma parte que converte a energia proveniente dele em sinal elétrico. O conjunto formado por um transdutor, um condicionador de sinal (amplificador) e um indicador é chamado de sistema de medição.  
 Quando o sinal é disponibilizado não por um indicador, mas na forma de corrente ou tensão já condicionado - (4 a 20) mA ou (0 a 5) V, geralmente - o dispositivo é chamado de transmissor.

Como o sinal é uma forma de energia, os sensores podem ser classificados de acordo com o tipo de energia que detectam. Veremos a seguir alguns exemplos de sensores utilizados no robô DUDE.

**3.2.3.1 Sensor ultrassônico**

Um **sensor ultrassônico** é um dispositivo que utiliza alta frequência de som para medir a distância entre itens determinados. Estes sensores são também conhecidos como transceptores, e são capazes de operar semelhante ao sonar. Enquanto o sonar é principalmente utilizado debaixo da água, os transceptores de ultrassom podem ser utilizados no ambiente terrestre, tendo o ar como meio de transmissão. Os sensores de ondas ultrassônicas são comuns em aplicações industriais e médicas, além de outras aplicações.

Este tipo de sensor consiste tipicamente de uma unidade de transceptor único, que é capaz de emitir e detectar o som. Este dispositivo cria um pulso sonoro que está além da faixa de audição do ouvido humano. A maioria dos objetos sólidos é capaz de refletir ondas sonoras. O transceptor utiliza um temporizador para determinar com precisão quanto tempo um pulso ultrassônico leva para “saltar” em um objeto, e retornar à unidade.

****

**3.2.3.2 Sensor seguidor de linha**

****Os sensores seguidores de linha são sensores que detectam as diferentes intensidades de reflexão da luz nos materiais. São muito usados para detectar linhas ou bordas com intensidades de reflexão diferentes da superfície onde estão colocados.

**3.2.3.3 Sensor de temperatura**

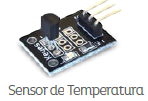
Temperatura é um indicador de condições térmicas de um corpo ou material homogêneo. Ela indica a energia gerada pela agitação das moléculas dos materiais.  
A medição de temperatura convencional é baseada na propriedade de certos materiais, para alterar sua forma física ou volume (dilatação), proporcional ao calor aplicado.

Um **Sensor de temperatura é um transdutor** que, submetido a uma mudança de temperatura, fornece uma resposta claramente dependente da temperatura.

**Transmissores**, também chamados de **transdutores de temperatura**, convertem a temperatura proveniente da variação de resistência ou da variação de tensão do **termopar**, em um sinal de saída analógico (corrente) linear ou ainda em um sinal de saída digital (protocolos), sendo que o sinal 4...20mA é o mais usado.

**Termopares** são sensores de temperatura que enviam diretamente uma variação de tensão proporcional à variação de temperatura, devido as suas propriedades termoelétricas.

São exemplos de sensores de temperatura: termômetros, termopares e termostatos.



* + - 1. **Sensor de luz**

A finalidade de um sensor fotoelétrico é converter um sinal luminoso (luz ou sombra) num sinal elétrico que possa ser processado por um circuito eletrônico.

Um sensor fotoelétrico pode ser tanto um transdutor como um sensor propriamente dito.

Dizemos que um sensor fotoelétrico é um transdutor quando ele converte energia luminosa (radiante) em energia elétrica. É o caso das fotocélulas que convertem diretamente luz em energia elétrica.

Por outro lado, temos sensores propriamente ditos que convertem luz em uma variação de uma grandeza elétrica qualquer como corrente ou resistência. Esse é o caso dos **LDRs** e dos fotodiodos.

O **LDR**, sigla em inglês de Light-Dependent Resistor, que significa resistor dependente de luz, nada mais é do que o que o próprio nome diz. Tipicamente, quanto maior a luz incidente nesse componente, menor será sua resistência.



No robô que iremos estudar estão inseridos os sensores de luz frontal e superior, conforme as figuras a seguir:





* + - 1. **Sensor infravermelho**

São sensores que podem ser **ativos** ou **passivos**. O primeiro é composto por um emissor de luz infravermelha e um receptor, que reage a essa luz. Já o segundo, não emite radiação, percebem a presença de pessoas no ambiente por meio da radiação de calor emitida pelo corpo em movimento (comprimento de onda na faixa de 10 mícrons). Estes sensores somente conseguem detectar o movimento se estiverem alinhados com o ocupante, não sendo possível detectar ocupantes caso haja paredes, janelas ou outros obstáculos no caminho. Devido à constituição de suas lentes, que segmenta a área em cones, estes sensores possuem limitações para detectar pequenos movimentos em grandes distâncias, maiores que 5 metros.

Os sensores infravermelhos são encontrados numa infinidade de aplicações eletrônicas. Podemos citar os leitores de códigos de barras, detectores de objetos ou alarmes de passagem, encoders (dispositivo [eletromecânico](https://pt.wikipedia.org/wiki/Eletromec%C3%A2nica) que conta ou reproduz [pulsos elétricos](https://pt.wikipedia.org/w/index.php?title=Pulso_el%C3%A9trico&action=edit&redlink=1) a partir do movimento rotacional de seu eixo), chaves ópticas e muito mais.



**3.2.4 Led RGB**

**RGB** é a abreviatura do sistema de [cores](https://pt.wikipedia.org/wiki/Cor) aditivas formado por [Vermelho](https://pt.wikipedia.org/wiki/Vermelho) (**R**ed), [Verde](https://pt.wikipedia.org/wiki/Verde) (**G**reen) e [Azul](https://pt.wikipedia.org/wiki/Azul) (**B**lue). O propósito principal do sistema RGB é a reprodução de cores em dispositivos eletrônicos como monitores de TV e computador, retroprojetores, scanners e câmeras digitais, assim como na fotografia tradicional. Modelos aditivos de luzes são combinados de várias maneiras para reproduzir outras cores.



**3.2.5 Buzzer**

Buzzer é um dispositivo de áudio que pode ser mecânico, eletrônico ou piezoelétrico (componente eletrônico emissor de som em frequência predefinida). Em outras palavras, o buzzer nada mais é do que um pequeno alto-falante capaz de emitir sons em diversas frequências. Ele é normalmente usado em projetos que necessitam de avisos sonoros, relógios com alarme e até para reproduzir músicas.



**3.2.6 Botão ou push buttons**

O interruptor momentâneo é um componente que conecta dois pontos de um circuito ao pressioná-lo. Um exemplo é ligar um LED quando pressionamos o interruptor.



* + 1. **Servomotor**

**Servomotor** é um motor elétrico e mecânico, ou seja, é uma combinação constituída por um circuito elétrico e um mecanismo para controlar com precisão o **Servo Módulo** com movimentos precisos girando em ângulos especificados através de um sinal.

Os **servos** são elementos importantes no mundo da Robótica, pois com eles podemos construir vários projetos como, por exemplo, garras, braços, robôs e etc.

O **servo** é um elemento semelhante a uma caixa de redução, composto basicamente por um motor CC e um conjunto de engrenagens.





Atividades A1 a A4 – Robótica, Robô, Aplicações e Conhecendo o robô DUDE: características e funções

* Formem duplas, pesquisem na internet e salvem, de forma conveniente, outras contribuições e (ou) curiosidades sobre as características e funções do robô DUDE para apresentar aos demais grupos.

Aulas 5 e 6 – Plataforma Arduino, Linguagem C e Algumas Aplicações

Arduino é uma plataforma de prototipagem eletrônica, criado por Massimo Banzi e David Cuartielles em 2005 com objetivo de permitir o desenvolvimento de controle de sistemas interativos, de baixo custo e acessível a todos.

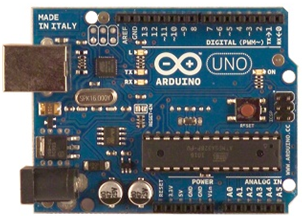
O projeto foi criado pensando em artistas e amadores, ou seja, não é necessário ter conhecimentos prévios em eletrônica ou programação para iniciar-se no mundo Arduino.

Com o Arduino é possível também enviar e receber informações de praticamente qualquer outro sistema eletrônico. Desta forma é possível construir por exemplo, um sistema de captação de dados de sensores, como temperatura, iluminação, processar e enviar esses dados para um sistema remoto por exemplo.

Outra característica importante é que todo material (software, bibliotecas, hardware) é open-source, ou seja, pode ser reproduzido e usado por todos sem a necessidade de pagamento de royalties ou direitos autorais.

A plataforma é composta essencialmente de duas partes: O Hardware e o Software.

* 1. O Hardware

Arduino uno

Vamos abordar com detalhes o hardware do Arduino. Resumidamente o hardware é uma placa eletrônica que:

* Possui todos componentes necessários para a maioria dos projetos;
* Contém uma eletrônica que permite usar a placa com diversas fontes de energia, baterias e fontes de alimentação;
* Permite o acoplamento de circuitos externos através de pinos de conexão em posições padronizadas;
* A eletrônica é baseada em componentes de fácil obtenção, inclusive no mercado brasileiro;
* O esquema da placa é livre, e pode ser facilmente modificado ou adaptado;
* A placa é programada, ou seja, escrevemos um software que ficará embutido no chip controlador (firmware).
  1. O Software

O Arduino é um compilador (C e C++) que usa uma interface gráfica construída em Java baseado no projeto Processing. Tudo isso resume-se a um programa IDE ([*Ambiente de Desenvolvimento Integrado*](https://pt.wikipedia.org/wiki/Ambiente_de_Desenvolvimento_Integrado)*),* muito simples de usar e de estender com bibliotecas que podem ser facilmente encontradas na internet (aos montes).

Ambiente de desenvolvimento do Arduino

Depois de criar o programa e compilar usando a IDE, o código gerado é enviado para a placa onde é gravado dentro do chip controlador. Esse software que roda na placa chama-se FIRMWARE.

As funções da IDE do Arduino são basicamente duas: Permitir o desenvolvimento de um software e enviá-lo à placa para que possa ser executado.

3. Linguagem de Programação

Linguagem de programação é um método padronizado para comunicar [instruções](https://pt.wikipedia.org/wiki/C%C3%B3digo_de_m%C3%A1quina) para um [computador](https://pt.wikipedia.org/wiki/Computador). É um conjunto de [regras sintáticas](https://pt.wikipedia.org/wiki/Sintaxe) e semânticas usadas para definir um [programa de computador](https://pt.wikipedia.org/wiki/Programa_de_computador). Permite que um programador especifique precisamente sobre quais dados um computador vai atuar, como estes dados serão armazenados ou transmitidos e quais ações devem ser tomadas sob várias circunstâncias.

Uma das principais metas das linguagens de programação é que programadores tenham uma maior produtividade, permitindo expressar suas intenções mais facilmente do que quando comparado com a linguagem que um computador entende nativamente (código de máquina).

Existem [várias linguagens de programação](https://pt.wikipedia.org/wiki/Lista_de_linguagens_de_programa%C3%A7%C3%A3o), mas nesta apostila nos restringiremos a comentar um pouco sobre a linguagem C.

**4. Linguagem C**

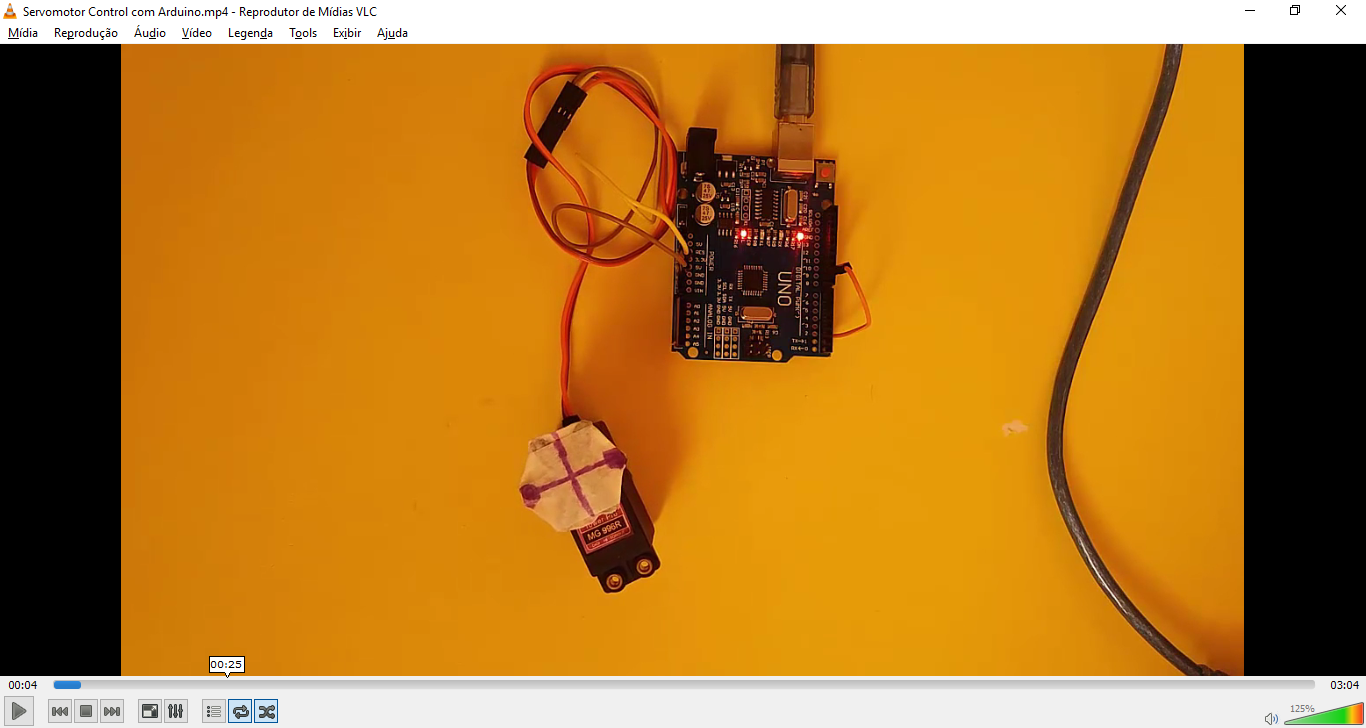
C é uma linguagem imperativa (descreve ações, enunciados ou comandos que mudam o estado de um programa) e procedural (especifica os passos que um programa deve seguir para alcançar um estado desejado), para implementação de sistemas. Seus pontos de design foram para ele ser compilado, fornecendo acesso de baixo nível à memória e baixos requerimentos do hardware. Também foi desenvolvido para ser uma linguagem de alto nível, para maior reaproveitamento do código. Seu objetivo principal é facilitar a criação de programas extensos com menos erros, recorrendo ao paradigma da [programação procedural](https://pt.wikipedia.org/wiki/Programa%C3%A7%C3%A3o_procedural), mas sobrecarregando menos o autor do [compilador](https://pt.wikipedia.org/wiki/Compilador), cujo trabalho complica-se ao ter de realizar as características complexas da linguagem.

5. Etapas para a construção de um protótipo

O processo de construção de um circuito de controle básico resume-se à:

* Escrever um programa usando a interface de desenvolvimento do Arduino;
* Conectar a placa do Arduino no computador através de um cabo (USB é o mais comum);
* Compilar o programa escrito;
* Enviar o programa compilado para a placa e observar o funcionamento.

Vejamos um vídeo de aplicação utilizando a Plataforma Arduino, um servo motor e uma protoboard.



Atividades A5 e A6 – Plataforma Arduino, Linguagem C e Algumas Aplicações

* Formem duplas e, através dos kits fornecidos, construam as experiências a seguir com Programação em C++ e a Plataforma Arduino.

**1. Fazer o Arduino piscar um LED de um em um segundo**

Com o Arduino conectado ao computador e devidamente configurado, mostraremos como programá-lo, em C++, para piscar um LED de um em um segundo.

// Existe um LED conectado no pino 13 da maioria dos Arduinos.

// Daremos um nome a este pino; Int (devolve um valor inteiro):

int led = 13;

// Esta função "setup"(configuração) roda uma vez quando a placa é ligada ou resetada (void=vazio)

void setup () {

// Configura o pino do led (digital) como saída (output)

pinMode (led, OUTPUT);

}

// Função que se repete infinitamente quando a placa é ligada

void loop () {

digitalWrite (led, HIGH); // liga o LED (HIGH = nível lógico alto; Write=Escreva)

delay (1000); // Espera um segundo

digitalWrite (led, LOW); // Desliga o LED (LOW = nível lógico baixo)

delay(1000); // Espera, demora (delay) um segundo

}

Agora clique em Upload para que o programa seja transferido para seu Arduino.

Se tudo foi feito corretamente, o led do pino 13, presente na placa do Arduino UNO, irá piscar intermitentemente.

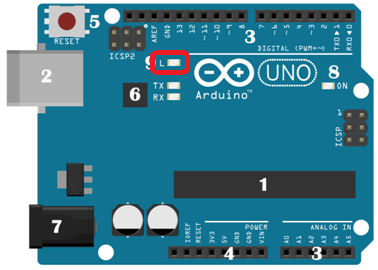


Figura 4 - Led conectado ao pino 13

**2. Conforme você pressiona um push button, um led é aceso**

Trata-se de fazer um botão acender um led quando pressionado e, quando solto, o led deverá apagar. Coloque os componentes como está sendo mostrado na imagem abaixo, bem como suas ligações:

Componentes utilizados: 01x Led Vermelho / 01x Resistor de 300Ω ( laranja, preto e marron, da esquerda para direita) / 01x PushButton / 01x Resistor de 10kΩ (10.000 Ω=marron, preto e laranja, da esquerda para direita) / cabos diversos.



Não importam as cores dos fios na hora de ligação, e sim fazer a ligação correta. Preste bastante atenção ao fazer as ligações pois apenas 1 fio no lugar errado pode comprometer todo o experimento.

A programação do experiemento será:

/\*

Button

Utiliza um botão de pressão, conectado ao pino 2, para ligar e desligar um LED conectado ao pino digital 13.

\*/

// Determinamos constantes para os números dos pinos utilizados

const int buttonPin = 2; // Número do pino do botão de pressão

const int ledPin = 13; // Número do pino do led

// Variáveis

int buttonState = 0; // Variável para leitura do estado do botão

// Executa uma vez ao ligar ou reiniciar a placa

void setup () {

pinMode (ledPin, OUTPUT); //inicializa o pino do LED como saída (OUTPUT)

pinMode (buttonPin, INPUT); // Inicializa o pin do botão como entrada (INPUT)

}

// Executa infinitamente quando liga a placa

void loop () {

// Lê o estado do botão (HIGH -> +5V -> botão press.) (LOW -> 0V)

buttonState = digitalRead (buttonPin);

// Testa se o botão está pressionado

// Se sim, o estado do botão e alto (HIGH)

if (buttonState == HIGH) {

digitalWrite (ledPin, HIGH); // Liga o LED

}

// Senão (Botão não pressionado)

else {

digitalWrite (ledPin, LOW); // Desliga o LED

}

Após compilar o código e fazer o upload na sua placa, você já deve poder apertar o botão e o led da protoboard acender e, quando soltar, o led deve apagar.

Aulas 07 a 10 – Ardublock - Programação Gráfica para Arduino

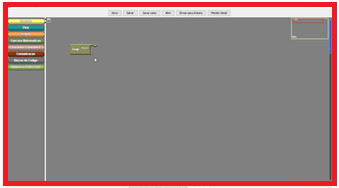
Ardublock é uma linguagem de programação que utiliza blocos de funções prontas. Do mesmo jeito que o Arduino ajuda entusiastas a entrar no meio da eletrônica e automação, o Ardublock ajuda a quem não tem conhecimento em linguagens de programação a criar programas para o Arduino de forma simples e intuitiva.

Como os blocos disponíveis do Ardublock se equiparam a funções de uma linguagem de programação, o Ardublock possui uma vasta possibilidade de utilização e aplicação.

[](https://www.circuitar.com.br/tutoriais/ardublock-programacao-grafica-para-arduino/#lightbox-content)

*Exemplo de programa feito no Ardublock*

Vejamos agora alguns vídeos de programação em Ardublock.



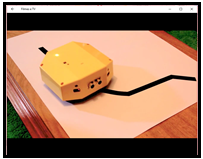


**Controlando o Dude com um Botão**





**Seguidor de Linhas**



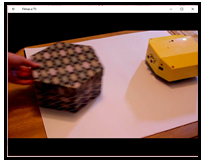


**Medindo Temperatura e a indicando com o Led**





**Sonar HD**





**1. Programando com o Ardublock**

##### **1.1 Led piscando, com uma cor, usando o setar pin digital**

Vamos demonstrar aqui como realizar um programa simples de piscar um LED no ambiente de programa em blocos (Ardublock). À esquerda do Ardublock se encontram todas as funções disponíveis, desde operadores lógicos e condições de estado até controles das entradas e saídas do Arduino, etc.

#### **Etapas**

##### **1.1.1 Função Loop**

Todo e qualquer programa para Arduino deve conter a função **loop**. Esta função é executada indefinidamente quando o Arduino é ligado. Portanto, é nela que queremos colocar nossos blocos.

* Aperte na seção **Controle** e arraste o bloco **loop** para a área de trabalho;

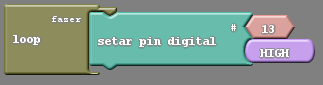
##### **1.1.2 Sentando o LED**

Para simplificação do projeto, iremos utilizar o LED interno, já embutido no Arduino, localizado no pino 13.

* Aperte na seção **Pins** e arraste o bloco **setar pin** **digital** para a área de trabalho;

Este bloco possui 2 argumentos: Número do pino e estado lógico desejado.

* Selecione o pino **13** e nível lógico **alto (HIGH)**;



##### **1.1.3 Tempo de espera**

Vamos utilizar o bloquinho **controle espera (ms)** para fazer um atraso de 1 segundo. Seu parâmetro é o tempo de atraso em milissegundos.

* Aperte na seção **Controle** e arraste o bloco **espera (ms)**;
* Configure o valor para 1000. (Correspondente a 1 segundo);

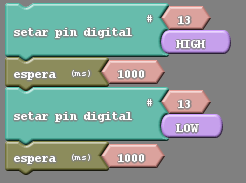
##### **1.1.4 Conexão**

O Ardublock, assim como a linguagem do Arduino, é sequencial, portanto ele executará uma função de cada vez. Para dar as ordens das instruções no Ardublock conectamos uma função em baixo da outra, simplesmente arrastando e conectando um bloquinho embaixo do outro.

* Faça isto para os 2 últimos blocos criados. Colocando o **espera (ms)** em baixo do **setar pin** **digital**.

Agora estes 2 bloquinhos formam um bloco só. Como queremos ligar o LED por um segundo e em seguida desligá-lo por um segundo precisamos de um segundo bloco deste para fazer a segunda operação.

* Clique com o botão direito neste bloco e aperte o botão **clone** para fazer uma cópia.
* Encaixe a cópia do bloco criado em baixo do bloco original, e configure o segundo **setar pin** **digital** para **LOW (desligado).**



##### **1.1.5 Conexão com o loop**

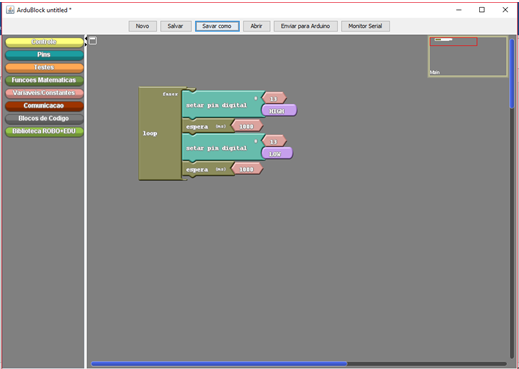
Este novo bloco agora representa o que queremos fazer no programa: Piscar o LED com o intervalo de 1 segundo.

* Para finalizar, ligue este bloco dentro do bloquinho **loop**, arrastando-o.
* Clique no botão **Enviar para Arduino** para gravar o programa no seu Arduino.



##### **1.1.6 Observações**

* Certifique-se na seção **tools** da janela do Arduino que a placa e a porta de comunicação foram selecionadas corretamente.
* O Ardublock se encarrega de configurar as entradas e saídas dos pinos, de forma a facilitar para o usuário.

****

##### **1.2 Led piscando, com cores diferentes, utilizando o led RGB**

#### **Etapas**

**1.2.1. Sentando o LED RGB**

Para a programação desejada utilize a **Biblioteca ROBO+EDU** e a seguir selecione, arraste e encaixe o **LED RGB** no Loop;

No LED RGB, selecione verdadeiro para a cor que deseja deixar em evidência e falso para as outras duas.



##### **1.2.2 Tempo de espera**

Através da função **Controle** selecione, arraste e encaixe o bloco **espera** **(ms)** no Loop e mude o tempo para 2000 (Correspondente a 2 segundos).



##### **1.2.3 Conexão**

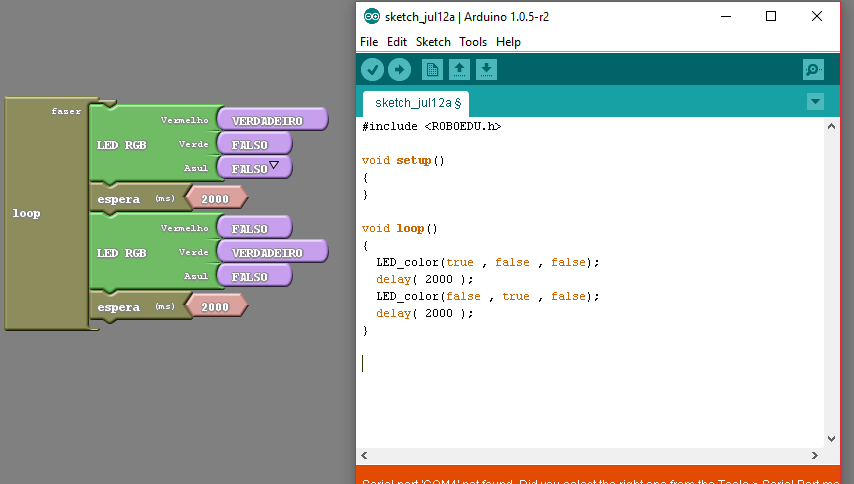
Agora estes 2 bloquinhos formam um bloco só. Como queremos ligar o LED por dois segundos em uma cor e dois segundos em outra cor, precisamos de um segundo bloco deste para fazer a segunda operação.

* Clique com o botão direito neste bloco e aperte o botão **clone** para fazer uma cópia.
* ****Encaixe a cópia do bloco criado em baixo do bloco original, e configure o segundo **LED RGB** para uma cor diferente da escolhida no bloco anterior.

##### **1.2.4. Enviando o programa para o Arduino**

Este novo bloco agora representa o que queremos fazer no programa: Piscar o LED, com cores diferentes, a cada 2 segundos.

* Clique no botão **Enviar para Arduino** para gravar o programa no seu Arduino.



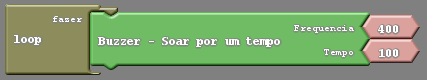
##### **1.3 Sirene**

#### **Etapas**

**1.3.1. Sentando o BUZZER**

Para a programação desejada utilize a **Biblioteca ROBO+EDU** e a seguir selecione, arraste e encaixe a função **Buzzer – Soar por um tempo** no Loop;

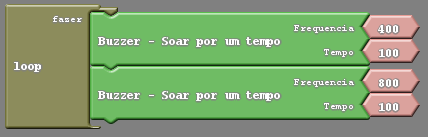
No **Buzzer – Soar por um tempo**, altere a frequência para 400 e o tempo para 100.



##### **1.3.2 Conexão**

Como queremos uma sirene, precisamos de um segundo bloco deste para alterar a frequência.

* Clique com o botão direito neste bloco e aperte o botão **clone** para fazer uma cópia.
* Encaixe a cópia do bloco criado em baixo do bloco original, e configure o segundo **Frequência** para 800.



##### **1.3.3. Enviando o programa para o Arduino**

Este novo bloco agora representa o que queremos fazer no programa: Uma **Sirene**, com frequência alternando entre 400 e 800, a cada 0,1 segundo.

* Clique no botão **Enviar para Arduino** para gravar o programa no seu Arduino.



##### **1.4 Sensor de luz de topo**

#### **Etapas**

**1.4.1. Sentando o SENSOR**

Para a programação desejada utilize **Controle** e a seguir selecione, arraste e encaixe a função **se/então** no Loop;

* Clique no menu **teste** e escolha a função **maior que** (>) e encaixe em teste;
* Na **Biblioteca ROBO+EDU** selecione a função **Sensor de luz de topo** e encaixe no lado esquerdo da função **maior que** (>);
* Na função **Variáveis/Constantes** selecione a função **Constante inteira** e insira o valor **50**.



**1.4.2. Sentando o Led RGB**

Na Biblioteca **ROBO+EDU** selecione, arraste e encaixe a função **Led RGB** ao lado de então.



##### **1.4.3 Conexão**

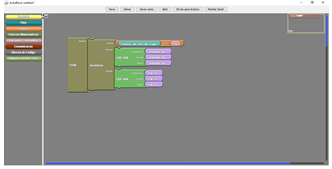
Queremos mostrar que: se o valor do sensor de luz **for maior que 50**, **então** liga o led, **senão** desliga o led.

* Clique com o botão direito neste bloco e aperte o botão **clone** para fazer uma cópia.
* Encaixe a cópia do bloco criado ao lado de **se/então**, e configure as cores para **falso**.



##### **1.4.4. Enviando o programa para o Arduino**

Clique no botão **Enviar para Arduino** para gravar o programa no seu Arduino.



##### **1.5 Sensor ultrassônico**

#### **Etapas**

**1.5.1. Sentando o SENSOR**

Para a programação desejada utilize **Controle** e a seguir selecione, arraste e encaixe a função **se/então** no Loop;

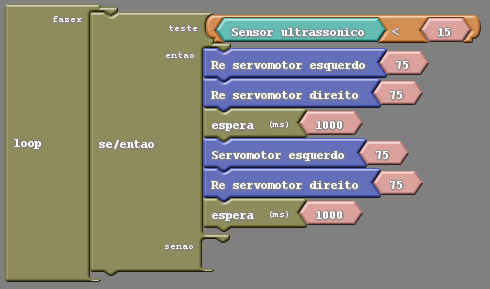
* Clique no menu **teste** e escolha a função **menor que** (<) e encaixe em teste;
* Na **Biblioteca ROBO+EDU** selecione a função **Sensor ultrassônico** e encaixe no lado esquerdo da função **menor que** (<);
* Na função **Variáveis/Constantes** selecione a função **Constante inteira** e insira o valor **15**.



**1.5.2. Sentando os servomotores no menu então**

Selecione, arraste e encaixe no menu **então:**

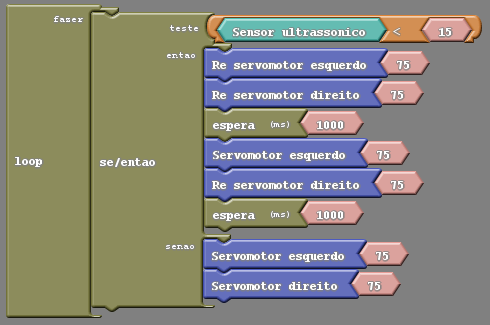
* **Re** **servomotor esquerdo** e **Re** **servomotor direito** obtidos na **Biblioteca ROBO+EDU;**
* No menu **Controle** selecione **espera (ms)** eencaixe abaixo da função Re servomotor direito inserindo o valor **1000**;
* Abaixo da função espera (ms), encaixe **Servomotor esquerdo** e, a seguir, Re **servomotor direito** e **espera (ms)** inserindo valor **1000**.



**1.5.3. Sentando os servomotores no menu senão**

Selecione, arraste e encaixe no menu **senão:**

* **Servomotor esquerdo** e **Servomotor direito** obtidos na **Biblioteca ROBO+EDU.**



Com os procedimentos anteriores ***se o valor do sensor ultrassônico for menor que 15, então o robô irá dar uma ré e virar para a direita 90 graus, senão o robô andará para frente.***

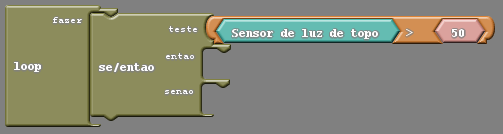
**1.6 Sensor de luz de topo azul e vermelha, senão verde**

#### **Etapas**

**1.6.1. Sentando o SENSOR**

Para a programação desejada utilize **Controle** e a seguir selecione, arraste e encaixe a função **se/então** no Loop;

* Clique no menu **teste** e escolha a função **maior que** (>) e encaixe em teste;
* Na **Biblioteca ROBO+EDU** selecione a função **Sensor de luz de topo** e encaixe no lado esquerdo da função **maior que** (>);
* Na função **Variáveis/Constantes** selecione a função **Constante inteira** e insira o valor **50**.



**1.6.2. Sentando os Leds RGB no menu então**

Selecione, arraste e encaixe no menu **então:**

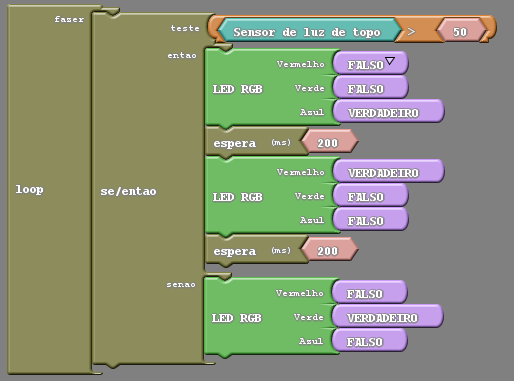
* **Led RGB** obtido na **Biblioteca ROBO+EDU** alterando para falso as cores vermelha e verde;
* No menu **Controle** selecione **espera (ms)** eencaixe abaixo da função Led RGB inserindo o valor **200**;
* Com o lado direito do mouse, em cima do Led RGB inserido, clique em **Clonar** e o arraste para baixo da função **espera (ms)** alterando a cor vermelha para verdadeiro e as cores verde e azul para falso;



**1.6.3. Sentando o Led RGB no menu senão**

Selecione, arraste e encaixe no menu **senão:**

* **O Led RGB** obtido na **Biblioteca ROBO+EDU** e altere para falso as cores vermelha e azul**.**



Com os procedimentos anteriores ***se o valor do sensor de luz de topo for maior que 50, então liga o led piscando nas cores azul e vermelha, senão liga o led na cor verde.***

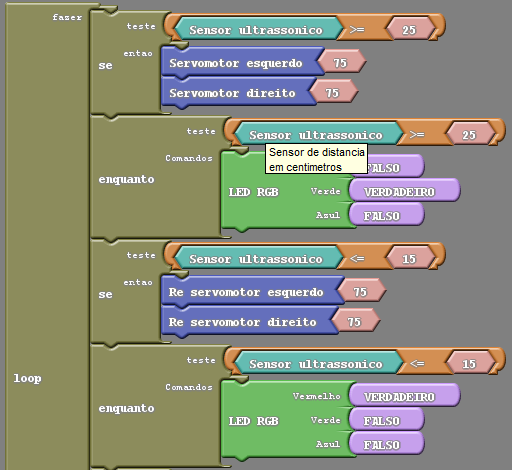
**1.7 Dog Inteligente com LEDs**

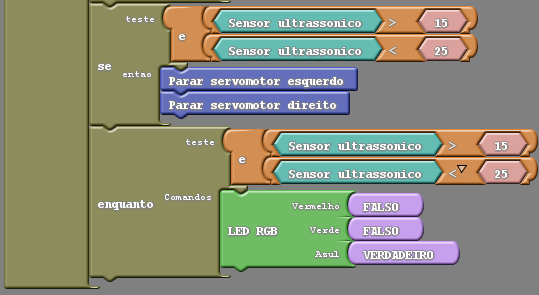
A programação abaixo consiste em:

* O Robô andar para **frente** e com a luz de topo **verde**, se o **sensor ultrassônico for maior ou igual a 25**;
* O Robô andar para **trás** e com a luz de topo **vermelha**, se o **sensor ultrassônico for menor ou igual a 15**;
* O Robô ficar **parado** e com a luz de topo **azul**, se o **sensor ultrassônico estiver entre 15 e 25.**

#### **Etapas**

Para isso, basta seguir as etapas apresentadas nas imagens.





Atividades A18 e A27 – Ardublock - Programação Gráfica para Arduino

* Formem duplas, pesquisem na internet outras programações em Ardublock, ampliem todas as programações efetuadas anteriormente e salvem, de forma conveniente, para apresentar aos demais grupos.

REFERÊNCIAS

[*https://pt.wikipedia.org/wiki/Rob%C3%B3tica*](https://pt.wikipedia.org/wiki/Rob%C3%B3tica)

<http://cienciaetecnologias.com/robotica-historia/>

<http://ciencia.me/WZJTu>

<http://wiki.icmc.usp.br/images/b/bc/Aula02-RMA-2011-Parte3.pdf>

<http://www.din.uem.br/ia/robotica/classif.htm>

<http://www.mecatronicaatual.com.br/educacao/797-controle-de-servo>

<http://www2.pcs.usp.br/~labdig/pdffiles_2014/controle-servo-semestral.pdf>

<https://lelinopontes.wordpress.com/tag/robos-da-antiguidade/>

<http://esabot.blogspot.com.br/2011/07/historia-da-robotica-seculo-xx-2a-parte.html>

<http://www.okconcursos.com.br/apostilas/apostila-gratis/130-informatica-para-concursos/1687-hardware-e-software-conceitos-basicos#.V3KrJo-cHIU>

<http://mundoeducacao.bol.uol.com.br/informatica/hardware-software.htm>

<http://www.robotizando.com.br/shields_index.php>

<http://canaltech.com.br/o-que-e/hardware/o-que-e-arduino/>

<http://www.mecanicaindustrial.com.br/598-o-que-e-um-sensor-ultrassonico/>

<http://www.unorobotica.com.br/docs/seguidor.pdf>

<http://powerquality.eaton.com/Brasil/Support/Documentation/BR-glossary.asp>

<http://www.asta.com.br/temperatura-conceitos.html>

<http://www.ecil.com.br/temperatura-industrial/>

<http://www.ebah.com.br/content/ABAAAAPOMAF/sensores-ldr>

<http://www.newtoncbraga.com.br/index.php/como-funciona/4883-art644>

<http://roboliv.re/conteudo/sensor-de-infravermelho>

<http://eletricamentefalando.blogspot.com.br/2011/09/sensor-infravermelho.html>

<http://grupocmaservicos.blogspot.com.br/2013/02/sensor-de-barreira-como-funciona_26.html>

<http://sensoreslabfisica.blogspot.com.br/>

<https://pt.wikipedia.org/wiki/RGB>

<http://www.comofazerascoisas.com.br/projeto-arduino-como-emitir-sons-com-o-buzzer.html>

<http://playground.arduino.cc/Portugues/LearningButton>

<http://www.servomotor.ind.br/o-que-e-servomotor.asp>

<https://pt.wikipedia.org/wiki/Linguagem_de_programa%C3%A7%C3%A3o>

<https://www.circuitar.com.br/tutoriais/ardublock-programacao-grafica-para-arduino/>

<http://www.robotizando.com.br/curso_arduino_o_que_e_arduino_pg1.php>

http://blog.vidadesilicio.com.br/apostila-arduino-basico/

http://www.leomar.com.br/brinquedos/images/stories/manuais/laboratorio/guia%20de%20robotica.pdf

http://paginapessoal.utfpr.edu.br/nacamura/oficinas\_2014\_2/oficinas-de-integracao-i-2014-2/Modulo6Piezo.pdf

**Bom Estudo!**