



# ARDUINO KIDS



A P O S T I L A K I T  
**ARDUINO KIDS**

V1.5 – Setembro/2025

Olá, Maker!

Parabéns pelo primeiro passo rumo a uma jornada incrível no mundo da tecnologia e da criatividade!

Esta apostila foi preparada especialmente para você e servirá como um guia completo, unindo teoria e prática, para aproveitar ao máximo o seu [Kit Arduino Kids](#). Com todos os componentes essenciais inclusos, o kit garante que você coloque cada projeto em prática e viva uma jornada de aprendizado completa.

Se você ainda é novo no universo Arduino, não se preocupe! Vamos te guiar desde os primeiros passos: Apresentação da plataforma, instalação, configuração e o passo a passo para você dominar todos os recursos da sua placa Arduino. E o melhor: com projetos práticos que vão te levar da programação à montagem, até ver o seu projeto ganhar vida!

Com o Kit Arduino Kids, as possibilidades são infinitas. Você poderá criar inúmeros projetos utilizando sensores, módulos, e muito mais. Isso e muito mais você encontrará no site da [Eletrogate](#).

Então, se você ainda não garantiu o seu, não perca mais tempo! Acesse agora mesmo o nosso site e garanta já o seu [Kit Arduino Kids](#) para desbloquear todo o seu potencial criativo! Tudo o que você precisa para soltar sua criatividade, aprender e se divertir você encontrará neste Kit.

Prepare-se para uma aventura repleta de aprendizado com conteúdo de qualidade e projetos práticos. Estamos ansiosos para ver tudo o que você vai criar.

Aproveite bastante e bons estudos!

A P O S T I L A K I T  
**ARDUINO KIDS**

**Itens inclusos no Kit:**

	<b>1</b>	Uno R3 + Cabo Usb para Arduino
	<b>1</b>	Ponte H Dupla L298N
	<b>1</b>	Micro Servo 9g SG90
	<b>1</b>	Módulo Sensor de Distância Ultrassônico HC-SR04
	<b>10</b>	Jumpers - Macho/Femea
	<b>1</b>	Protoboard 400 Pontos
	<b>2</b>	Led Difuso 5mm Vermelho
	<b>2</b>	Led Difuso 5mm Verde
	<b>2</b>	Led Alto Brilho 5mm Branco
	<b>10</b>	Resistor 10K 1/4W
	<b>10</b>	Resistor 330R 1/4W
	<b>1</b>	Kit Chassi 2WD (2 rodas) Robô para Arduino
	<b>1</b>	Módulo Bluetooth HC-06
	<b>1</b>	Buzzer Ativo 5v
	<b>1</b>	Suporte Para Sensor Ultrassônico HC-SR04
	<b>10</b>	Jumpers - Macho/Macho
	<b>1</b>	Adaptador Bateria 9v com Plug
	<b>2</b>	Led Difuso 5mm Amarelo
	<b>2</b>	Led Difuso 5mm Azul
	<b>10</b>	Resistor 22K 1/4W
	<b>5</b>	Push Button (Chave Táctil) 6x6x5mm

A P O S T I L A K I T  
**ARDUINO KIDS**

## Apresentação

Esta apostila tem o intuito de disseminar a cultura maker e o aprendizado em robótica de forma divertida, com o público voltado para crianças acima de sete anos e adolescentes que possuem interesse em aprender um pouco mais sobre tecnologias modernas na área de eletrônica. Os conteúdos abordados são focados no ensino prático, sem perder os detalhes teóricos essenciais.

É importante ressaltar que para crianças mais novas, é aconselhável a supervisão de um adulto, uma vez que fios metálicos podem ser perfurantes, e algumas montagens erradas podem ocasionar em danos aos componentes.

Em um dos exemplos de uso desta apostila é necessário fazer a soldagem em terminais de componentes eletrônicos. Nesta apostila não há instruções sobre manuseio do ferro de solda, portanto, se você é criança ou não tem experiência, busque ajuda de alguém capacitado, pois o mau uso dos equipamentos de solda pode acarretar queimaduras e acidentes domésticos.

Esperamos que o leitor tenha um excelente aprendizado e que consiga extrair o melhor de seus componentes, elaborando novas montagens de forma criativa e testando novos componentes.

A P O S T I L A K I T  
**ARDUINO KIDS**

## Sumário

<b>Apresentação .....</b>	4
<b>Sumário .....</b>	5
<b>Parte I - Conhecendo as ferramentas básicas .....</b>	6
<b>Introdução .....</b>	6
<b>Entenda o que é uma protoboard .....</b>	6
<b>Entenda o que são os jumpers .....</b>	9
<b>Observando os diagramas de ligação .....</b>	9
<b>Entendendo o que é o Arduino .....</b>	12
<b>Entendendo o que é a programação .....</b>	12
<b>Instalando os programas para começar a programar .....</b>	13
<b>Primeiros passos no Ardublock .....</b>	26
<b>Parte II – Seção de Exemplos Práticos .....</b>	29
<b>Exemplo 1 - Sirene de Bombeiro .....</b>	29
<b>Exemplo 2 - Semáforo de Carros .....</b>	34
<b>Exemplo 3 - Mini Piano com Sons .....</b>	36
<b>Exemplo 4 - Sensor de Proximidade para Cegos .....</b>	40
<b>Exemplo 5 - Cancela Inteligente Controlada por Smartphone .....</b>	44
<b>Exemplo 6 - Construindo um Carrinho Autônomo .....</b>	50
<b>Considerações finais .....</b>	65

## Parte I - Conhecendo as ferramentas básicas

### Introdução

Olá, aventureiro(a) e curioso do universo Arduino! Estamos bastante felizes em saber que você está interessado em iniciar esse novo percurso. Aqui começaremos a nossa jornada no objetivo de aprender o que é o Arduino e como usar ele para fazer e/ou melhorar os nossos próprios brinquedos, e ver como ele pode nos ajudar a fazer muito mais coisas.

Acredito que, provavelmente, goste de videogames ou já jogou pelo menos por algumas vezes, certo? Nos jogos a gente sempre evolui quando ganhamos novas habilidades e ferramentas, por exemplo, no Minecraft podemos conseguir picaretas de diamantes que vão nos ajudar a minerar rapidamente e conseguir blocos, e com esses blocos iremos precisar de nossas habilidades para conseguir construir grandes monumentos; essas habilidades vamos ganhando praticando, e com o tempo.

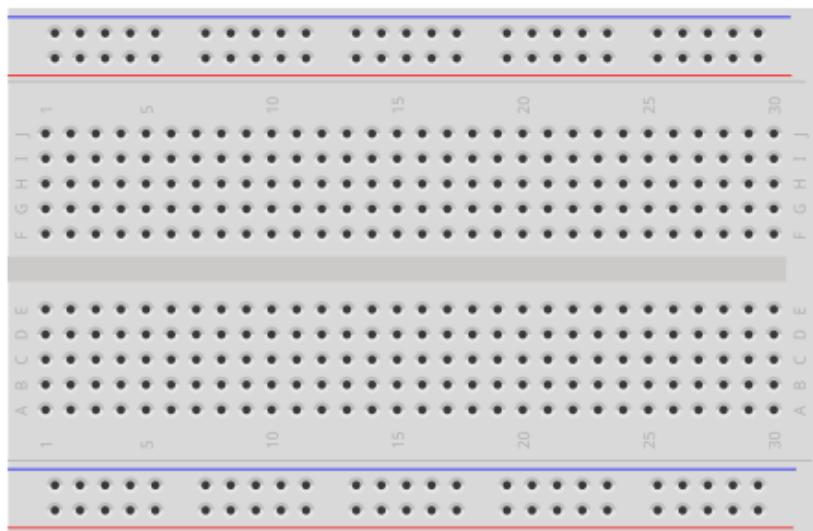
Na vida real é igual o jogo, e iremos avançar só com novas habilidades e ferramentas; e hoje, vamos conhecer uma nova ferramenta: o arduino e alguns de seus módulos, e iremos começar a desenvolver duas habilidades: a de programar o arduino, e a de montar circuitos. No final desta apostila teremos como recompensa um dos carrinhos mais legais visto, e feito por você mesmo, utilizando as ferramentas e as habilidades adquiridas.

Mas antes de sair criando tudo, iremos precisar conhecer algumas coisas básicas, que serão as nossas espadas e escudos, como verdadeiras ferramentas de sobrevivência no mundo da eletrônica, e nessa jornada precisaremos entender o que são algumas coisas.

### Entenda o que é uma protoboard

A protoboard é um componente que serve para a prototipagem. Ela é algo parecido com essa imagem abaixo. E tem um funcionamento vital para nossas montagens: ela serve para unir os fios e terminais dos outros componentes, só encaixando nos seus furos.

A P O S T I L A K I T  
**ARDUINO KIDS**

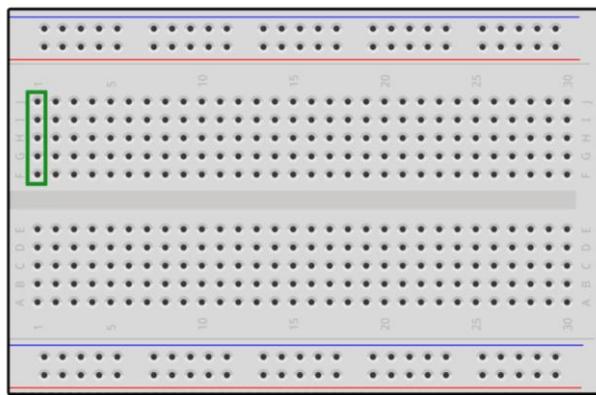


Graças à protoboard todo o circuito consegue estar conectado e funcionando, sem que a gente precise fazer nenhuma emenda de fio ou soldar, tudo simples. Ela funciona da seguinte forma:

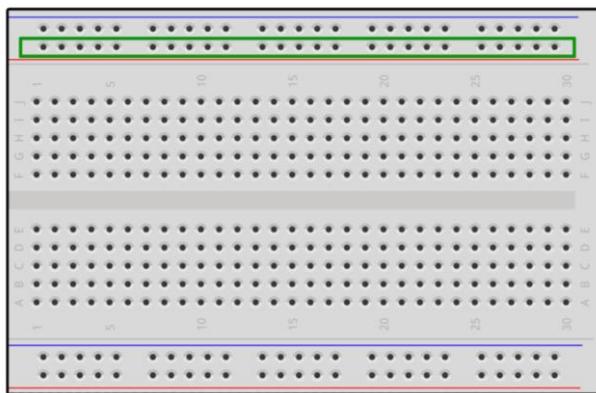


Toda linha vertical que vem 5 pontos unidos, como essas 5 marcadas na imagem, está conectada uma a outra fisicamente. A fins de curiosidade, internamente há uma ligação física de cobre, o que significa que todos os pontos são como se fossem um só, uma emenda perfeita, não importando se está ligando um fio no primeiro ponto e outro no último. Mas vale frisar: a emenda só vale entre 5 furos juntos quando estão na vertical.

A P O S T I L A K I T  
**ARDUINO KIDS**



Nos pontos que ficam na extremidade e que tem os furos juntos na horizontal, o cenário é diferente:



Assim, esses pontos da extremidade não são ideais para unir coisas referentes aos componentes do circuito, e sim colocar os fios de energia do circuito, porque vão facilitar e muito nossa vida. Nessa apostila, observe com paciência como a protoboard está sendo usada, e você vai notar coisas preciosas.

## Entenda o que são os jumpers

Os jumpers são esses fiozinhos coloridos que vem no kit.



Eles servem como o braço direito para as protoboards e os Arduinos, porque são eles que fazem todo o serviço de ligação dos componentes com o Arduino e fazem as coisas acontecerem. Eles são como as veias e artérias do nosso corpo, que levam sangue a todas as partes e conectam tudo, mas que no caso dos jumpers o seu verdadeiro papel deles é levar a energia para os componentes e fazer as comunicações.

Os jumpers serão usados conectando as portas do Arduino nos componentes, e a protoboard fará a união disso tudo para a gente. Se ficou um pouco confuso, não se preocupe, logo a frente terá exemplos práticos para ficar muito mais claro.

## Observando os diagramas de ligação

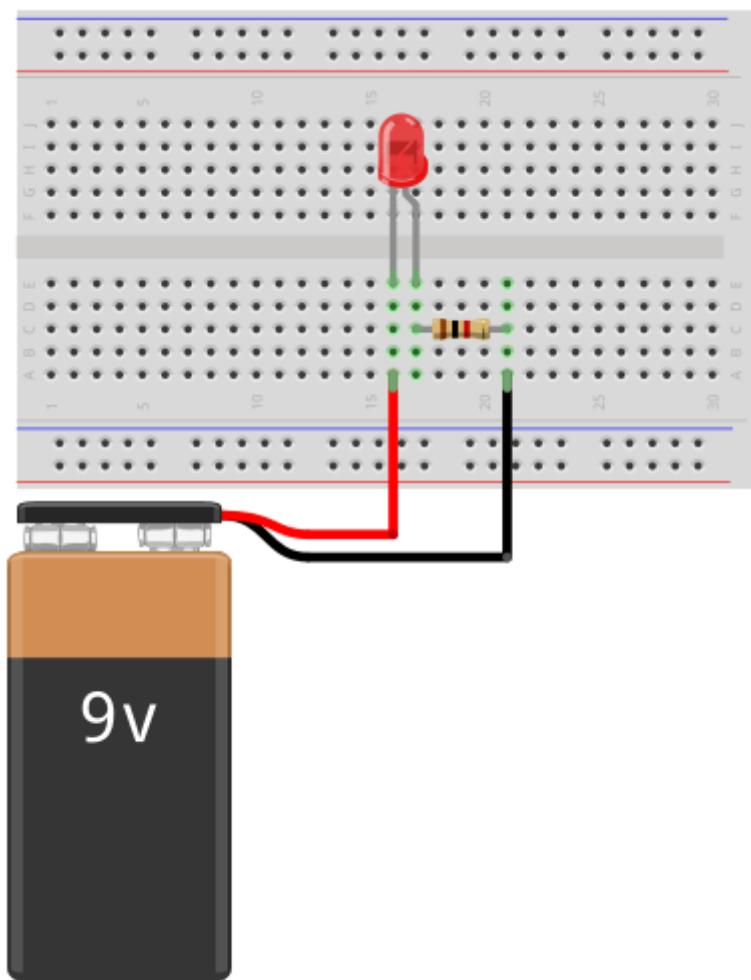
Os diagramas de ligação são maneiras que encontramos para desenhar nossos circuitos e ficar fácil de montar. No exemplo abaixo, nós iremos mostrar um diagrama de como ligar um LED com uma bateria, e para reforçar nosso aprendizado, vamos mostrar também como seria o mesmo circuito se não houvesse protoboard, onde teríamos que emendar os fios.

Observe com muita atenção o diagrama abaixo, porque apesar de ser simples, podemos ver o funcionamento na prática de uma protoboard. Se sentir à vontade, monte também e veja

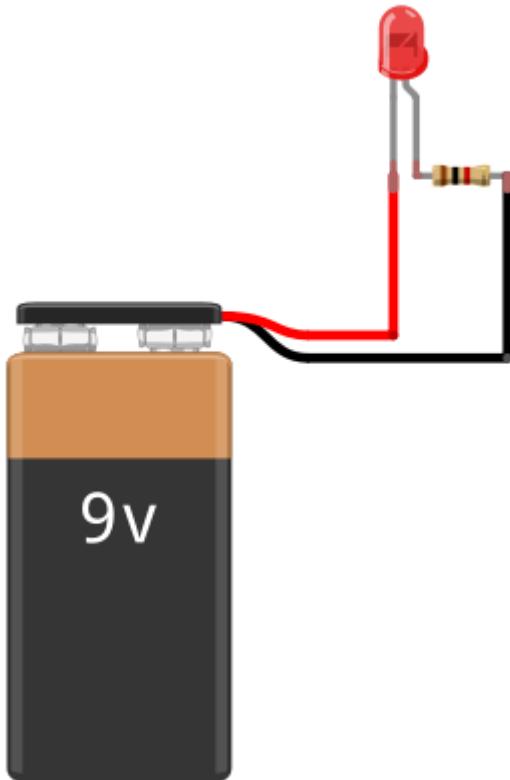
A P O S T I L A K I T  
**ARDUINO KIDS**

funcionando, só coloque o resistor de 1K, que é o que tem as cores de linha marrom, preto e vermelho.

(Observação: se caso não funcionar, inverta os terminais do LED).



Nessa montagem, de modo simplificado, podemos imaginar a energia como bolinhas que percorrem os fios. Todo circuito possui umas regrinhas, e a primeira regrinha é: a energia sempre deve voltar pro mesmo lugar que saiu, mas sempre pelo **caminho diferente do que ela veio!** Se essa regra não for cumprida, o circuito não funcionará. Pensar assim nos ajudará bastante a entender porquê de o LED ligar. Neste diagrama a energia (pode imaginar as bolinhas) sai do fio vermelho da bateria de 9V, e usa a protoboard como ponte para chegar até o LED. A energia passará pelo primeiro terminal (fio) do LED, irá caminhar para dentro do LED e sairá no outro terminal. Depois disso, passará pelo resistor, seguindo o caminho em direção ao fio preto, e voltará para onde veio, pelo fio preto (caminho diferente).



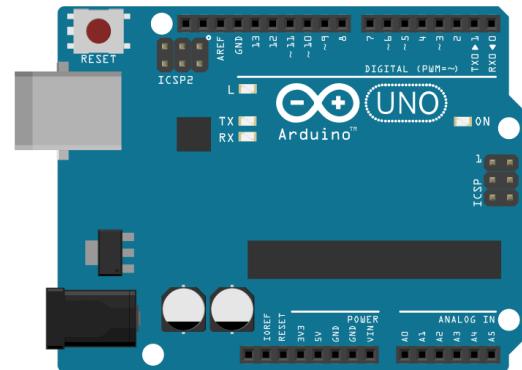
Se não existisse protoboard, teríamos que fazer a montagem assim como na imagem acima, emendando todos os fios, e o circuito também ficaria menos protegido, e com a aparência de mais frágil. Por isso a protoboard é tão importante.

Nesse exemplo, vimos como ligar um LED, mas e se quisermos que esse LED funcione de maneira automática? Piscando em momentos específicos e sem a gente precisar colocar a mão?

Por esse motivo iremos entender e aprender a usar o Arduino! Ele irá nos permitir fazer essas automações e projetos que conseguem fazer verdadeiros robôs que fazem coisas por nós.

## Entendendo o que é o Arduino

Falando de maneira simples, o Arduino é uma placa com circuitos integrados e mais alguns outros componentes eletrônicos que acompanha o kit Arduino Kids, e é como essa que está ao lado. Ela permite que seja controlada, através de comandos nos circuitos, e também verificar o que está acontecendo através de medições. Com ela, a gente pode criar novos projetos dando vida às coisas. Ele dá vida às coisas através de pinos, esses que possuem números e que ficam em cima e embaixo da placa.



fritzing

O maior circuito integrado (componente preto retangular com várias perninhas) da placa Arduino é o **Microcontrolador**. Ele é o componente que controla tudo.

Possuímos pinos que podemos programar os de 0 a 12, que estão em cima, e esses A0 a A5 que ficam embaixo. Estes pinos são importantes porque são com eles que vamos poder programar para dar vida aos nossos projetos.

A placa também possui alguns outros pinos que são interessantes e importantes, e eles são os 5V, GND, que ficam na parte debaixo e meio a esquerda. Eles servirão para alimentar nossos circuitos, eles terão papel vital para a maioria dos circuitos, dando a energia necessária para fazer as coisas funcionarem. O pino de 3.3V é importante também, e faz o mesmo papel do 5V, o uso dele é para componentes que não suportam 5V, e precisam de algo menor, como 3.3V, ou para alguns outros usos.

## Entendendo o que é a programação

Para programar o Arduino, nós usamos uma **linguagem de programação**, e essa linguagem é a forma que conseguimos para se comunicar com a placa, já que o arduino ainda não é capaz de ouvir e entender nossos comandos através de linguagem oral ou ler algum texto escrito.

O microcontrolador, como nós humanos, fala uma linguagem específica. Nós, por exemplo, estamos falando **português**, mas existem diversas outras linguagens para se comunicar, como: inglês, alemão, russo, francês, japonês. Como falamos em português, precisamos que para alguém converse conosco, ele entenda nossa língua, e use uma lógica coerente para

poder falar (não bastando apenas saber falar as palavras certas). Por exemplo, a frase: “Vamos fazer um super projeto hoje!”, não seria entendível se mudasse algumas palavras de lugar, tipo: “Fazer projeto vamos um super hoje!”, ficaríamos sem compreender e se fosse um comando, pode ser que faríamos a coisa errada. Com a programação é a mesma coisa, para programar a gente precisa de um **ALGORITMO**. Algoritmo é diferente de Algarismos, fechou? **Um algoritmo é um passo a passo de como fazer alguma coisa.** Por exemplo: uma receita de um doce de pudim, que podemos encontrar no Google, é um algoritmo! Lá podemos ver um passo a passo do tipo:

- 1º Separe uma xícara de chá de açúcar, um leite condensado, 3 ovos
- 2º Coloque a açúcar numa panela de fundo largo e deixe-a derreter por até ficar dourado
- 3º Coloque meia xícara de chá de água quente no açúcar derretido e mexa com uma colher até engrossar
- 4º Despeje essa calda feita no fundo de uma forma para pudim.
- 5º Bata em um liquidificador o leite condensado e os 3 ovos e depois de batido despeje na forma.
- 6º Cubra com papel-alumínio e leve ao forno em 180°, em banho maria, por cerca de 1 hora e 30 minutos.
- 7º Coloque na geladeira e espere por cerca de 7 horas
- 8º Tire da forma e coloque em um preto apropriado, e está pronto.

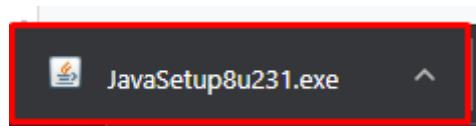
Viu como é simples compreender um algoritmo? A receita de pudim vem cheia de passos a passos, que basta seguir e fazer. O Arduino e qualquer máquina programável (como o seu computador, celular etc.) também segue os passos a passos, a única diferença está em como escrevemos esse passo a passo. No caso das máquinas, precisamos seguir uma **lógica exata** e uma estrutura específica na hora de digitar os comandos. Nessa apostila não iremos aprofundar no ensino de programação, aqui aprenderemos uma programação que é visual, feito montando blocos! Como se fossem blocos coloridos de montar brinquedos. Aqui, nesta apostila, iremos mostrar nos exemplos alguns códigos montados e você poderá brincar com eles para ajudar no aprendizado, modificando-os, e personalizando do jeito que achar melhor.

## Instalando os programas para começar a programar

Para tudo executar corretamente, iremos precisar do programa do Arduino instalado no computador e de outro programa que se chama Java Runtime, esse Java tem a função de

A P O S T I L A K I T  
**ARDUINO KIDS**

fazer uma ferramenta que iremos usar funcionar dentro do programa do Arduino funcionar, porque sem ela, não será possível fazer os nossos exemplos da apostila. Então, iremos começar a instalação pelo Java. Para baixar o Java, basta digitar no navegador de internet: <http://bit.ly/javaeg>. E o download



Uma tela como esta deverá se abrir, clique em “Instalar”.

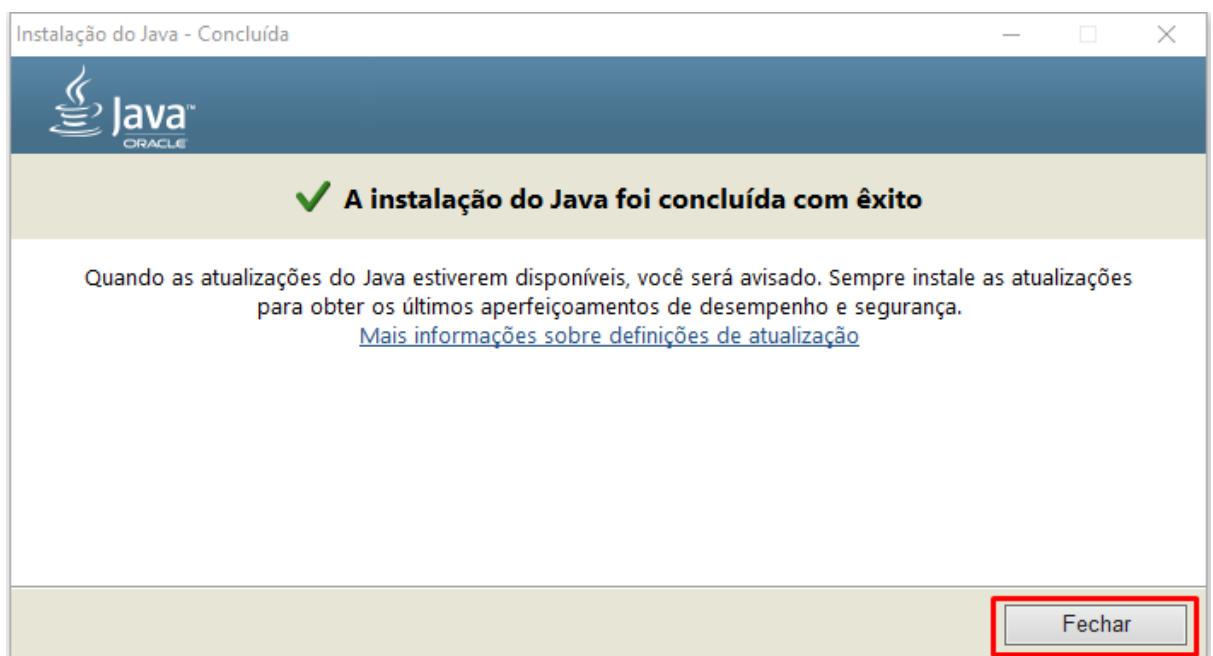


E virá uma tela com uma barra mostrando a instalação, só aguardar.

A P O S T I L A K I T  
**ARDUINO KIDS**



Quando concluir, só apertar em fechar.



Com o Java instalado, vamos agora fazer o download dos programas referentes ao Arduino. Antes de mais nada, precisamos instalar a versão mais nova da Arduino IDE, apenas para garantirmos que o driver (programinha responsável por fazer o computador conversar com o Arduino) esteja em sua versão mais nova.

A P O S T I L A K I T  
**ARDUINO KIDS**

Vamos começar sua instalação fazendo o download do instalador, que pode ser encontrado no [site oficial – clicando aqui](#).

Recomendamos sempre que você instale a versão estável mais recente disponível para seu sistema operacional. Como estamos utilizando o Windows 10 em nosso computador, vamos selecionar a opção **Windows Win 10 and newer, 64 bits**:

## Downloads



The screenshot shows the Arduino IDE 2.3.3 download page. On the left, there's a thumbnail of the IDE icon and the text "Arduino IDE 2.3.3". Below it is a brief description of the new features. In the center, there's a link to the documentation and a note about nightly builds. On the right, there's a "DOWNLOAD OPTIONS" section with links for Windows (highlighted with a red box), Linux, and macOS. Below that is a "Release Notes" link.

A tela seguinte nos propõe fazer uma contribuição para a plataforma, pode clicar em “Just Download” – Aqui é de suma importância que seu navegador não traduza a página, pois essa opção não é exibida quando traduzida.

## Download Arduino IDE & support its progress

Since the 1.x release in March 2015, the Arduino IDE has been downloaded **87.253.218** times — impressive! Help its development with a donation.

\$3    \$5    \$10    \$25    \$50    Other

**CONTRIBUTE AND DOWNLOAD**

or

**JUST DOWNLOAD**

Na tela seguinte, clique novamente em “Just Download”.

A P O S T I L A K I T  
**ARDUINO KIDS**

## Stay in the Loop: Join Our Newsletter!

As a beginner or advanced user, you can find inspiring projects and learn about cutting-edge Arduino products through our **weekly newsletter!**

email \*

I confirm to have read the [Privacy Policy](#) and to accept the [Terms of Service](#) \*

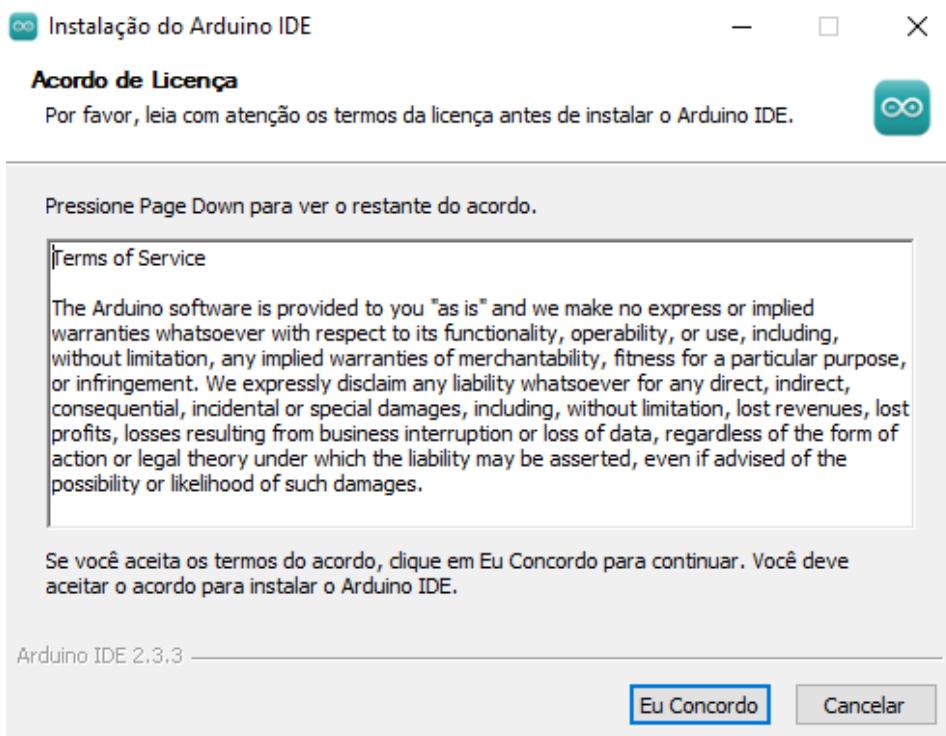
I would like to receive emails about special deals and commercial offers from Arduino.

**SUBSCRIBE & DOWNLOAD**

or

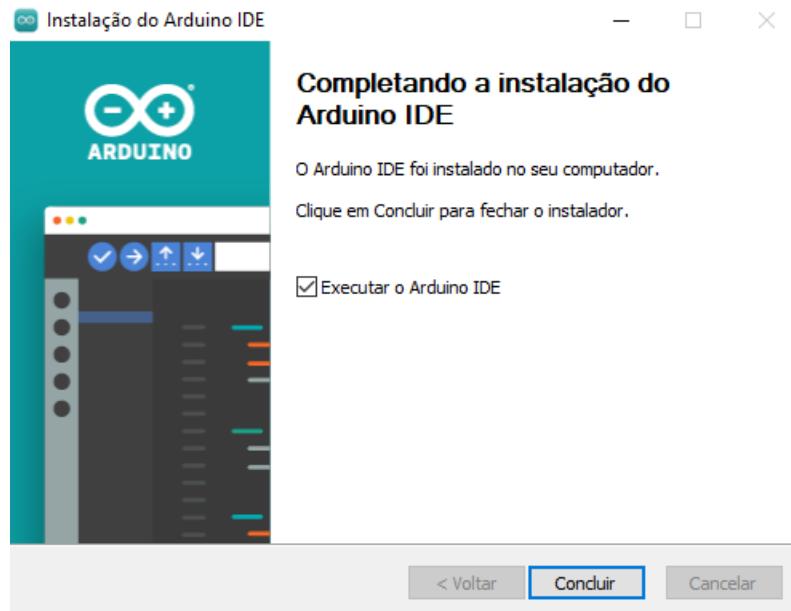
**JUST DOWNLOAD**

Após esses passos, o download irá iniciar automaticamente. Quando terminar, execute o arquivo baixado e clique em “Eu concordo” na janela exibida.



Nas telas seguintes, basta clicar em “Próximo” e “Instalar”. Ao término, clique em concluir para abrir a IDE.

A P O S T I L A K I T  
**ARDUINO KIDS**

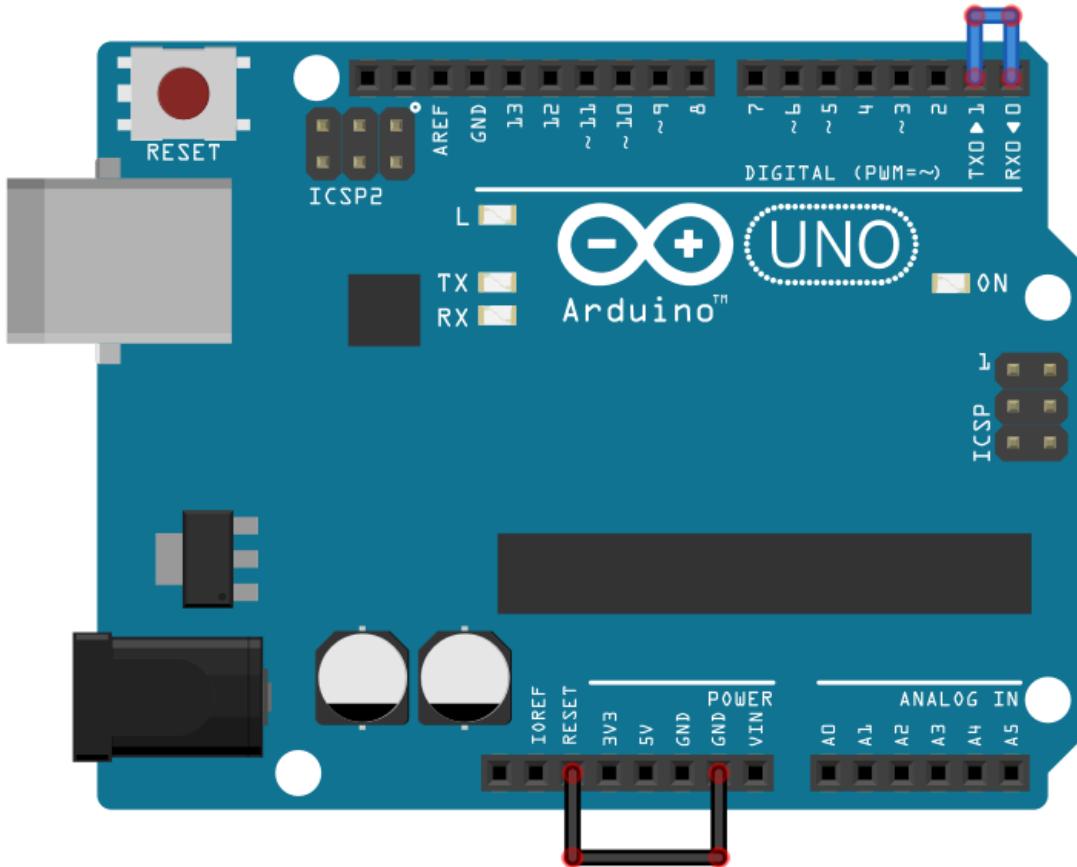


Aqui, pode ser que seja solicitada a instalação de alguns complementos, basta permitir a instalação. Se solicitada alguma confirmação relacionada ao acesso à internet, marque a caixinha referente às redes privadas e desmarque a caixinha das redes públicas, confirmando na sequência.

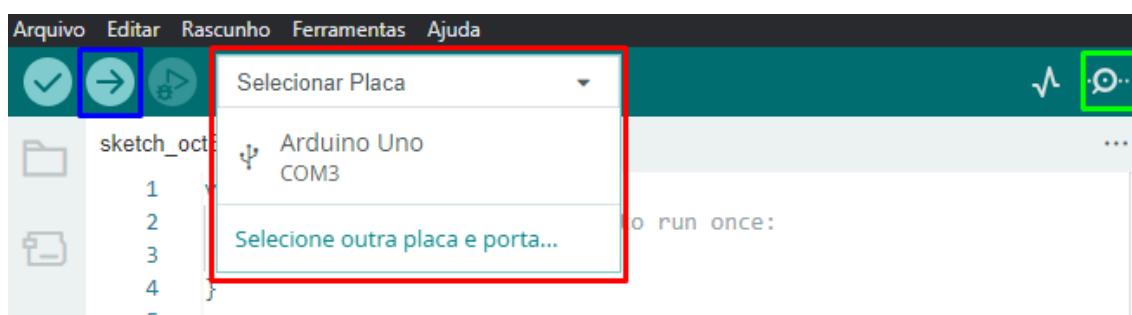
Com a IDE aberta, vamos fazer um simples teste para verificar o funcionamento da placa. Primeiro, precisamos conectar o pino RESET do Arduino ao GND e conectar os pinos 0 (RX) e 1 (TX) entre si, usando jumpers. Isso nos permite realizar um teste chamado loopback, que nada mais é do que um teste de comunicação entre o Arduino e o computador, onde enviamos uma mensagem e a placa nos retorna exatamente o que foi enviado se tudo estiver funcionando corretamente. Abaixo, o diagrama de conexão:

# A P O S T I L A K I T

# **ARDUINO KIDS**



Com a conexão feita, vamos conectar a placa ao computador e preparar a Arduino IDE. Nela, precisaremos apenas selecionar o modelo de placa e a porta COM em que seu Arduino está conectado e abrir um recurso chamado Monitor Serial. Veja abaixo onde encontrá-lo:



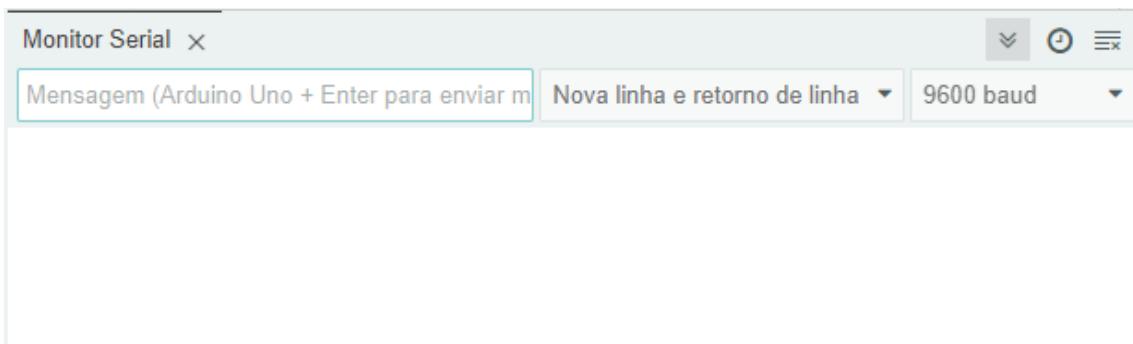
**Azul:** Botão “Upload”, usado para enviar o código para o Arduino;

**Vermelho:** Aqui você visualiza todas as placas conectadas ao computador. Selecione a que for correspondente ao seu Arduino;

**Verde:** Botão para abrir o monitor serial.

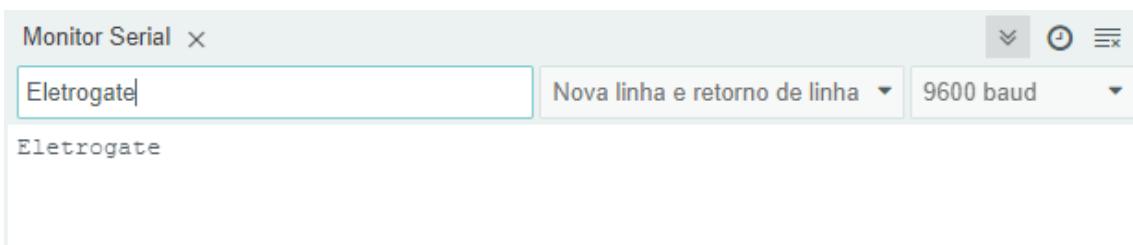
A P O S T I L A K I T  
**ARDUINO KIDS**

Especificamente nesse teste, não precisaremos carregar nenhum código, basta selecionar corretamente a placa e abrir o monitor serial. A seguinte aba será aberta na parte inferior da IDE:



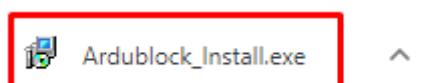
Nela, podemos usar o campo de texto para enviar comandos/mensagens para o Arduino e configurar a velocidade de comunicação.

Para fazer o teste de loopback, você só precisa digitar uma mensagem e apertar a tecla Enter, para enviá-la. O Arduino deverá retorná-la na sequência, como mostrado abaixo:



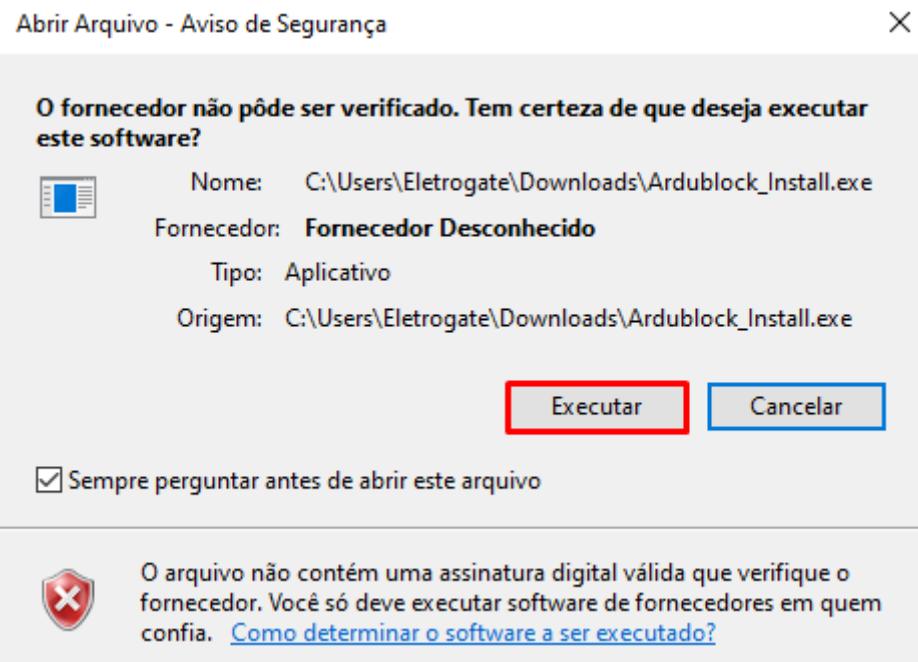
Se tudo funcionou corretamente, você está pronto para iniciar a montagem e programação dos exemplos! Digite no navegador de internet o link [http://apps.eletrogate.com/Ardublock\\_Install.exe](http://apps.eletrogate.com/Ardublock_Install.exe) para fazer o download do Ardublock (disponível somente para Windows), e aguarde o Download concluir.

Com o download concluído, clique em cima do nome do arquivo.

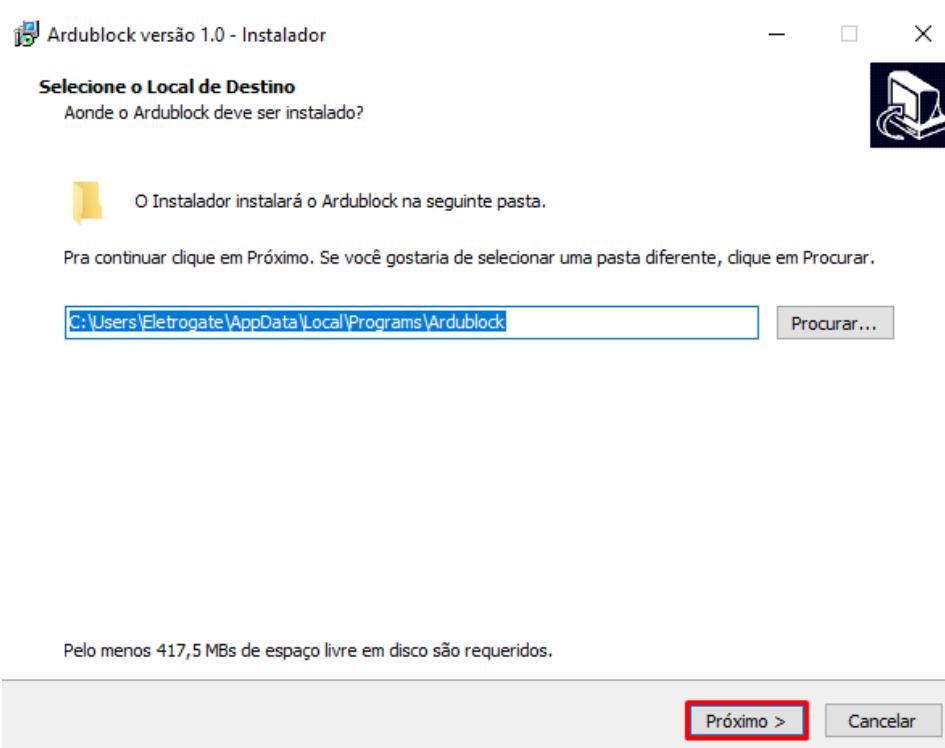


Uma mensagem como esta do Windows pode ser que apareça, e clique em executar.

A P O S T I L A K I T  
**ARDUINO KIDS**

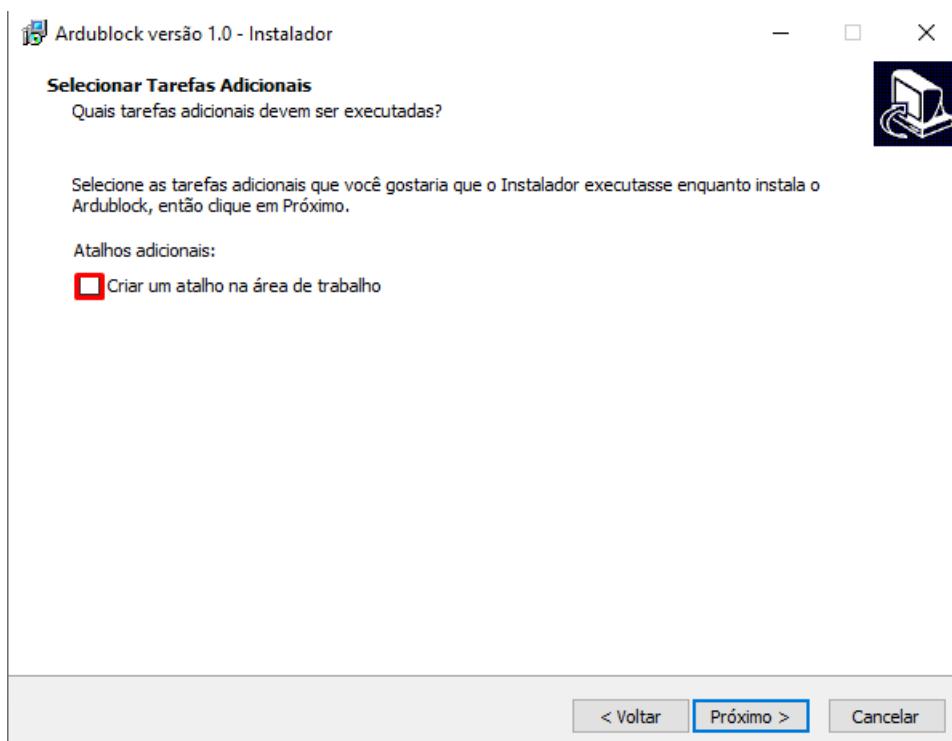


O instalador irá perguntar se você quer mudar o local de onde o programa será instalado, nessa etapa você pode dar próximo, como mostrado na imagem.



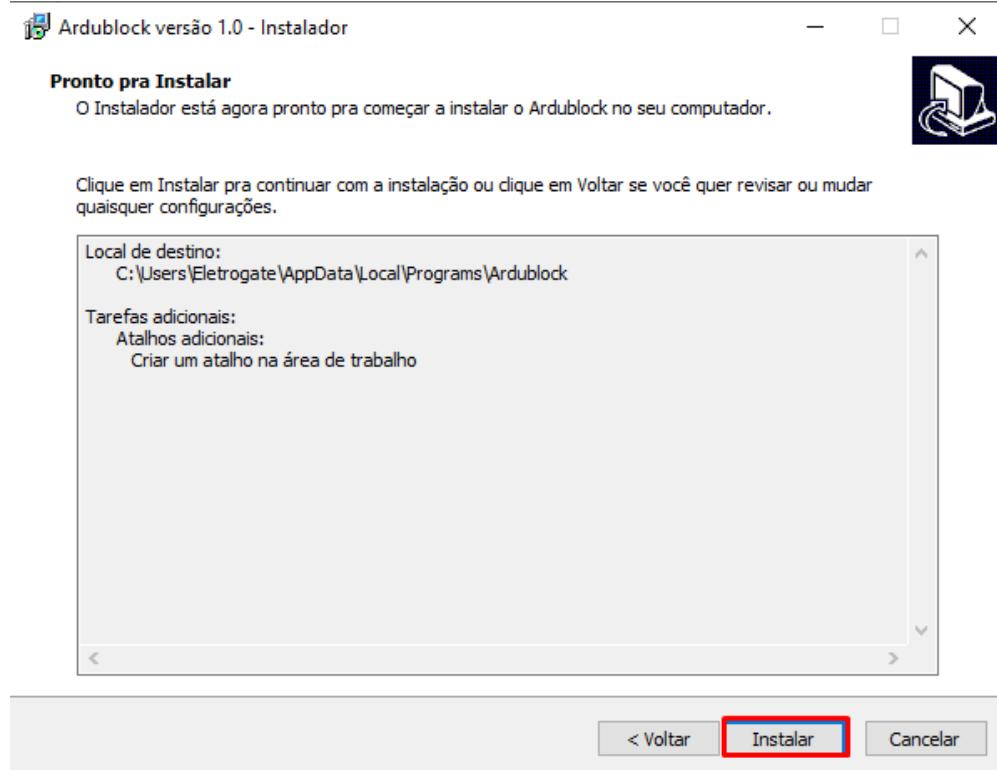
A P O S T I L A K I T  
**ARDUINO KIDS**

É importante marcar a opção de criar um atalho na área de trabalho, como na imagem abaixo, para ficar fácil de abrir o programa em outras vezes.



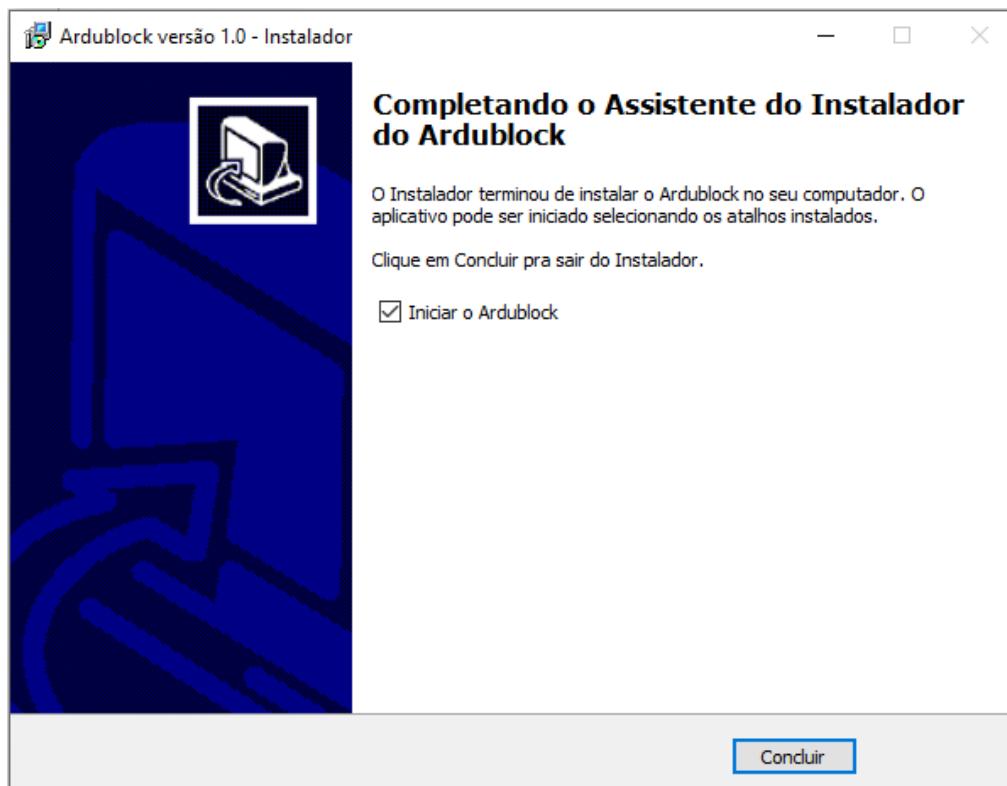
Confirme a instalação na opção instalar.

A P O S T I L A K I T  
**ARDUINO KIDS**



E o programa estará instalado! Só apertar concluir que o Arduino com o Ardublock irá abrir automaticamente.

A P O S T I L A K I T  
**ARDUINO KIDS**

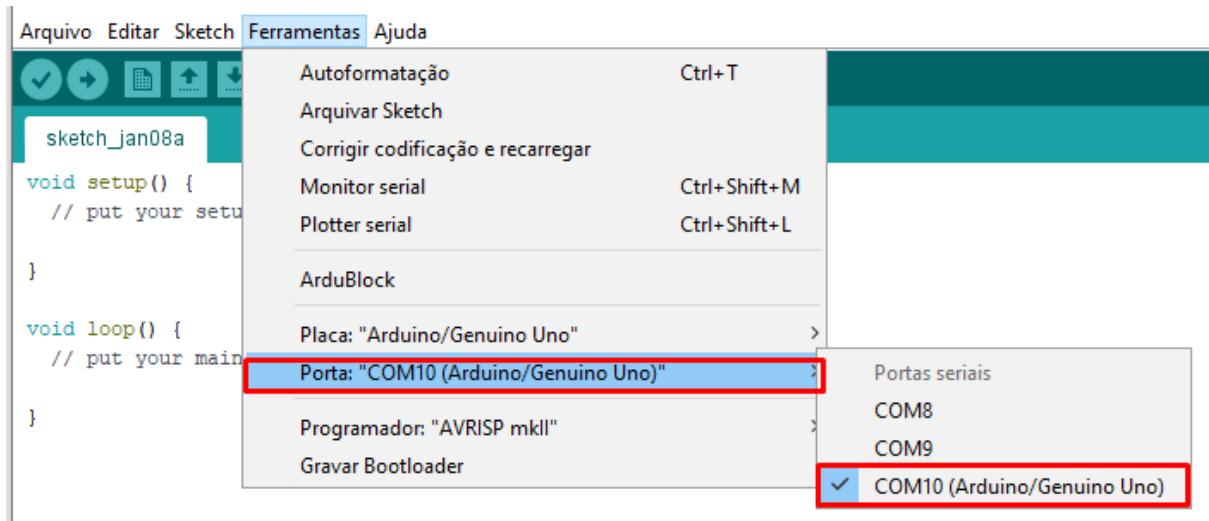


Veja que na área de trabalho já tem o ícone para abrir de maneira mais rápida!

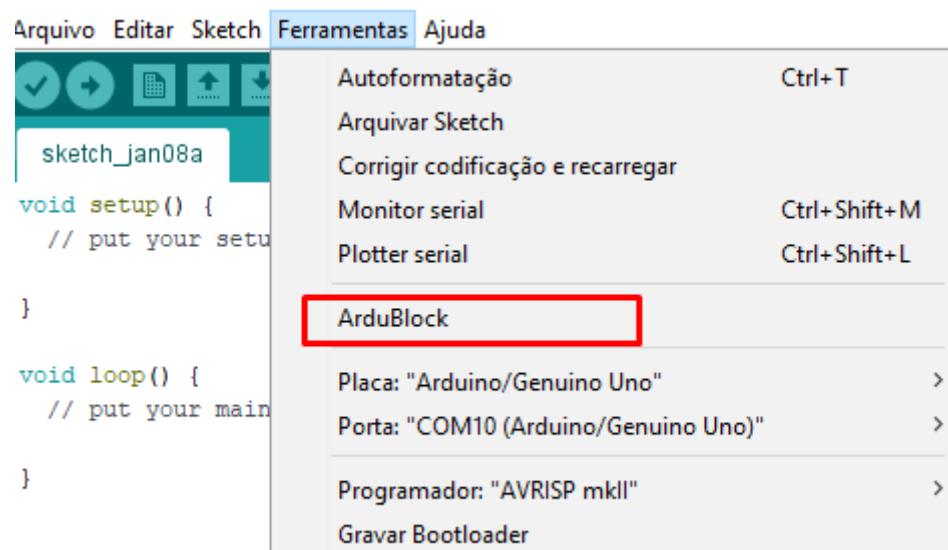


Abrindo o programa, vamos nos deparar com essa interface, como a imagem abaixo. A primeira coisa que sempre devemos fazer é: sempre que conectarmos a placa do Arduino ao computador, devemos abrir o programa do Arduino e ir em “Ferramentas>Porta” e selecionar a porta que vem escrito “Arduino/Genuino Uno” (Como na imagem abaixo). No meu caso, foi a porta COM10, mas no seu pode ser diferente. **É preciso sempre verificar se a porta correta foi selecionada, porque se não for, não irá funcionar.**

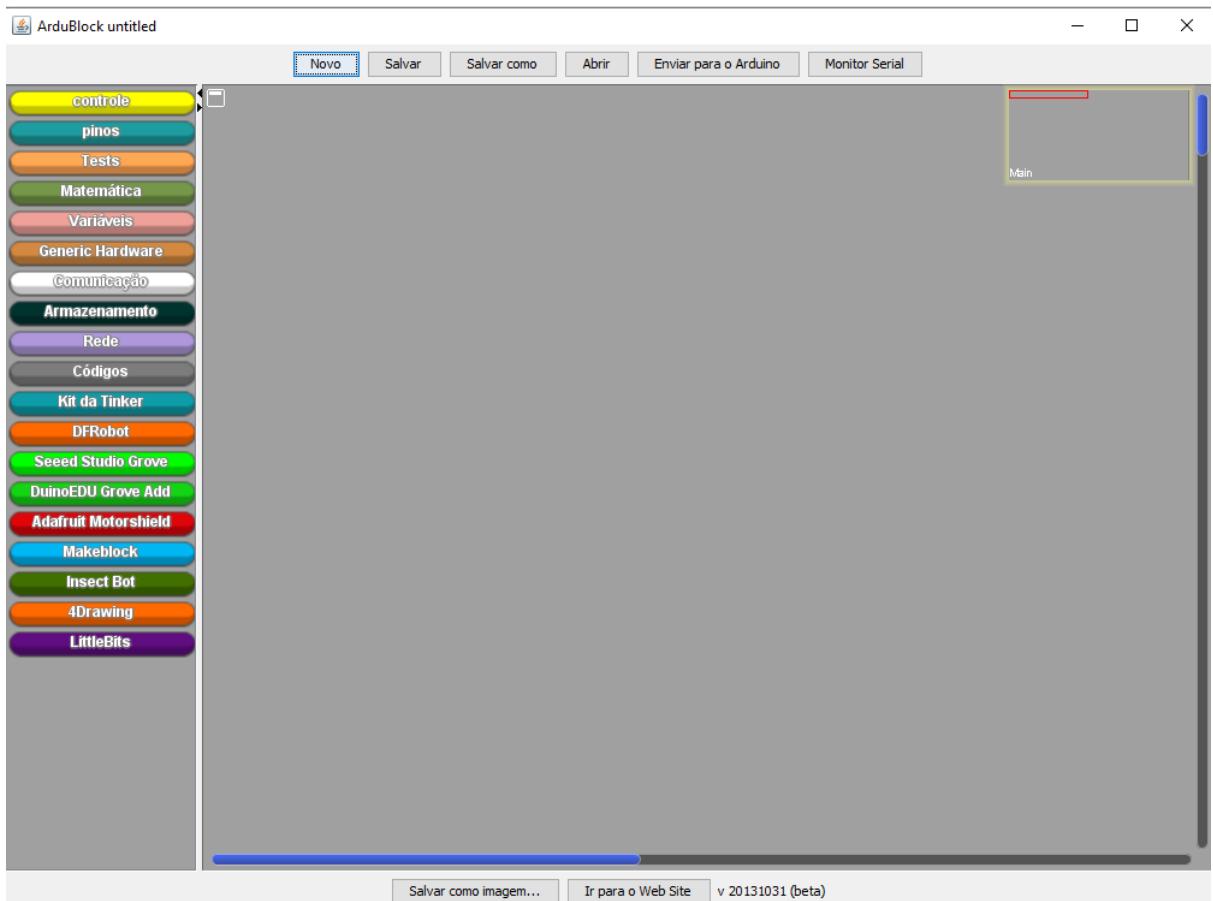
A P O S T I L A K I T  
**ARDUINO KIDS**



Agora com todos os passos feitos, estamos prontos para programar! Abra o Ardublock em “Ferramentas > Ardublock”



A P O S T I L A K I T  
**ARDUINO KIDS**



E algo dessa forma deve ter aparecido. E é o Ardublock que vai permitir que a gente programe o Arduino de maneira fácil e mais divertida.

## Primeiros passos no Ardublock

Nesta aba do programa, todos os blocos disponíveis para usarmos estão organizados nestes tópicos.

A P O S T I L A K I T  
**ARDUINO KIDS**



E quando clicamos em alguma destas partes, abrirão todos os blocos relacionados a ela. O exemplo abaixo mostra todos os blocos relacionados a “Controle” que podemos usar.

A P O S T I L A K I T  
**ARDUINO KIDS**



Percorra todos os blocos disponíveis no programa pelos tópicos, e veja quanta coisa tem que dá para juntar e fazer coisas diferentes.

O bloco “Program” ou o bloco “Sempre” são obrigatórios a serem usados, ou um ou outro. Eles são os blocos que vão abrigar todos os outros blocos dentro, por isso são obrigatórios ter.

Para montar os programas, basta arrastar os blocos para a área cinza da direita, e ir encaixando eles um abaixo do outro, como no exemplo abaixo. O computador irá fazer o que pedimos, lendo os blocos de cima pra baixo.



## Parte II – Seção de Exemplos Práticos

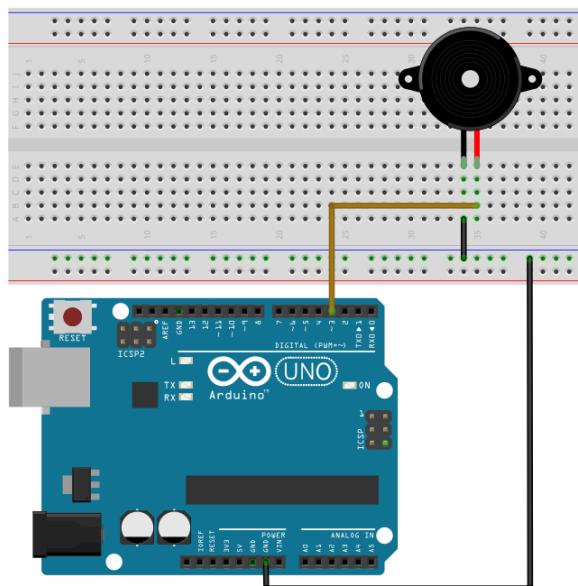
### Exemplo 1 - Sirene de Bombeiro

Como primeiro programa vamos montar um projeto bem simples e legal, que é fazer uma sirene de bombeiros. Iremos aprender também a como enviar o programa para a placa do Arduino e como salvar ele. Não pule este primeiro exemplo, será bem útil o aprendizado.

Para começar, vamos pegar um Buzzer, que é um componente que serve para emitir sons quando é acionado. Então, vamos usar ele para fazer uma sirene, e para isso iremos precisar de:

- Buzzer
- 3 jumpers macho/macho
- Arduino
- Protoboard
- Arduino e cabo de conectar no computador

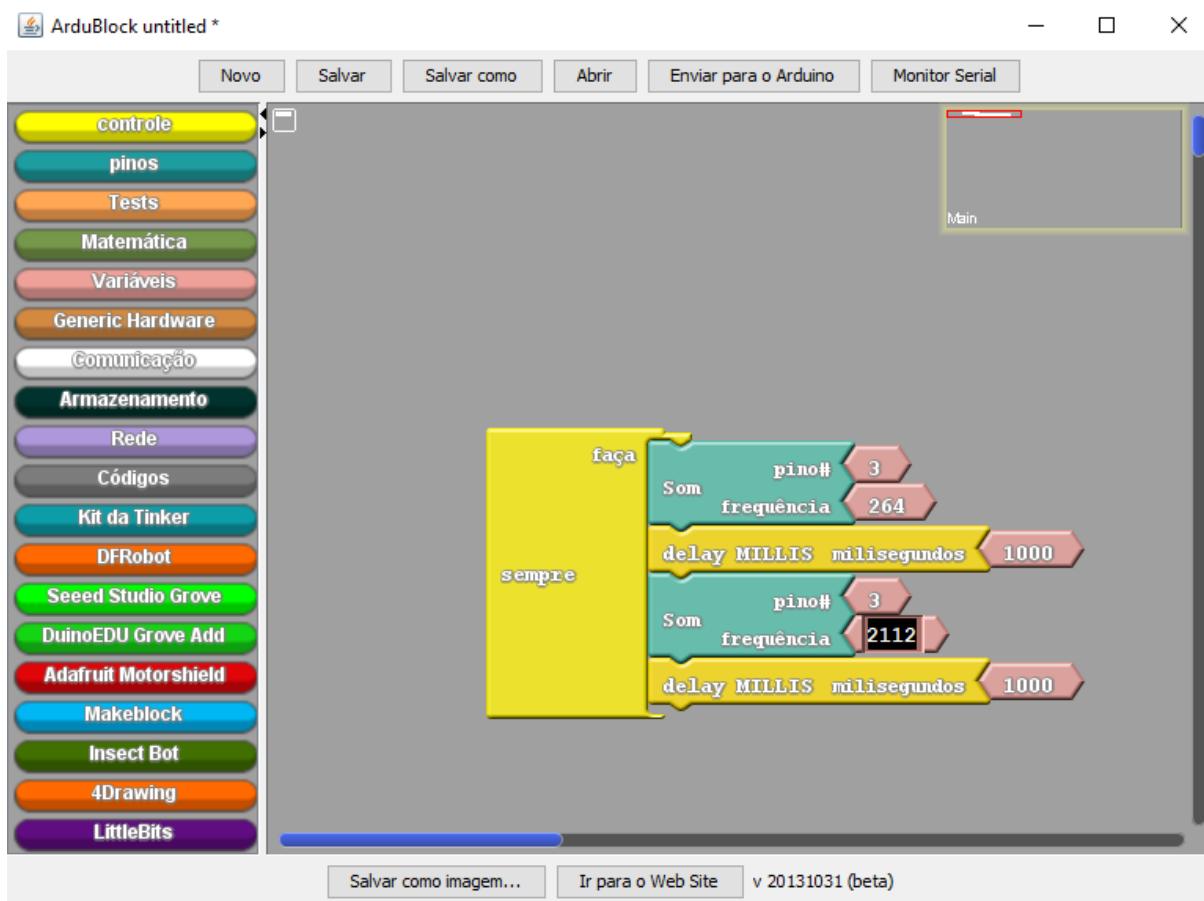
Fazendo a seguinte ligação:



Uma observação interessante é que o seu Buzzer não é igual ao da imagem, mas a ligação pode ser feita normalmente, pois, no Buzzer tem um +, que indica o terminal que é o positivo; nesse terminal, encaixe a porta 3, e no outro, o GND. Com isso, a montagem está concluída e iremos programar!

A P O S T I L A K I T  
**ARDUINO KIDS**

Abrindo o Ardublock, iremos iniciar um Novo projeto. Clique em Novo, no canto superior esquerdo, e vamos montar um código como este abaixo. Para achar os blocos amarelos, só clicar em “Controle” no canto esquerdo da tela, e ir sempre se orientando pelas cores. Quase sempre as cores do bloco indicam onde eles estão localizados, na parte esquerda. Para colocar os blocos na área de montagem, só arrastar eles.

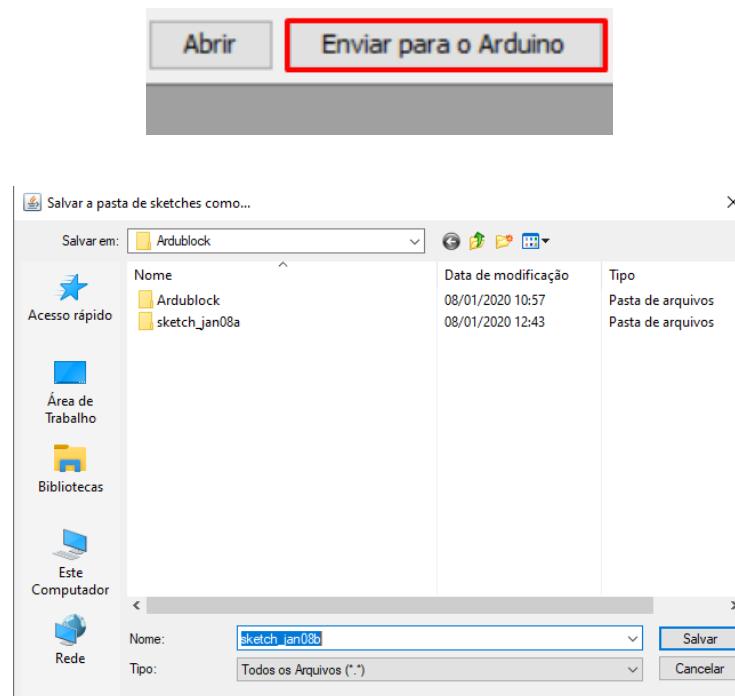


Como dito anteriormente, o bloco sempre e o bloco “sempre” e o bloco “program” são os mais importantes para nós. Porque um programa sempre irá precisar de algum dos dois, sendo que na maioria das vezes usamos o “sempre”. Isso porque o Arduino funciona com **rotinas**, mas o que são **rotinas**? Rotinas são tarefas específicas que o Arduino sempre ficará repetindo. Exemplo: Neste programa, o Arduino ficará sempre ligando o buzzer em um tom, espera um tempo, liga em outro tom, espera um tempo, liga em outro tom, e depois repete tudo de novo, e não para de repetir esses passos.

**Obs:** Com o bloco delay MILLIS, nós escolhemos quantos milissegundos a placa irá esperar antes de fazer o próximo comando. Neste caso, selecionamos 1000 milissegundos, que é igual a 1 segundo.

A P O S T I L A K I T  
**ARDUINO KIDS**

Com o código pronto, clique em “Enviar para o Arduino”, na parte superior da tela, e uma tela como esta irá aparecer:



Nesta tela devemos escolher algum lugar para salvar nossos projetos. Crie uma pasta nova para salvar os projetos, ou escolha alguma para a gente salvar. Salve o projeto na pasta escolhida, e o projeto será enviado para o Arduino.



```
sketch_jan08b
void setup()
{
}

void loop()
{
    tone(3, 264);
    delay( 1000 );
    tone(3, 2112);
    delay( 1000 );
}
```

Você pode observar que surgiram alguns textos aqui sem a gente precisar fazer nada. Esse é o grande papel do Ardublock, ele faz as coisas ficarem fácil de entender e fazer, e converte tudo depois para estes textos.

A P O S T I L A K I T  
**ARDUINO KIDS**

Na parte debaixo você pode observar se o programa foi carregado ou não. Se foi carregado, uma mensagem como esta irá aparecer:

```
Carregado.  
O sketch usa 2.628 bytes (8%) de espaço.  
Variáveis globais usam 32 bytes (1%)
```

Mas se apareceu uma mensagem laranja, semelhante a essa:

```
Problema ao carregar para a placa. Veja http://www.arduino.cc/en/Guide/Troubleshooting#upload para sugestões. Copiar mensagens de erro  
avrdude: ser_open(): can't open device "\\.\COM4": O sistema não pode encontrar o arquivo especificado.  
Problema ao carregar para a placa. Veja http://www.arduino.cc/en/Guide/Troubleshooting#upload para sugestões.  
13 Arduino/Genuino Uno em COM4
```

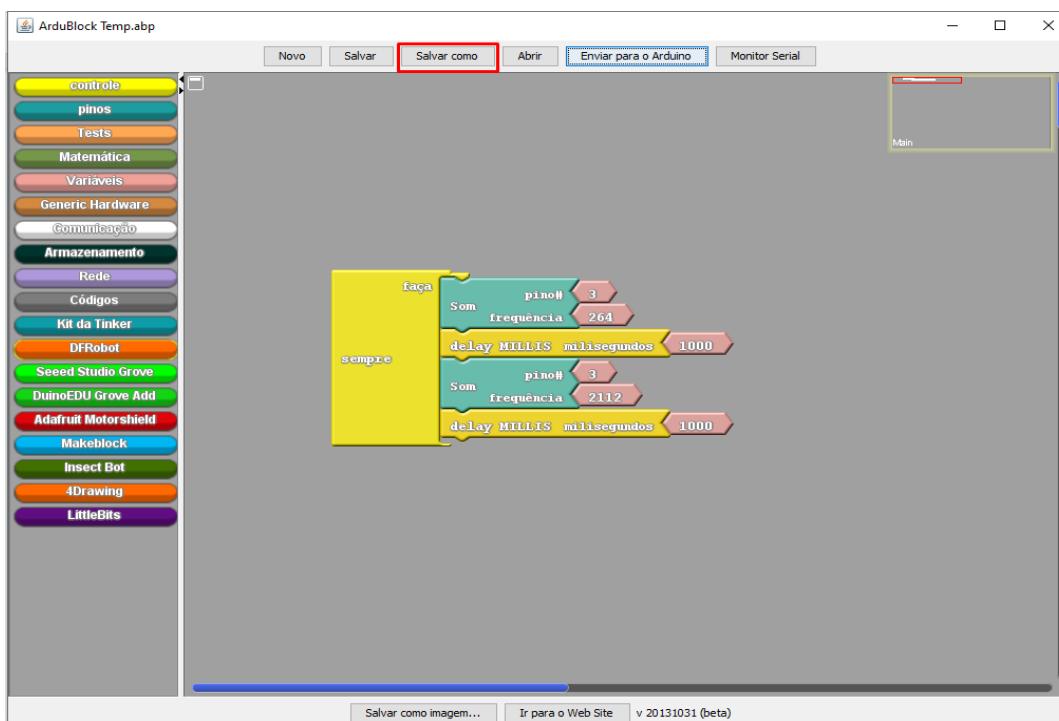
Não se desespere, é simplesmente porque a porta não foi selecionada como aprendido lá na parte de instalação dos programas. Faça a seleção da porta correta em Ferramentas e depois Portas e selecione a opção que tem escrito “Arduino/Genuino”.

Feito todos os passos, então, neste momento, uma sirene dos bombeiros deverá estar funcionando. Legal, né?

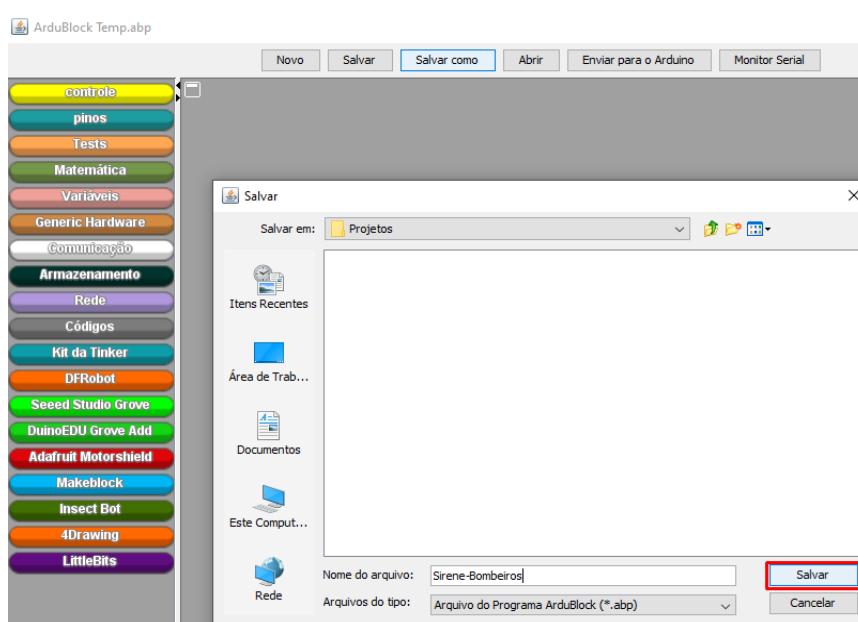


A P O S T I L A K I T  
**ARDUINO KIDS**

Para salvar nosso projeto em blocos, para abrir de novo no Ardublock, e não em códigos escritos, clique em Salvar como, e salve ele no mesmo lugar escolhido anteriormente. Coloque um nome sugestivo, como “sirene-bombeiro”, para que você lembre no futuro com facilidade.



E clique em “Salvar”.



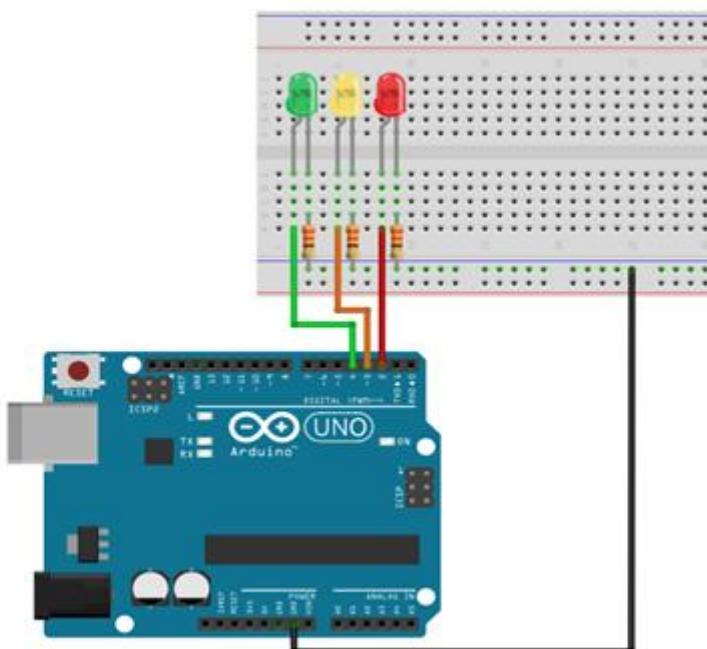
## Exemplo 2 - Semáforo de Carros

Para fazer um semáforo, iremos precisar de algo que emita luz. No kit, temos um conjunto de LEDs que são como mini lâmpadas que acendem quando recebem energia. O maior problema dos LEDs é que eles só podem receber uma quantidade pequena de energia, e se eles receberem mais do que deveria, eles queimam. Pensando nisso, iremos usar os resistores. Nessa montagem os resistores têm a função de limitar a energia que chegará no LED, para não o deixar queimar.

Para fazer esse circuito, iremos precisar de:

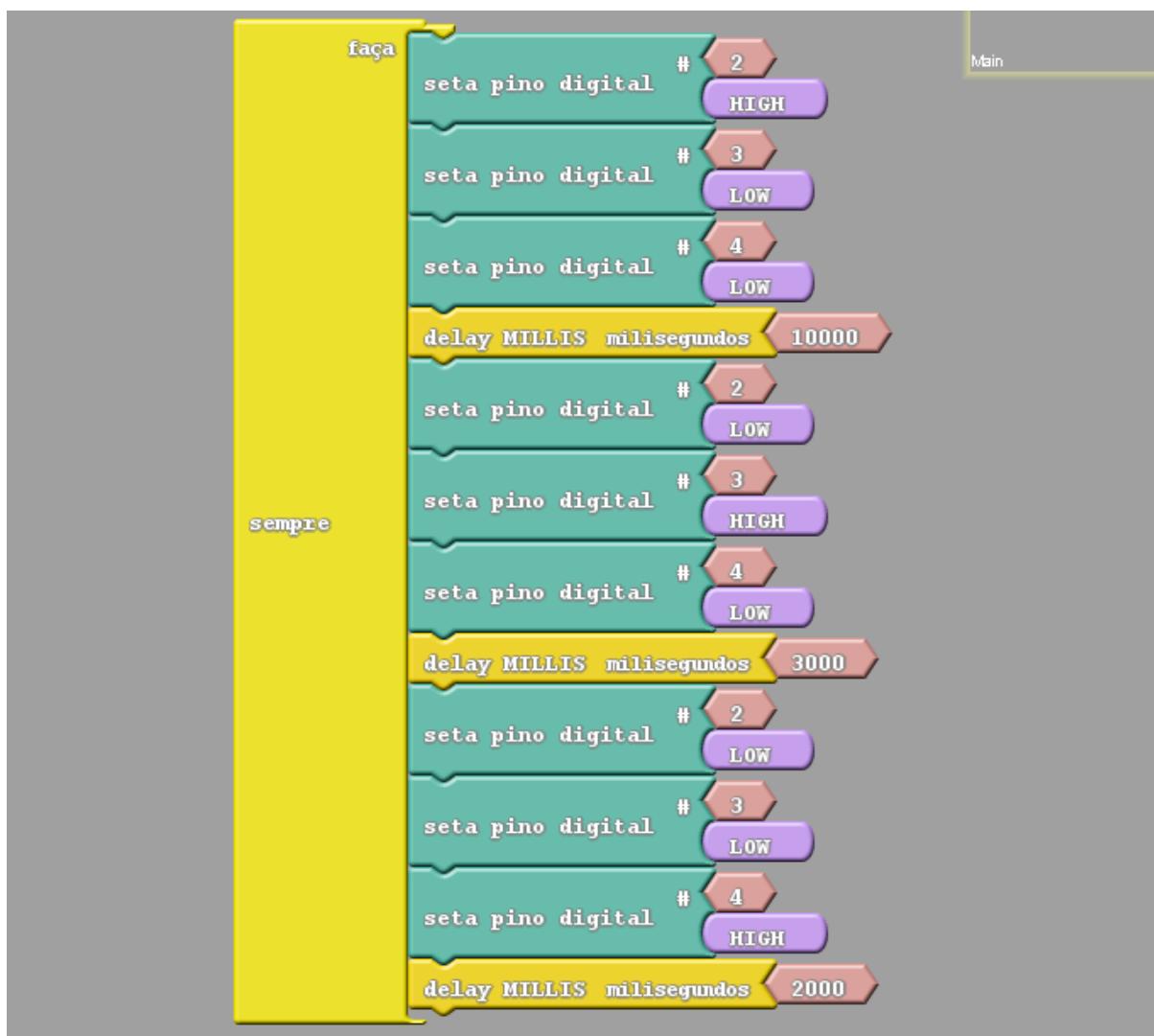
- 1 LED vermelho
- 1 LED amarelo
- 1 LED verde
- 3 resistores 330Ω (Faixas laranja, laranja, marrom)
- Jumpers
- Protoboard
- Arduino e cabo de conectar no computador

A montagem final do circuito é a seguinte:



A P O S T I L A K I T  
**ARDUINO KIDS**

E os blocos montados no ardublock são estes:



**Observação:** Os LEDs têm **polaridade**, o que significa que eles só ligam de em uma posição certa, e da outra eles não funcionam. Se caso um LED não ligou, inverta seus terminais na montagem, e ele deverá funcionar. Se mesmo assim não funcionar, revise a montagem.

Terminando estes passos, você terá um mini-semáforo. Para ajustar ele para outra contagem de tempo entre cada sinal, só mudar o valor que acompanha o bloco “delay MILLIS”!.



### **Exemplo 3 - Mini Piano com Sons**

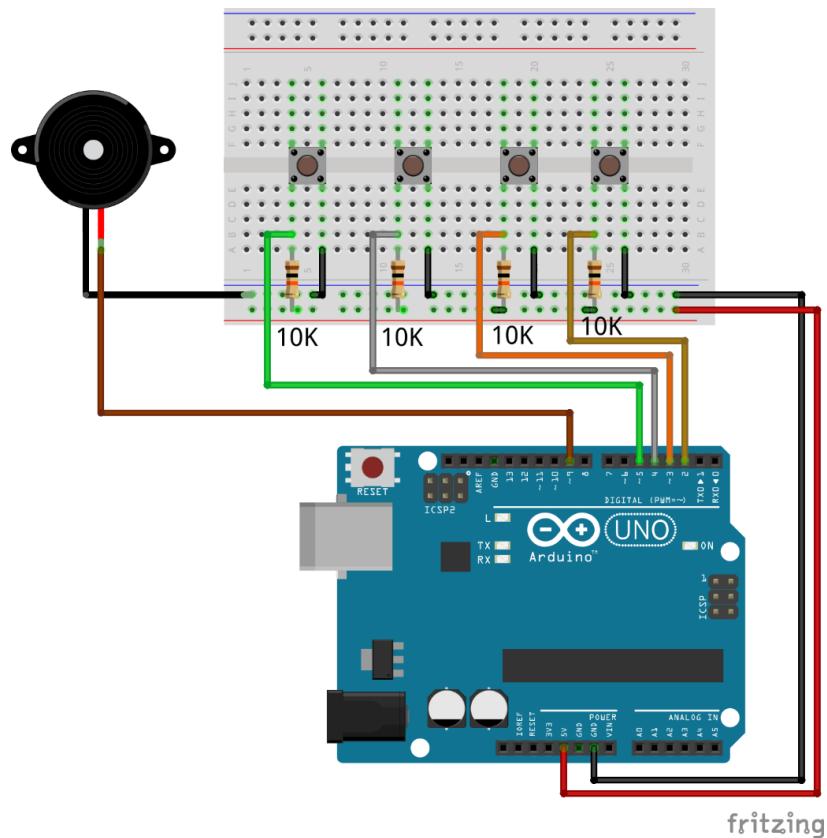
Este projeto usará o buzzer, chave táctil para acionar o buzzer em notas musicais específicas, para cada botão apertado. A chave táctil funciona da seguinte forma: Quando apertamos este botão, ele deixa a energia fluir, assim, o Arduino consegue detectar que o botão foi apertado. Aproveitando esse funcionamento do botão, iremos criar algo que tem o mesmo funcionamento do teclado, com a diferença de que não temos uma caixa de som potente e com definição.

Para montar o projeto precisaremos de:

- 4 chaves tactil
- 4 resistores de 10K (faixas marrom, preta e laranja)
- 1 Buzzer ativo
- Jumpers
- Protoboard
- Arduino e cabo de conectar no computador

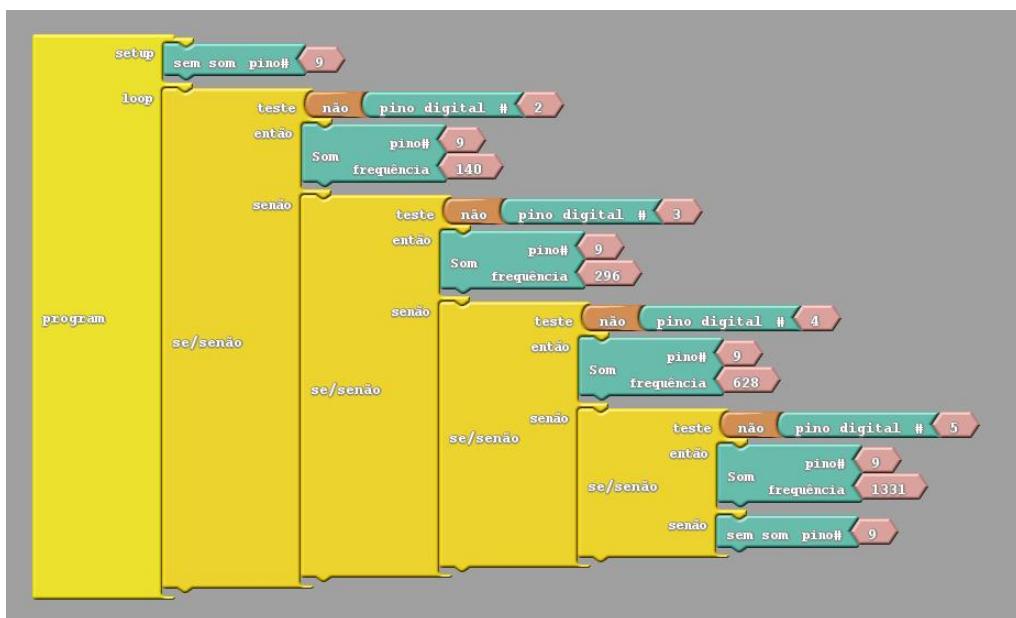
A montagem deve ser feita da seguinte forma:

APOSTILA KIT  
ARDUINO KIDS



Verifique que um botão sempre deve estar acompanhado de um resistor de 10k (Que possui as cores marrom, preto e laranja), como na montagem. Confira se todos os cabos estão conectados como na imagem.

Você deve montar o seguinte código no Ardublock:



A P O S T I L A K I T  
**ARDUINO KIDS**

Perceba que um novo bloco surgiu: O Se/Senão. Este bloco é extremamente útil e muito usado na programação das mais diversas coisas. Ele compara, neste caso, se a porta selecionada está ativo ou não, se caso estiver ativo, ele fará a primeira coisa, que está escrito em “então”, se não estiver ativo, ele fará o que está escrito em então.

No nosso código, o Arduino irá ver se o botão 2 foi apertado, se não foi, ele irá verificar o pino 3, e depois o 4 , e depois o 5; se nenhum botão foi apertado, ele irá fazer o último senão, que é não ligar nenhum som.

E com estes passos cumpridos, só enviar o código como aprendido anteriormente, verificando sempre se em “Ferramentas>Portas” se está selecionada a porta que tem “Arduino/Genuino Uno”. Com isso, aprendemos mais sobre funcionamento de teclados e pianos, e como usar botões.

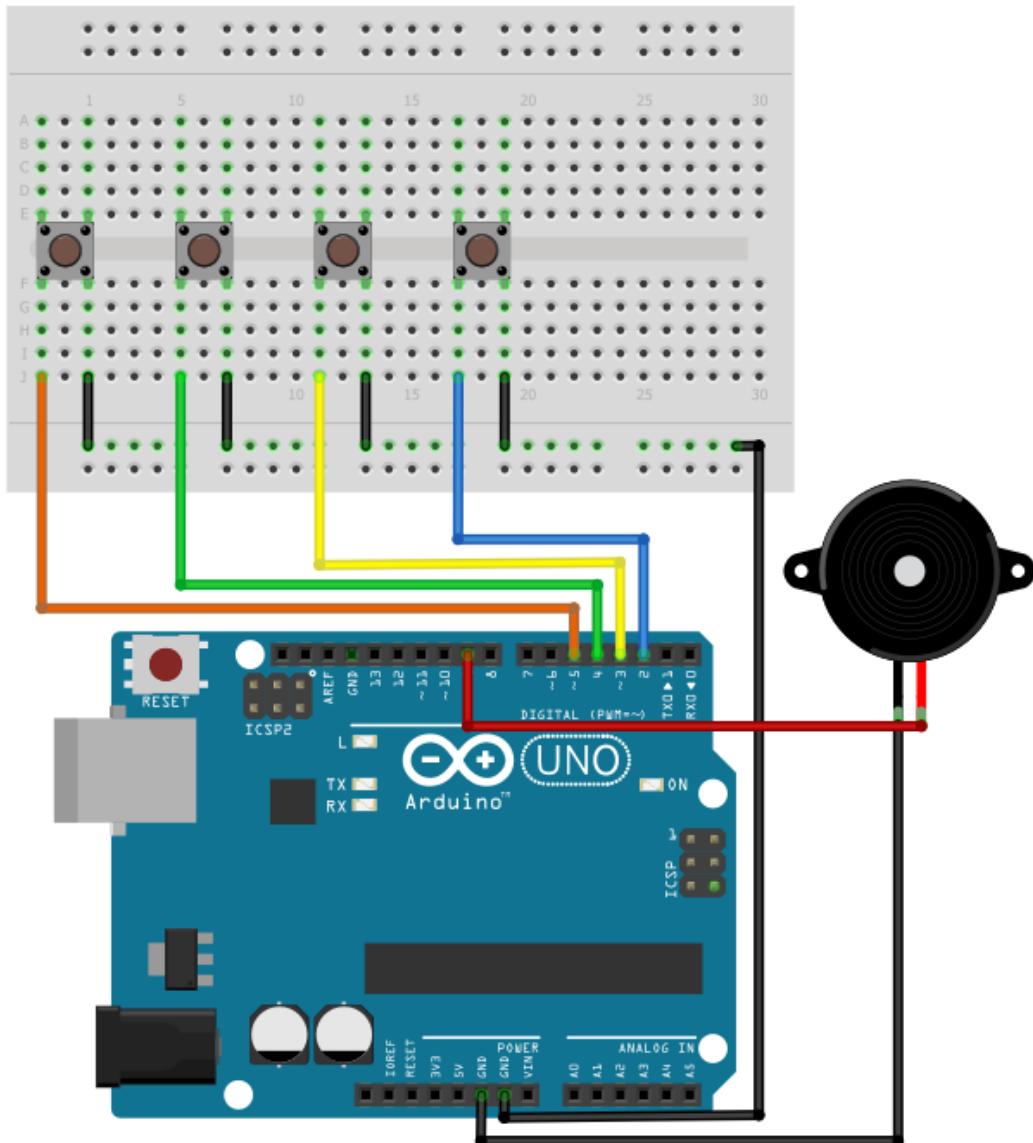
### **Montagem alternativa**

Aqui aprenderemos como fazer o mesmo projeto, mas sem utilizar os resistores. Essa montagem é muito interessante quando se quer fazer uma montagem mais limpa e reduzir os problemas de mal contato.

Para isso, faça a seguinte montagem abaixo:

# A P O S T I L A K I T

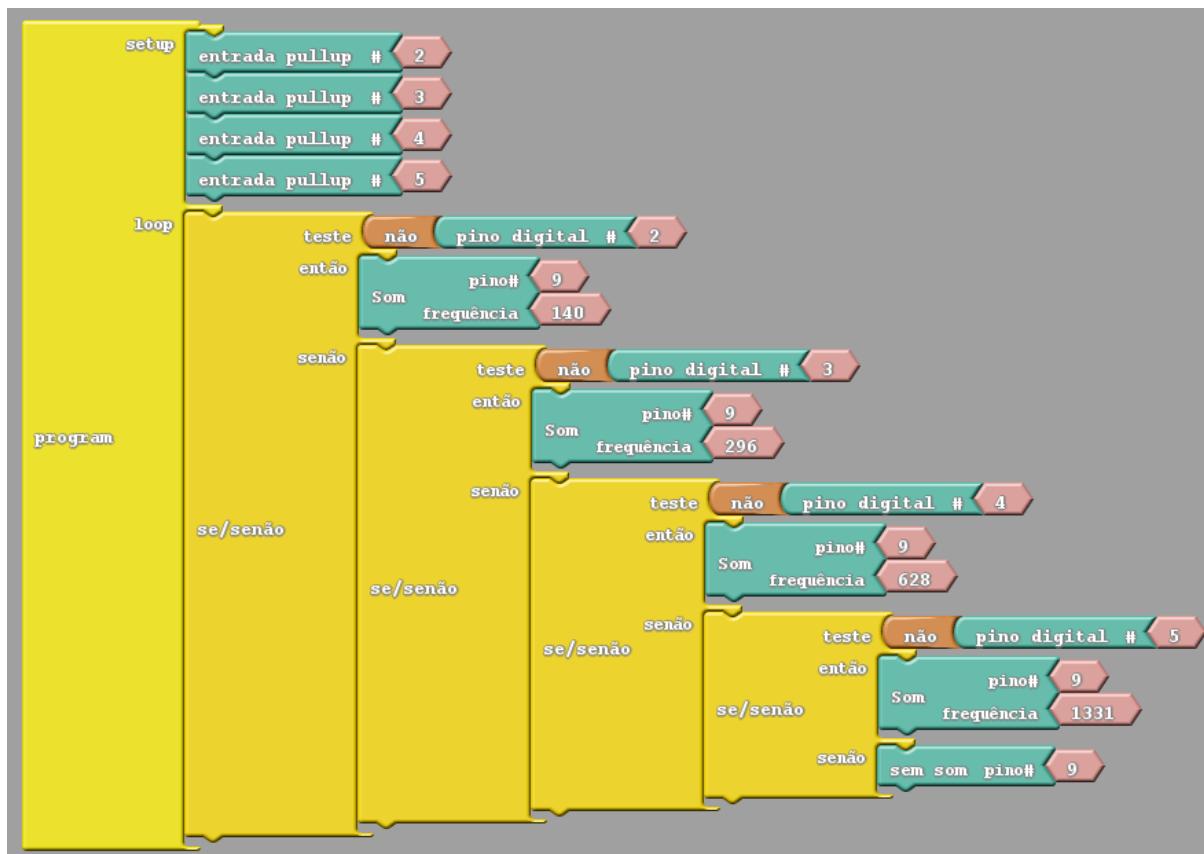
# **ARDUINO KIDS**



O código a ser carregado é diferente, porque vamos usar um recurso que habilita os resistores internamente na placa Arduino Uno. Esse recurso se chama entrada pullup, que estamos usando no início do código.

Talvez você deva estar se perguntando: Para quê serve o resistor nessa montagem, seja na protoboard ou internamente? A pergunta é interessante, porque parece que não serve para nada, mas os resistores são extremamente importante para que a placa entenda se o botão foi pressionado ou não. Então, toda vez que você usar um botão, você deve ativar os resistores internamente ou ligar diretamente na protoboard.

Monte o seguinte código no Ardublock para a montagem alternativa:

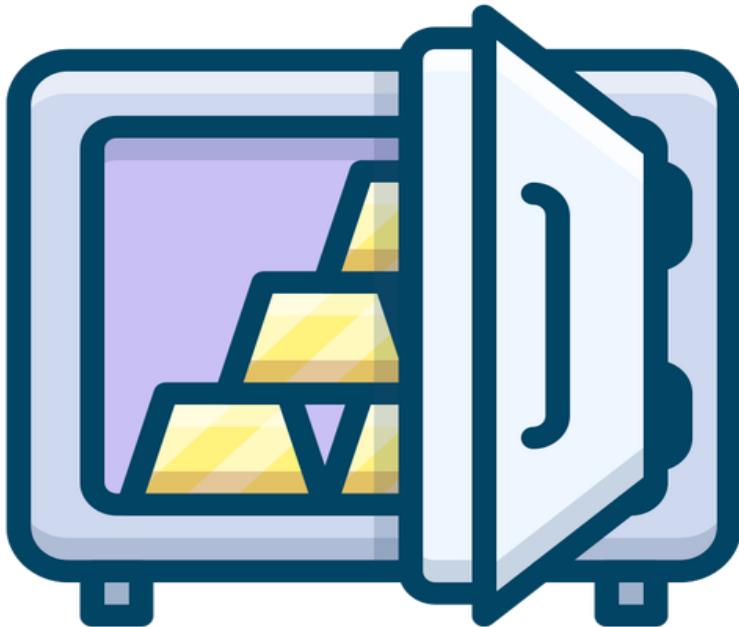


E com isso você aprendeu dois jeitos de tocar suas músicas com seu mini piano.

## Exemplo 4 - Sensor de Proximidade para Cegos

O sensor ultrassônico serve para medir distância. Ele funciona da mesma forma que um sensor de ré dos automóveis, e da mesma maneira em que morcegos e golfinhos fazem para saber a distância de objetos. O sensor ultrassônico tem o seguinte princípio: Ele envia um som curto, e conta quanto tempo demora para este som voltar. Pensando nisso, iremos usar um buzzer e o sensor de distância e fazer um projeto para auxiliar pessoas cegas, onde, se um objeto estiver próximo de colidir com a pessoa, será feito avisos sonoros diferentes, dependendo da distância.

Para este projeto, além dos clássicos acionamentos das portas, teremos que aprender um pouco sobre o que são variáveis. Para compreender o que são variáveis, podemos pensar em um cofre que guarda nosso dinheiro.



Podemos guardar dentro do cofre desde nenhum dinheiro, até quantidades muito grandes, como milhões. O interessante de um cofre, é que o valor que está dentro muda de acordo com a nossa vontade, podemos tirar dinheiro, colocar dinheiro, e assim, o mesmo cofre, pode armazenar vários valores diferentes. Com as variáveis em um programa de computador também são assim, elas podem armazenar qualquer valor, e podemos alterar esse valor quando quisermos.

Existem vários tipos diferentes de dados, e para isso, precisarão de tipos de variáveis diferentes: assim como na vida real existem moedas, ouro, e dinheiro em cédula, e para cada tipo de dinheiro desses, precisamos de um cofre diferente. Por exemplo, existe o tipo de dado “inteiro”, esse tipo de número não tem vírgula, e então, para armazená-lo precisamos declarar a variável que irá salvar ele como “inteiro”. Existem outros dois tipos comuns de dados: que são os de números reais e os de caracteres. Os números reais são os números que podem ter vírgulas, e caracteres são qualquer dado que aparece na tela, sendo letras ou símbolos.

No nosso programa, iremos criar uma variável inteira de nome “distância”, que irá armazenar para a gente o valor lido pelo sensor de distância, para não precisar ficar lendo toda hora que fazer essa operação de chamar o sensor para ele enviar o som, contar o tempo, e converter em distância.

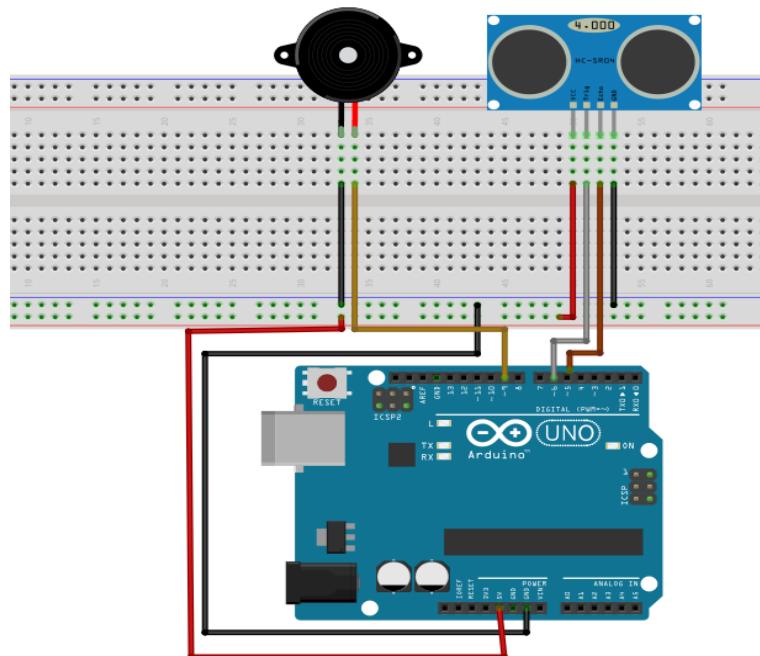
A POSTILA KIT  
**ARDUINO KIDS**

O código do nosso programa, como no anterior que vimos, também usará o “se/senão”, porque o comportamento do som vai depender da distância medida.

Para este projeto iremos precisar de:

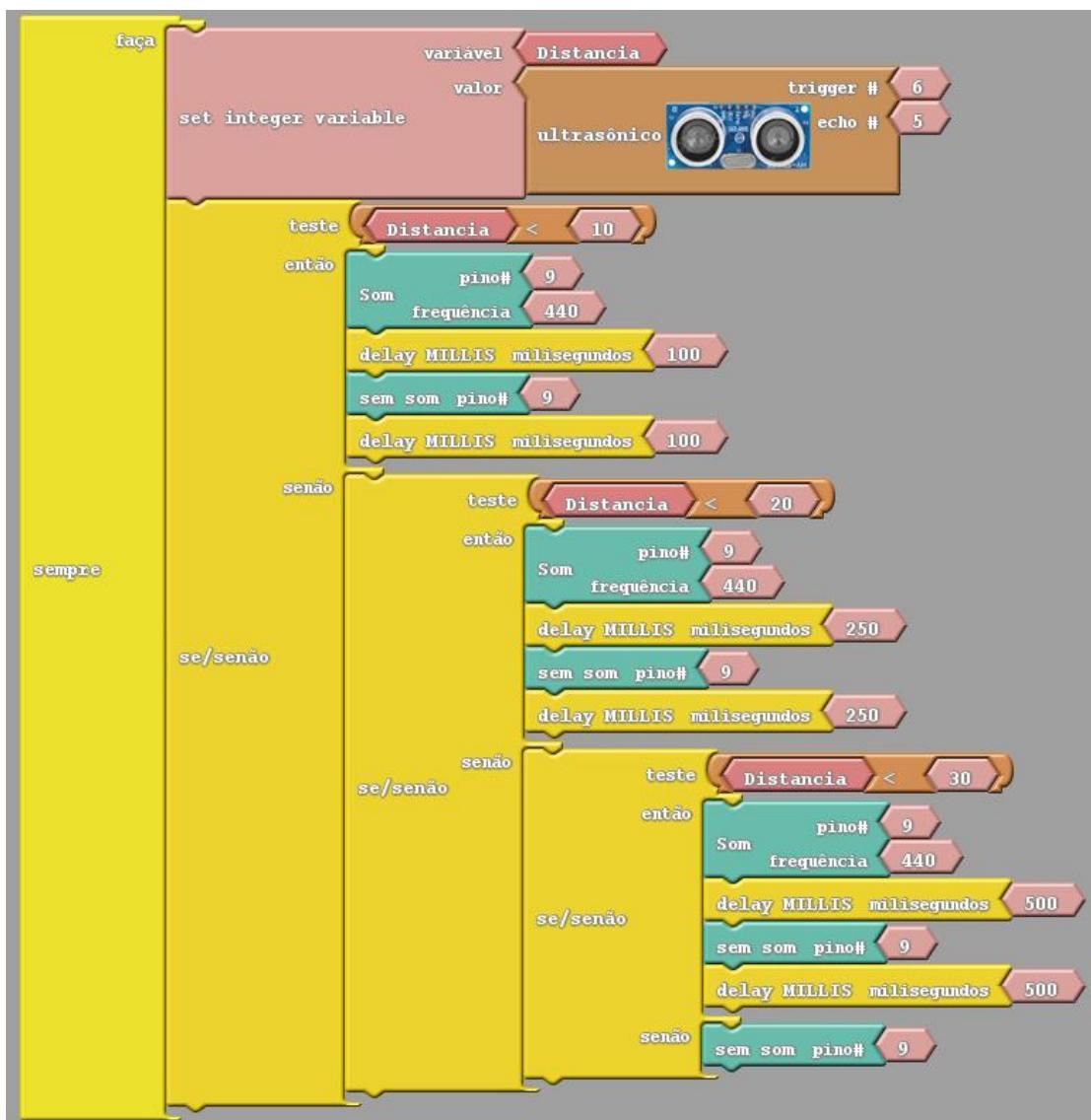
- Buzzer Ativo
- Sensor ultrassônico
- Jumpers
- Protoboard
- Arduino e cabo de conectar no computador

A montagem para o circuito é esta:



E este é o código utilizado:

A P O S T I L A K I T  
**ARDUINO KIDS**



Note que esses vários se/senão dentro do outro é para garantir que o programa funcione corretamente. Olhe na primeira linha do código e tente entender como se faz para declarar variáveis no Ardublock: Primeiro colocamos o nome na variável, e o valor embaixo. Note que o valor que estamos colocando é a distância do sensor ultrassônico. Falando no bloco do sensor ultrassônico, é bem simples, note que ele pede dois valores, o de trigger e echo. Note que no seu sensor ultrassônico, na placa, tem marcado quais são os pinos de echo e trigger, e com isso, sempre declare o pino certo para o programa. (No nosso caso eram os pinos 6 e 5, mas poderiam ser outros).

A P O S T I L A K I T  
**ARDUINO KIDS**



Vale lembrar que se você não estiver achando um bloco, quase sempre basta olhar a cor e procurar pela cor dele na guia de organização dos blocos da esquerda.

### **Exemplo 5 - Cancela Inteligente Controlada por Smartphone**

O servo motor diferencia de um motor comum porque podemos escolher o quanto ele vai girar, e em qual posição ele vai ficar; porque o motor comum, quando energizado, só gira, e não podemos escolher qual posição final dele. Um servo motor é ideal para projetos onde precisamos dizer exatamente quanto o motor deve girar. O exemplo ideal é uma cancela: Ela precisa ficar na horizontal, para evitar a passagem de um carro; mas depois da identificação ou pagamento (no caso de um pedágio) a cancela deve subir e ficar na vertical e aguardar um tempo e voltar para a posição horizontal.

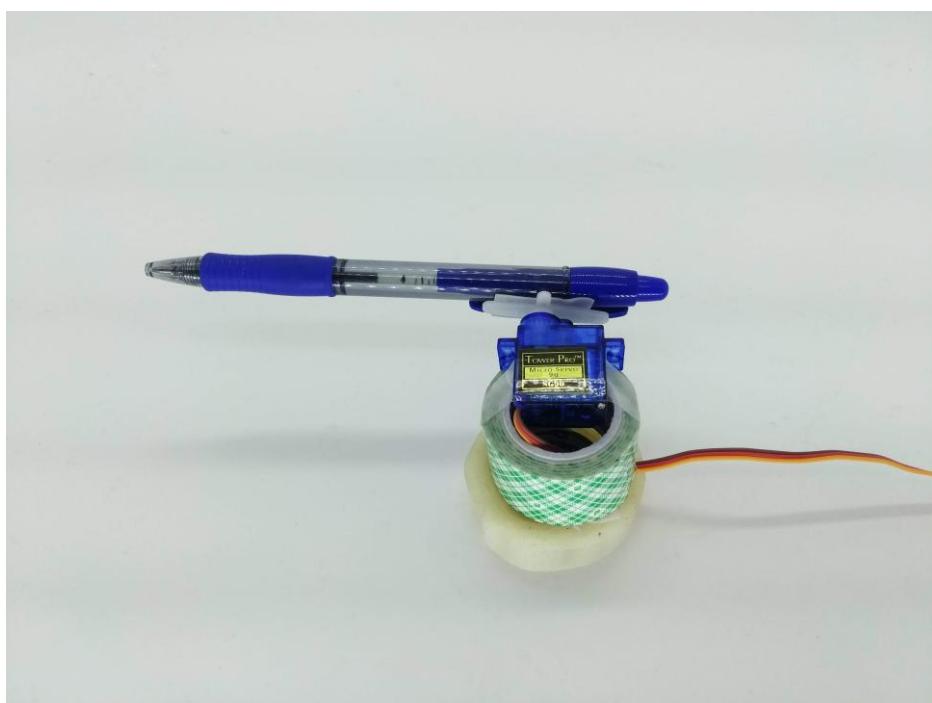
O bluetooth é uma tecnologia que permite a comunicação sem fios entre componentes, podendo conectar o nosso celular ao Arduino, onde daremos comandos pelo celular a uma placa que não está ligada a nada.

Neste projeto iremos implementar uma cancela inteligente que receberá ordem de subir e descer a distância. Só iremos precisar de:

- Módulo Bluetooth HC-06
- Resistor 22K (Faixa Vermelha, vermelha e laranja)
- Resistor 10K (Faixa Marrom, preta e laranja)
- Microsservo 9g
- Materiais de escola e escritório: Caneta, 2 rolos fitas adesivas pequenas e o que achar melhor para montar uma base, como veremos.
- Jumpers
- Protoboard
- Arduino e cabo de conectar no computador

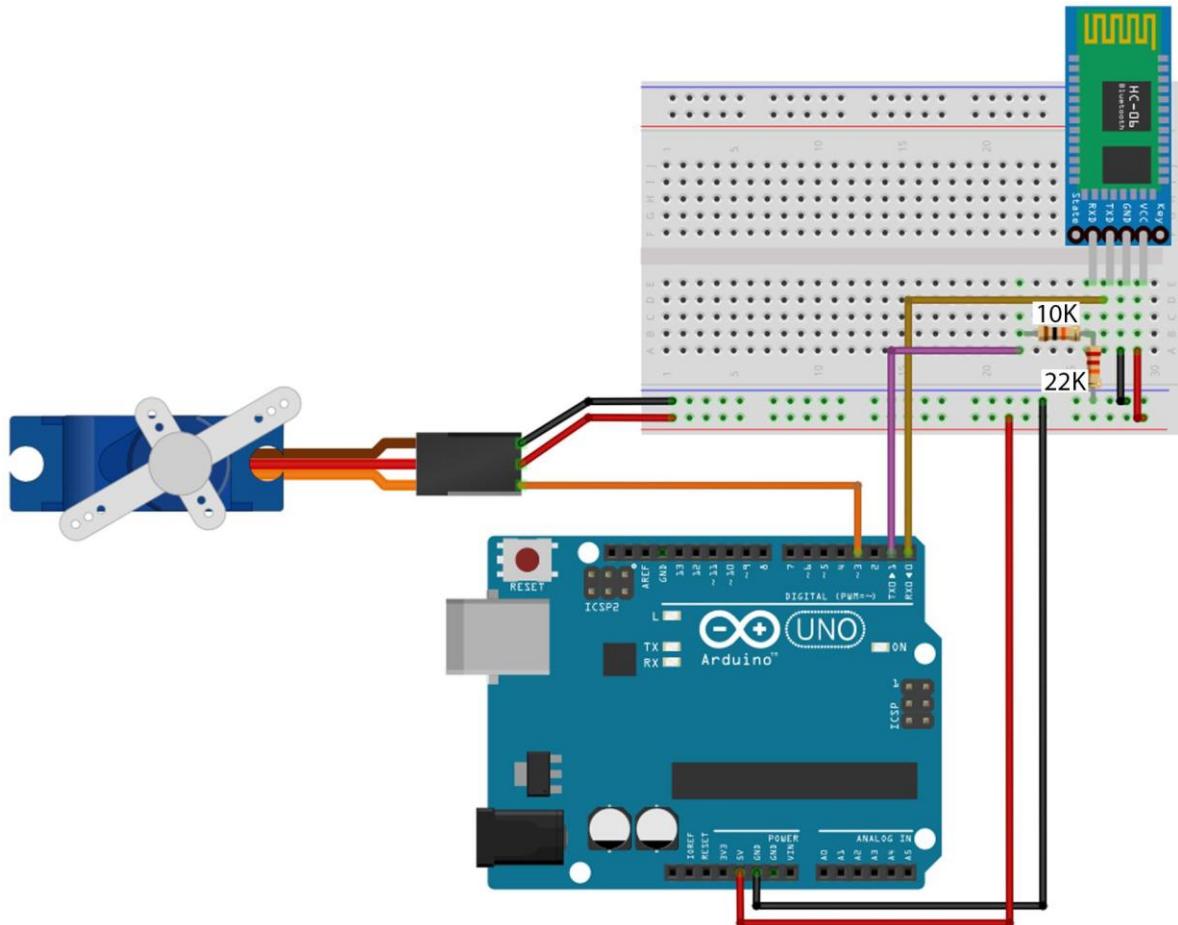
A P O S T I L A K I T  
**ARDUINO KIDS**

Primeiro, monte uma estrutura parecida com a imagem abaixo, usando o microservo 9g. Observe dentro da embalagem do servo que vem alguns parafusos e uns acessórios brancos que encaixam no servo. Use algum destes acessórios para colar o servo na caneta, ou lápis, e depois cole a parte azul do servo numa base que você criou, para ficar parecido com uma cancela. Com tudo feito, a sua cancela deve estar parecida com a da imagem abaixo:



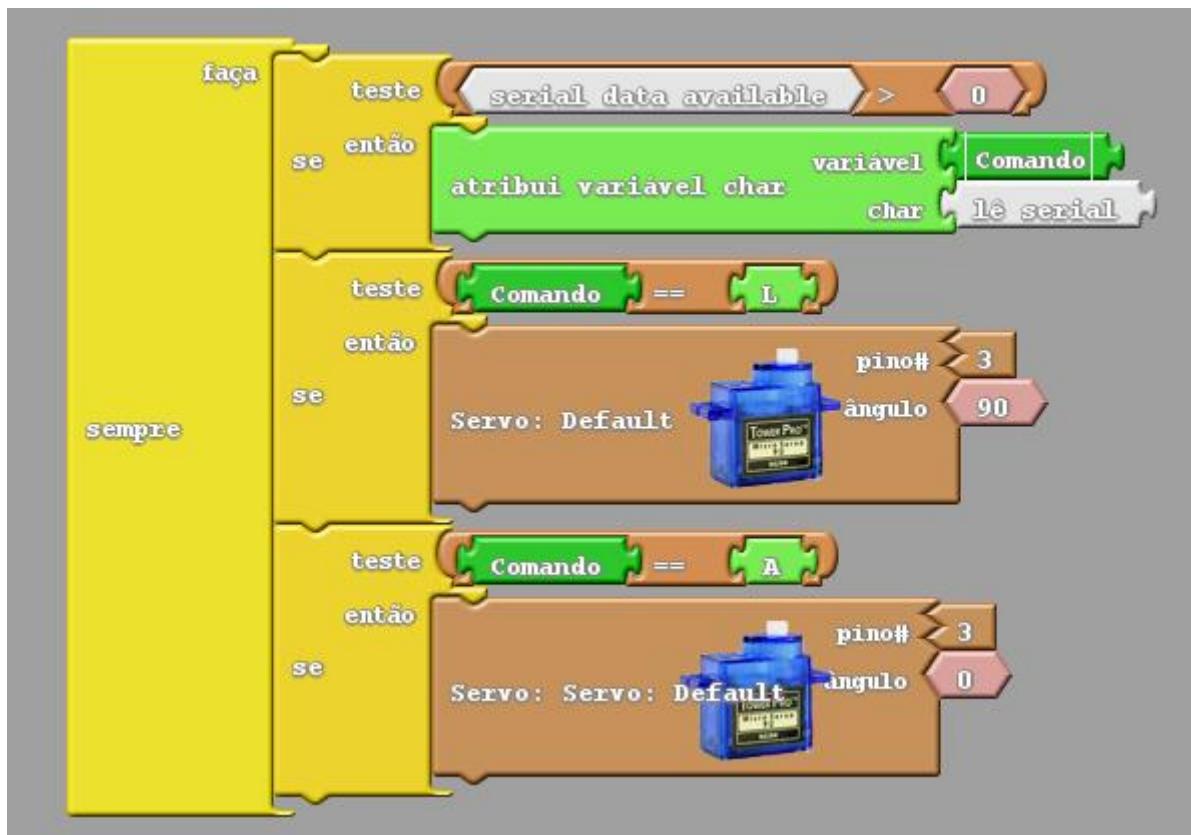
A P O S T I L A K I T  
**ARDUINO KIDS**

E agora devemos fazer a montagem do circuito para fazer funcionar.



Nessa montagem estamos usando resistores, um de 10K e outro de 22K. Essa montagem de resistores serve para converter os 5V que vem do pino TX da placa Arduino em 3.3V, que é o necessário para o módulo Bluetooth. Faça a montagem com muita calma e prestando bastante atenção. Outro ponto importante é ter cuidado para não inverter os pinos de alimentação 5V e GND nos componentes, porque pode danificar eles.

O código utilizado é este:

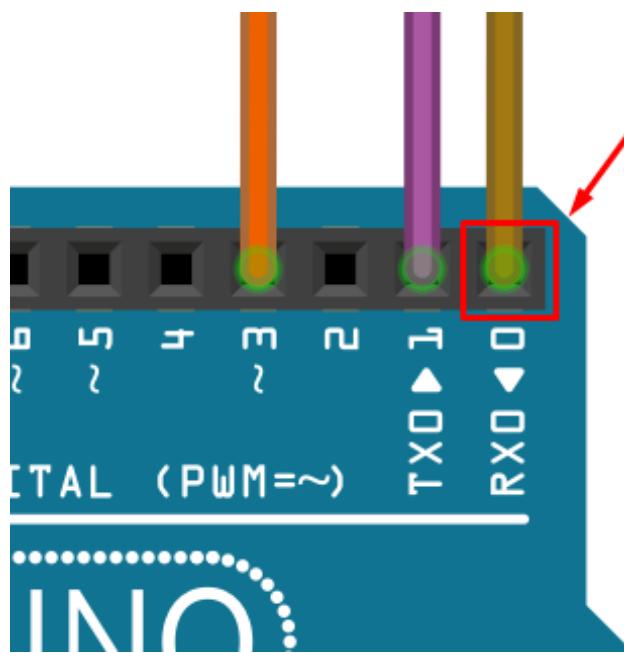


Neste código estamos usando algumas coisas novas, e outras que não seguem as regrinhas das cores. A primeira coisa novidade é, estamos usando os blocos de comunicação: Esses blocos vão nos permitir fazer a conexão bluetooth com o celular. No primeiro “Se”, estamos verificando se o celular enviou algum comando, e se enviou, a gente vai salvar esse comando, que é com o bloco “atribui variável char”. **Esse bloco, que é verde, foge a regra das cores, porque ele se encontra, na verdade, na parte de variáveis.** Os outros blocos “Se” servem para ver se o comando é L (Levantar) ou A (Abaixar), e fazem os respectivos comandos.

#### OBSERVAÇÃO:

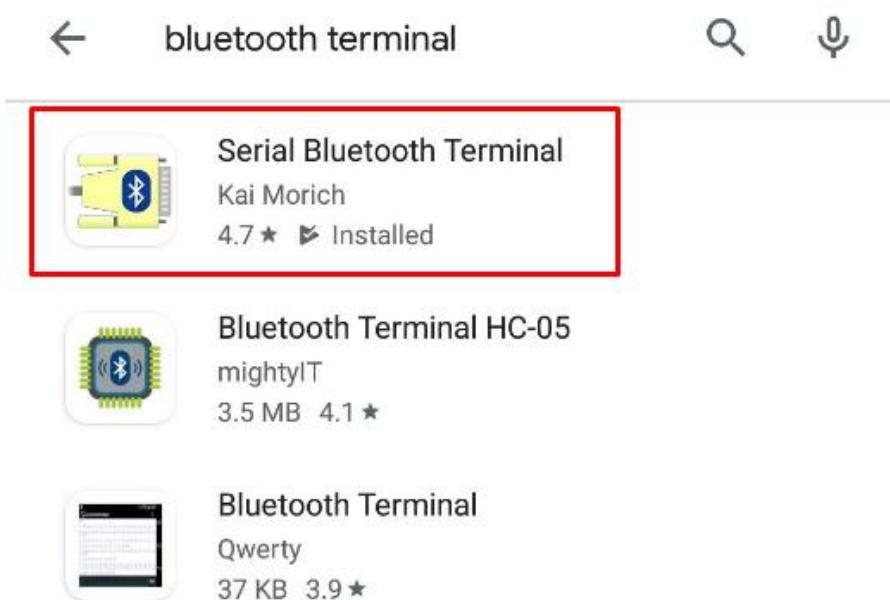
Ao enviar o programa para o Arduino, desconecte o jumper que está na porta 0 (RX). Quando o programa for enviado e estiver confirmado, você pode recolocar o jumper. Só lembre que sempre que for enviar o programa, precisa retirar enquanto está carregando.

A P O S T I L A K I T  
**ARDUINO KIDS**



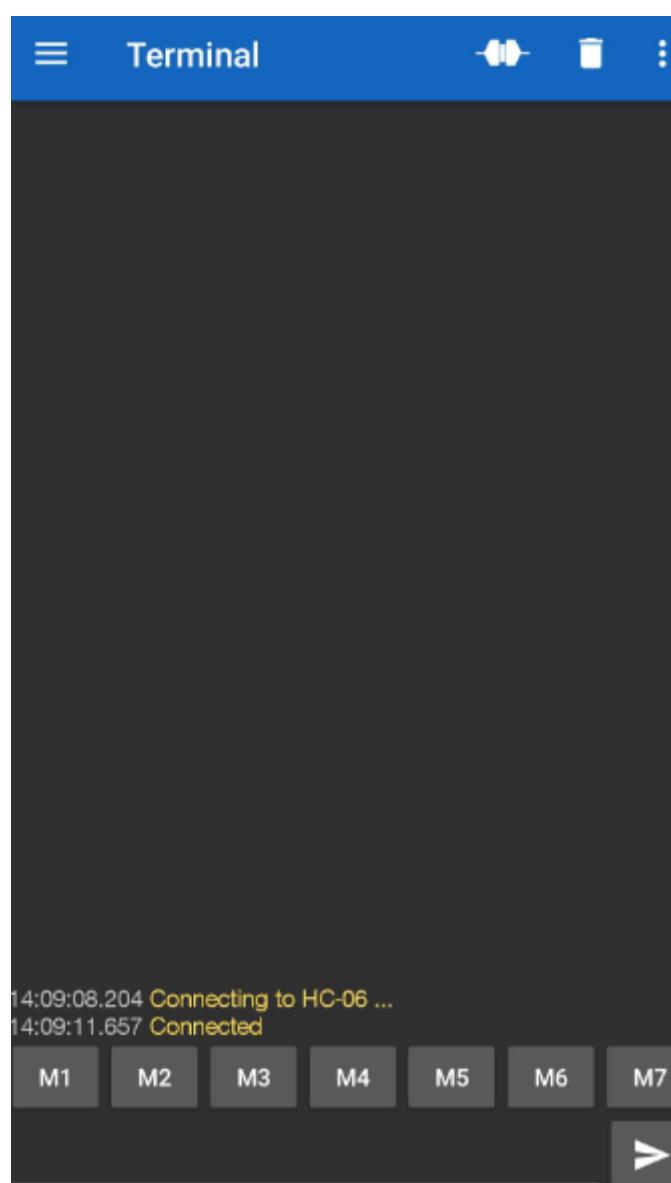
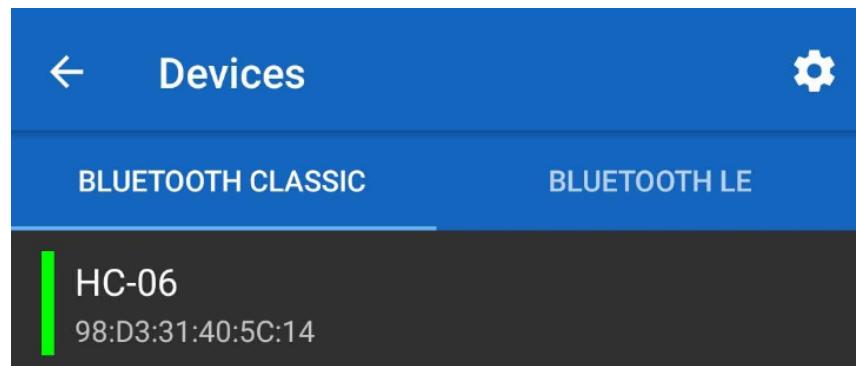
Para fazer a comunicação com o celular, iremos precisar de um celular Android, ou Windows, porque infelizmente Iphone não oferece suporte ao bluetooth 3.0, que é o que usamos.

No caso do Android, que é o mais comum, abra as configurações do seu celular, vá no bluetooth, e pareie o dispositivo HC-06. Se pedir senha, coloque “1234”. Depois vá na loja de aplicativos (Google Play) e procure por “Bluetooth Terminal”, e baixe o que está marcado na imagem.



A P O S T I L A K I T  
**ARDUINO KIDS**

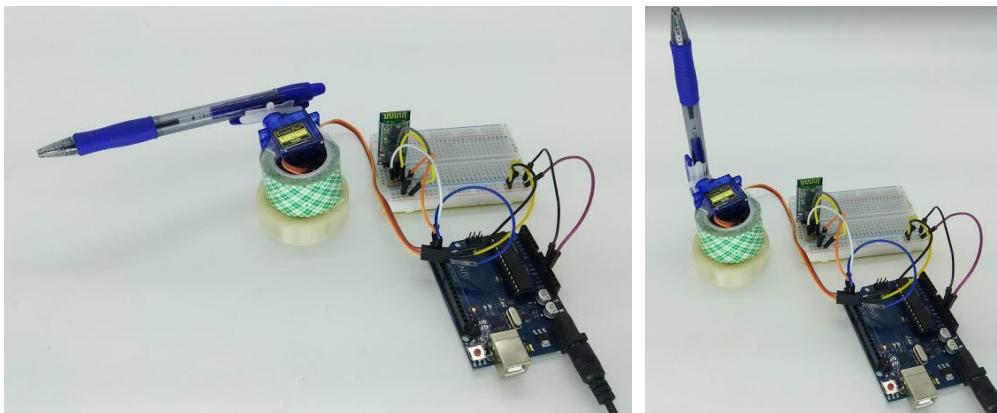
Assim que finalizar a instalação, abra o aplicativo. No aplicativo, abra as opções adicionais no lado esquerdo, e vá em Devices (Ou dispositivos), e selecione HC-06.



A P O S T I L A K I T  
**ARDUINO KIDS**

E com esses passos concluídos, podemos enviar os comandos para a nossa cancela subir e descer com os comandos **L** (Levantar) e **A** (Abaixar), que podemos digitar.

O resultado esperado é este:



Com isso, aprendemos como enviar e receber comando Bluetooth, e com isso podemos integrar o Bluetooth a qualquer outro projeto aprendido anteriormente e fazer coisas como ligar LEDs, acionar o Buzzer, e diversas outras coisas.

## **Exemplo 6 - Construindo um Carrinho Autônomo**

A principal vantagem deste carrinho é que iremos poder integrar a maioria dos conceitos aprendidos aqui, podendo controlar e personalizar o carrinho do melhor jeito, aproveitando os exemplos anteriores.

Esta montagem do carrinho também está disponível no nosso em <https://blog.blog.eletrogate.com>, onde existem diversas outras montagens e experimentos.

Vale a pena conferir.

Para o nosso projeto final, iremos precisar dos seguintes componentes:

- Módulo Ponte H Dupla L298N
- Chassi 2WD em acrílico

A P O S T I L A K I T  
**ARDUINO KIDS**

- 2 Motores DC 3V – 6V com caixa de redução
- 2 rodas 68mm de borracha
- Roda boba (universal)
- Suporte para 4 pilhas AA
- Jogo de parafusos e acessórios
- Jumpers macho-fêmea
- Jumpers macho-macho
- Adaptador de bateria 9V
- Chave Gangorra On/Off
- 4 Pilhas Alcalinas AA
- Bateria Alcalina 9V
- Protoboard
- Arduino e cabo de conectar no computador

Iremos começar pegando as peças primárias que compõem o carrinho, como está mostrado abaixo, e começaremos a soldar algumas coisas e a montar o chassis (chassis significa “esqueleto” do carrinho).



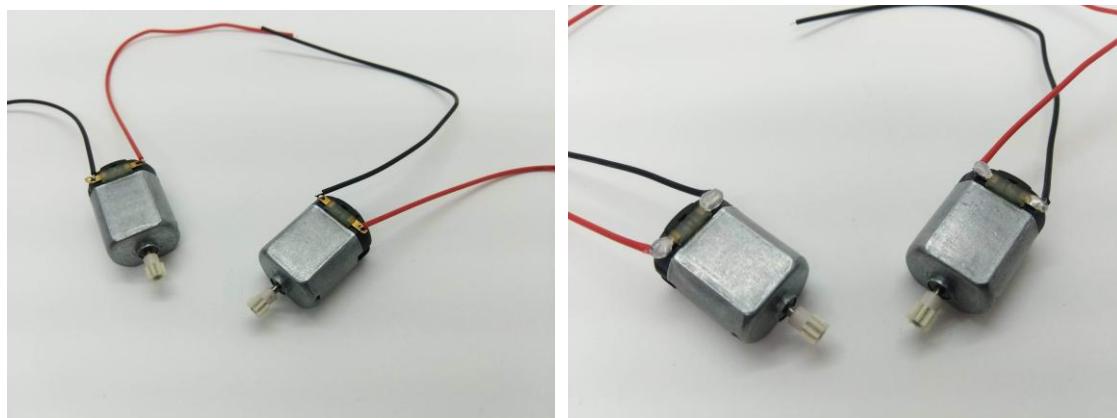
A P O S T I L A K I T  
**ARDUINO KIDS**

Agora iremos soldar os fios que vem nos locais corretos, e para isso, **peça a ajuda de um adulto ou leve em um profissional**, pois, pode ser perigoso se você for criança e fazer por conta e risco.

Para soldar os fios no motor, o primeiro será remover o motor da caixa de redução (a caixa de redução é essa parte amarela de plástico que vem nele). Primeiro, remova essa alça transparente, e então, puxe o motor.



Agora pegue 2 fios, solde ao motor e faça um acabamento colocando um pouco de cola quente, para evitar qualquer tipo de curto-circuito ou contato indesejado.



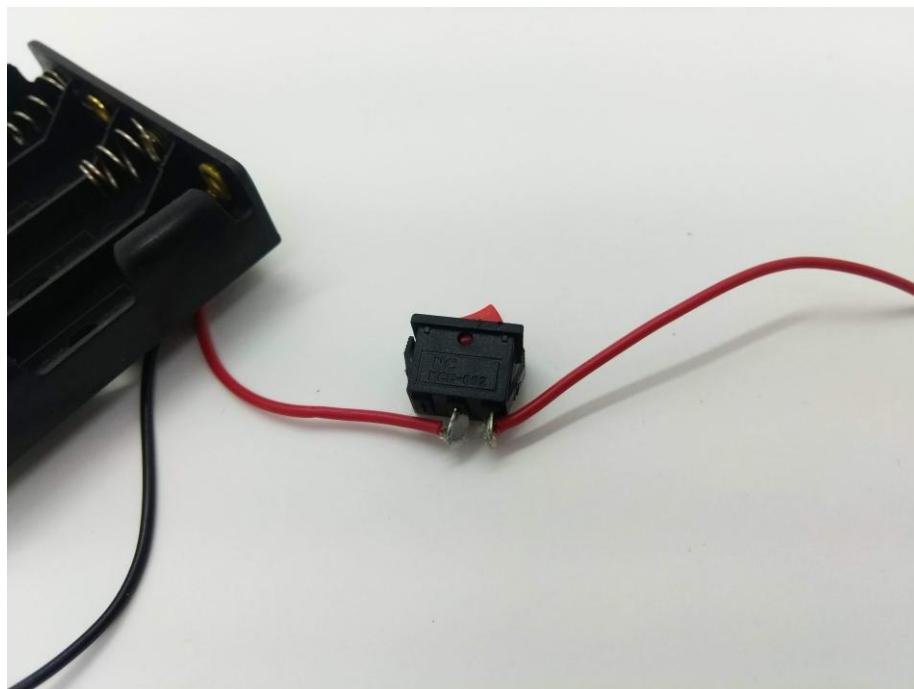
E faça isso, no caso, para os dois motores. E depois, volte com os motores para as suas respectivas caixas de redução (as caixas amarelas) e coloque a alça transparente, e assim, os motores já estão prontos para serem usados.

A P O S T I L A K I T  
**ARDUINO KIDS**



Aproveitando que tudo já está sendo soldado, iremos fazer agora a solda de uma chave gangorra (aquele chave on/off que vem no kit) no suporte de pilhas, para podermos desligar o carrinho e economizar energia.

O procedimento será bem simples, basta colocar a chave on/off no final do fio vermelho do suporte e soldar em um dos dois terminais, e pegar um outro fio disponível (pode ser um jumper macho/macho que veio junto ao kit) e soldar no outro fio. Se quiser também, uma possibilidade é cortar o fio vermelho ao meio, descascar só um pouco cada ponta, e fazer como nos passos anteriores; a desvantagem fica sendo a limitação onde poderemos colocar a chave para desligar na nossa montagem.



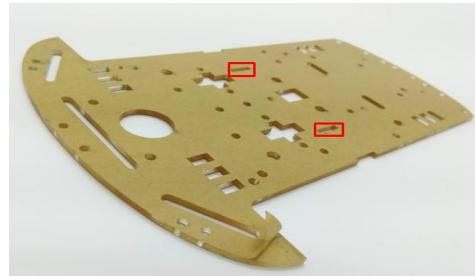
A P O S T I L A K I T  
**ARDUINO KIDS**

Com tudo soldado, podemos voltar a montagem da estrutura do carrinho. Observe que a peça maior do carrinho parece de madeira, mas só parece, porque na verdade ele e suas peças são transparentes, de acrílico. Se quiser deixar o seu carrinho mais estético, remova a película marrom de ambos os lados, só puxando, como na imagem abaixo.

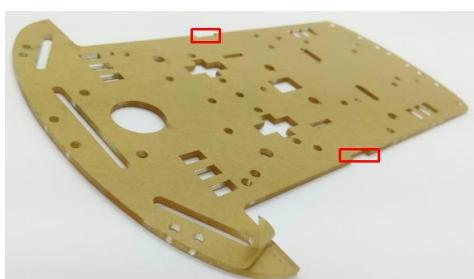


Agora fique com essas peças em mãos, pegue os motores e o conjunto de parafusos, e faça o seguinte:

1° - Encaixe duas peças menores dessas em formato de T, nas fendas internas.



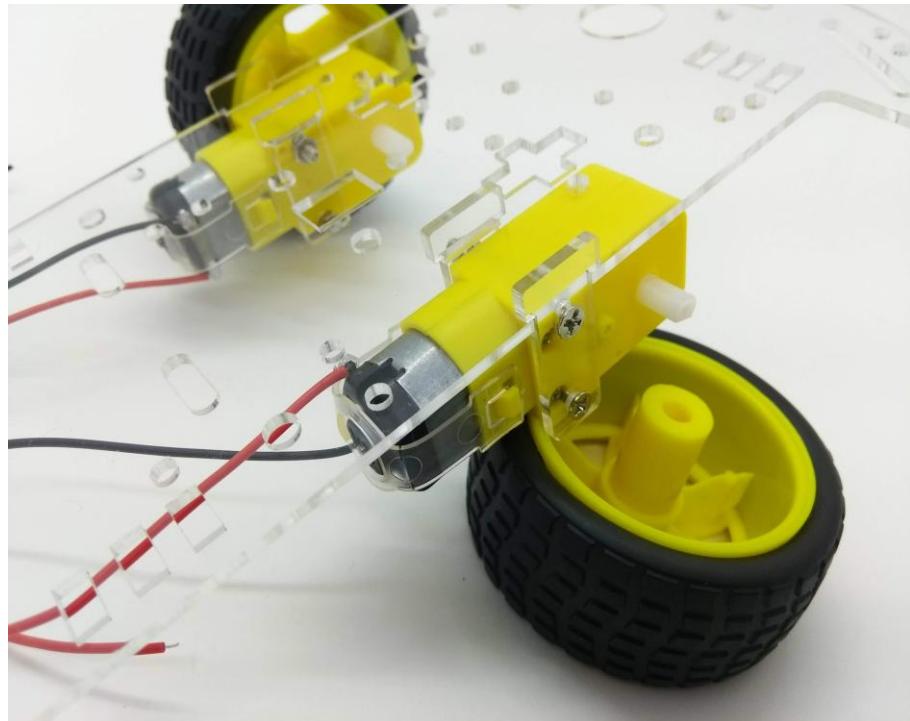
2° - Coloque o motor com seus furos laterais alinhados aos furos do suporte menor, que foi encaixado anteriormente



3° - Encaixe outros dois suportes pequenos em formato de T com os furos alinhados aos furos do motor também e parafuse usando porcas.

É interessante observar que devemos colocar o eixo do motor virado para a extremidade do chassi onde tem um círculo vazado no acrílico. Fazendo para os dois lados, e encaixando a roda no eixo do motor, o resultado obtido deverá ser este:

A POSTILA KIT  
**ARDUINO KIDS**

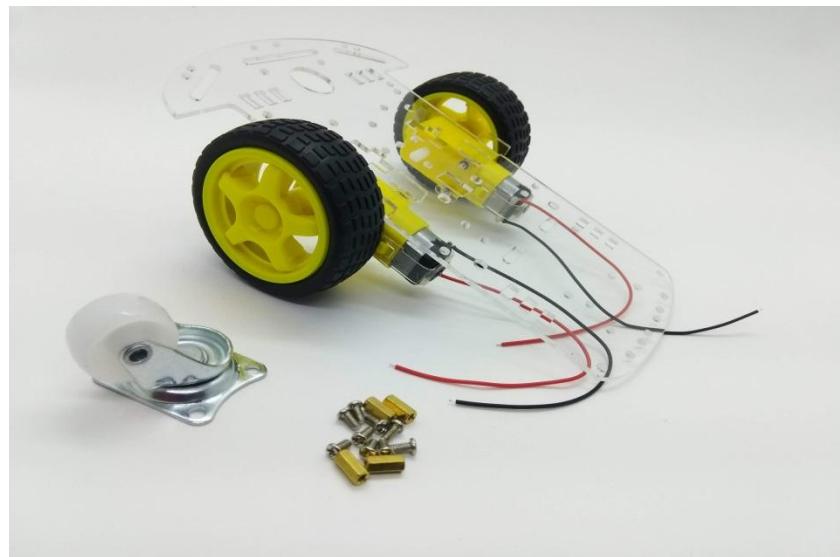


E encaixando a roda no eixo do motor, fica assim:

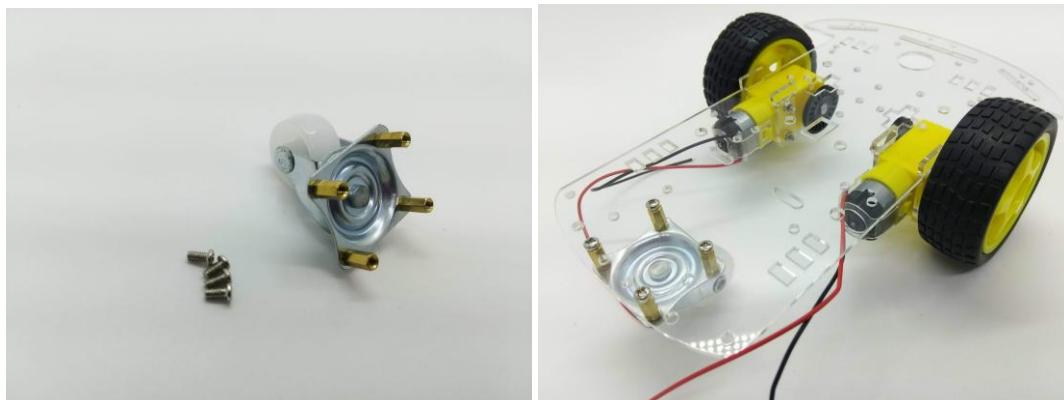


A P O S T I L A K I T  
**ARDUINO KIDS**

O próximo passo será a instalação da roda boba, e para isso precisaremos das seguintes peças:



Começaremos parafusando o extensor, que é essa peça dourada no kit de parafusos, à roda boba, e depois parafusando o extensor ao chassi do carrinho, da seguinte forma:

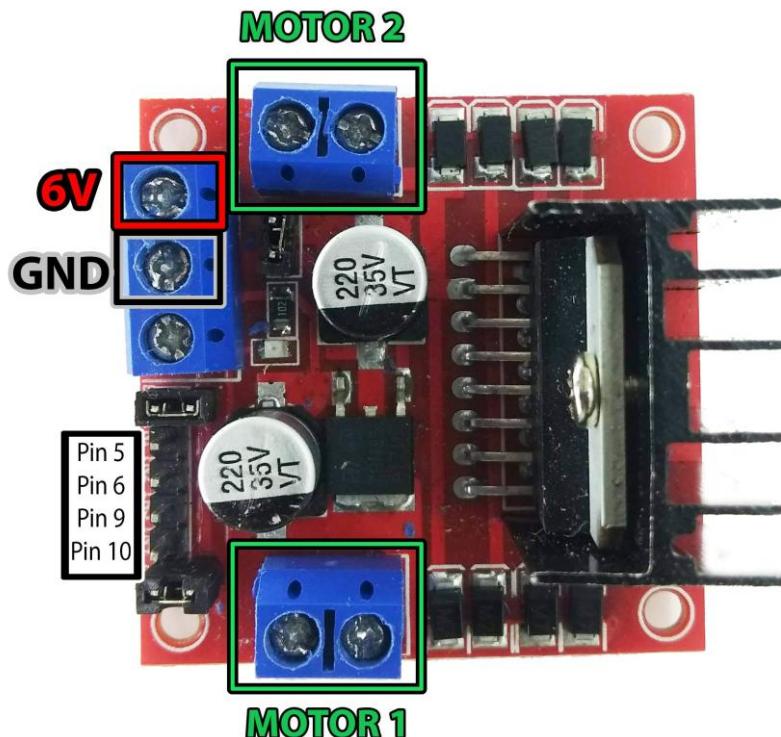


E assim, está concluído a montagem do esqueleto básico do carrinho, e agora iremos montar a parte para controlar ele.

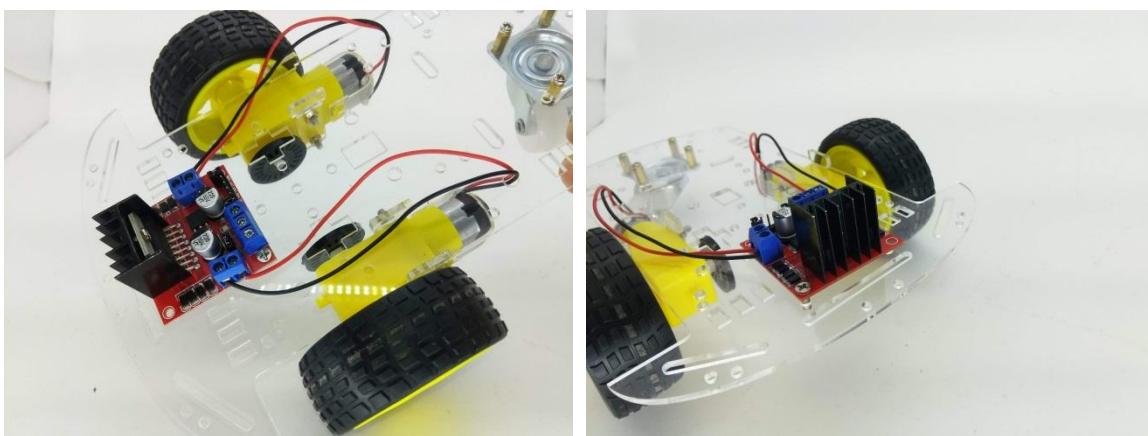
Para fazer o acionamento dos motores, precisaremos de algo que consiga fornecer energia o suficiente para isso, e que possa ser controlado. Para isso iremos usar um driver ponte H L298N. A principal vantagem de se usar um módulo destes é que ele permite que a gente possa controlar o motor nas duas direções.

A P O S T I L A K I T  
**ARDUINO KIDS**

O módulo ponte H usado é desta forma:

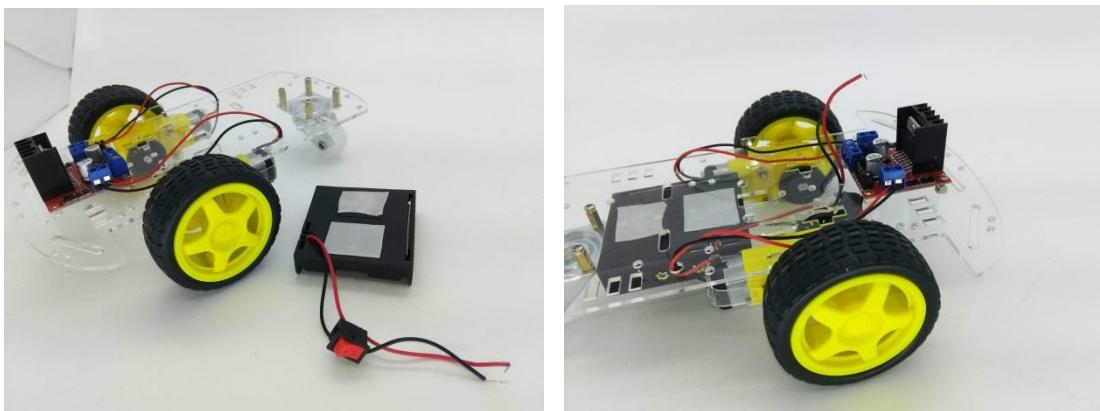


Onde iremos, inicialmente, fazer as conexões dos motores 1 e 2. Inicialmente não será necessário preocupar com qual fio do motor vai em qual lado para ser apertado; basta afrouxar o parafuso do borne, conectar o fio e apertar o parafuso selecionando um fio, e depois prenda o outro. A minha montagem ficou da seguinte forma:

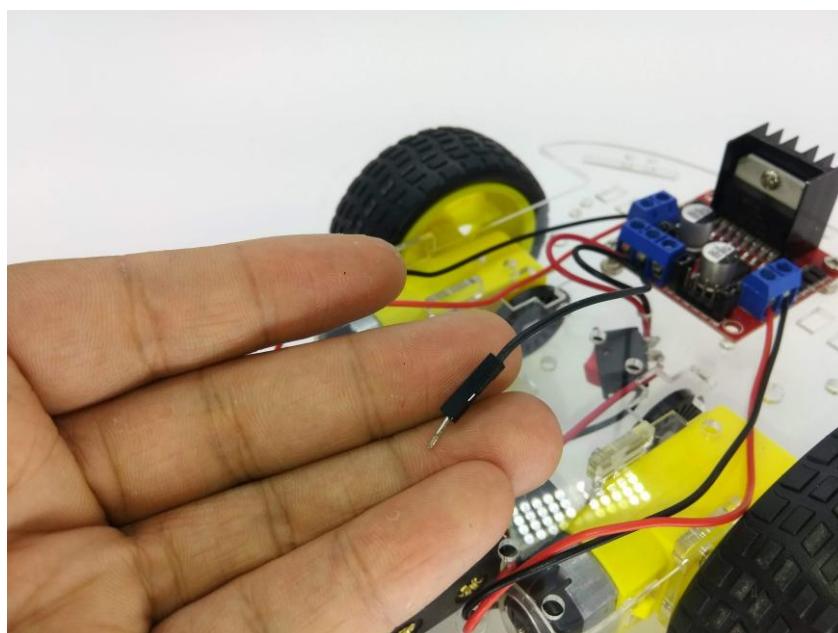


A P O S T I L A K I T  
**ARDUINO KIDS**

Perceba pela foto acima que há um parafuso no módulo ponte H. Esse parafuso está ligado com uma arruela ao chassi. Ela foi adicionada porque isso fará com que possamos dispensar o uso de colas e o deixar bem fixo, mas se quiser, pode usar fita dupla face. Com a ponte H instalada e os motores conectados, faremos a instalação do suporte de pilhas. Colocamos uma dupla face nas costas do suporte e então ele foi fixado na parte debaixo do chassi, da seguinte forma:



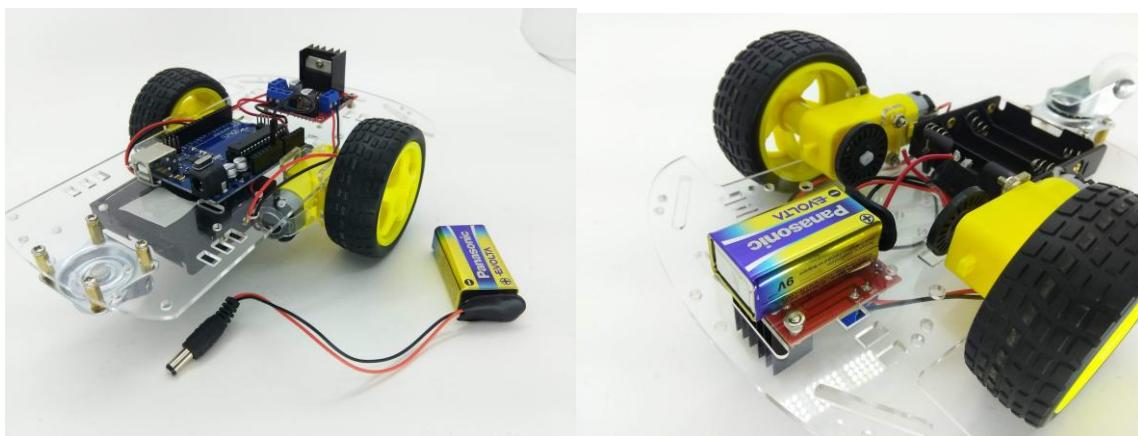
Iremos conectar os fios de alimentação na ponte H. Vamos colocar o fio vermelho, que sai do suporte de pilhas, no borne 6V do módulo ponte H, e o preto do suporte de pilhas no borne de GND. Para consultar essas informações, basta ver a imagem da ponte H um pouco mais acima, neste post. Para ajudar, ainda nesta etapa das conexões, coloque um jumper macho/macho dentro do borne de GND junto com o fio preto do suporte de energia, que mais tarde ele será usado. A ligação ficou assim:



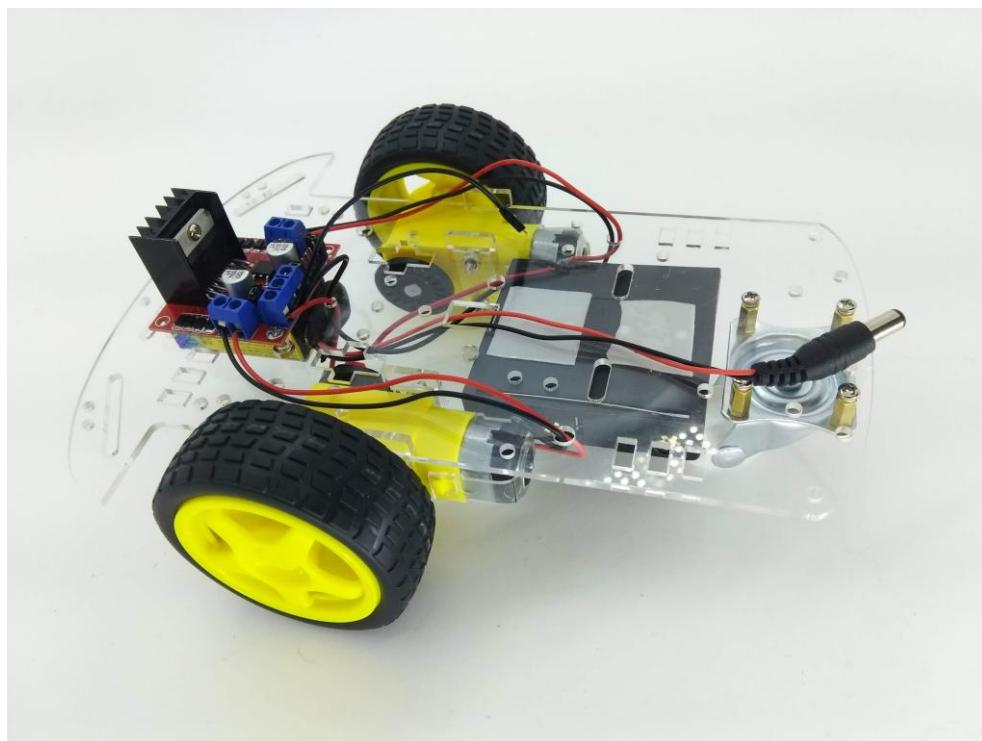
A P O S T I L A K I T  
**ARDUINO KIDS**

O jumper colocado é para ser usado sendo colocado na porta GND do Arduino. Agora só está faltando uma bateria de 9V, para fornecer energia ao Arduino. Essa alimentação separada foi escolhida porque as 4 pilhas possuem uma disponibilidade de corrente elétrica um pouco baixa, e ligando todo o circuito no suporte, ele não consegue fornecer energia, e os motores acabam ficando parado.

Colocamos uma fita dupla face na bateria de 9V também, e instalamos ela exatamente embaixo (no chassis) do módulo ponte H. O resultado foi este:

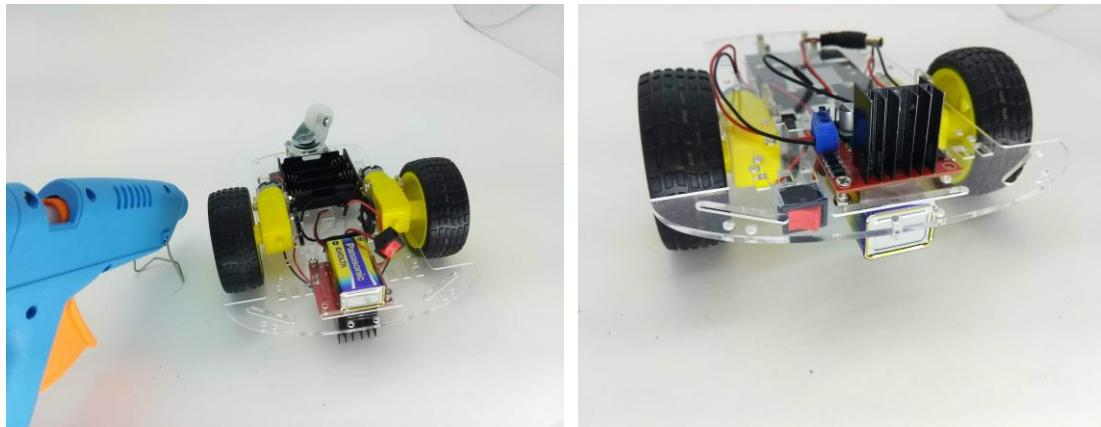


Até o momento nosso carrinho está assim, certo? Se não, revise alguns passos anteriores:

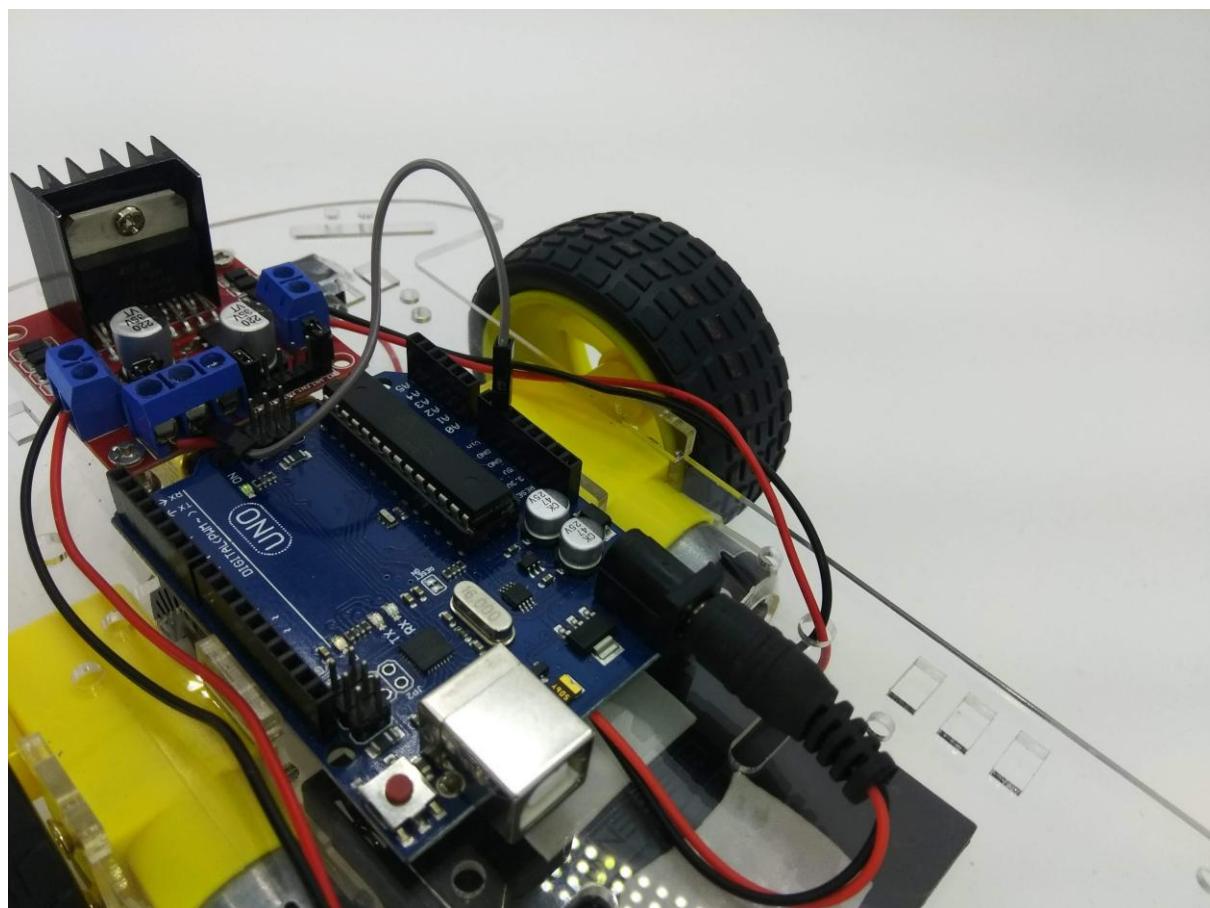


A P O S T I L A K I T  
**ARDUINO KIDS**

Para melhorar um pouco a estética e facilitar o uso, instalamos a chave on/off bem na frente do carrinho, e colamos com cola quente. Para isso precisamos estender um pouco mais o fio.

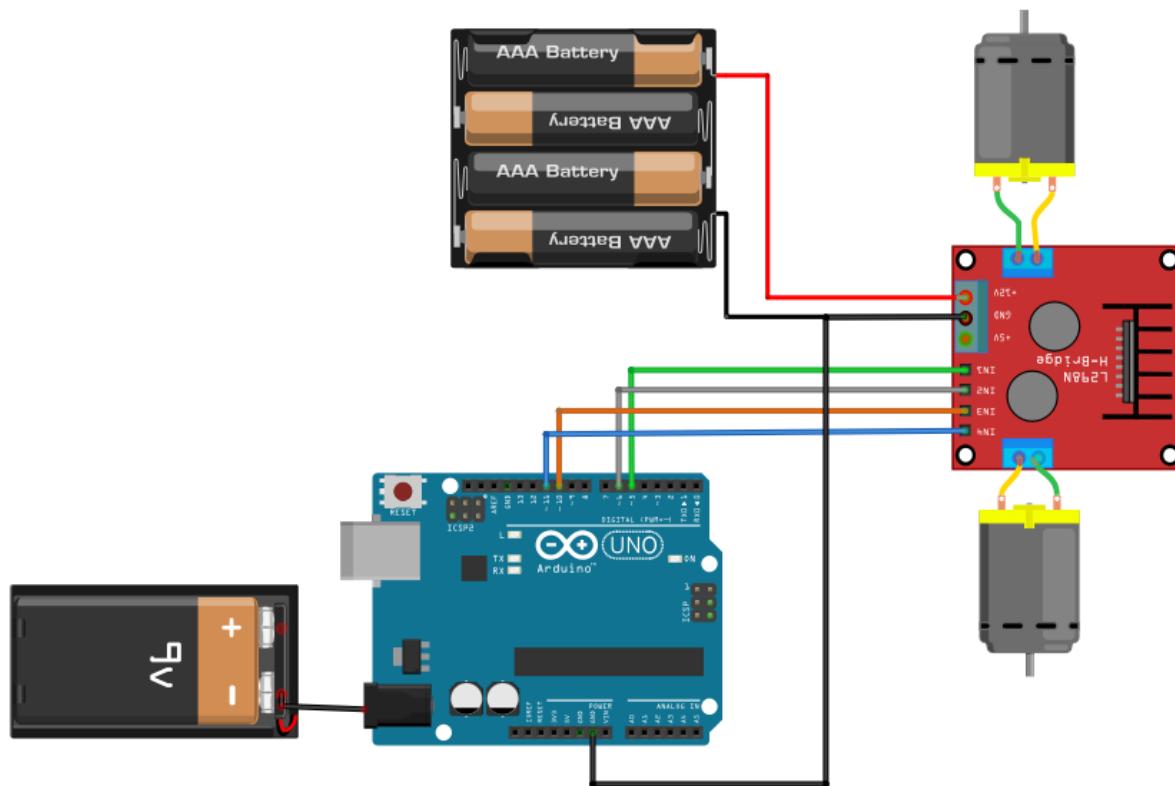


Então, agora, para termos um carrinho já funcional só falta colocar na montagem a placa do Arduino e fazer corretamente as ligações. A conexão com o Arduino pode ser feita da seguinte forma:



A POSTILA KIT  
**ARDUINO KIDS**

Colocando a placa na parte central. As conexões da ponte H com o Arduino serão feitas da seguinte forma:



Observações:

- Não precisa remover nenhum jumper da placa. Todos eles podem ser mantidos.
- Ao colocar o cabo USB, remova a alimentação da bateria de 9V.

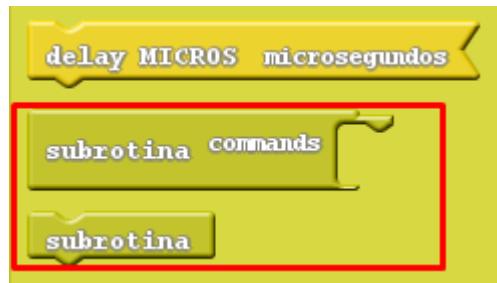
Com o carrinho montado e com todas as conexões finalizadas, testar ele será simples: Vamos usar um código no Ardublock:

### Hora da programação

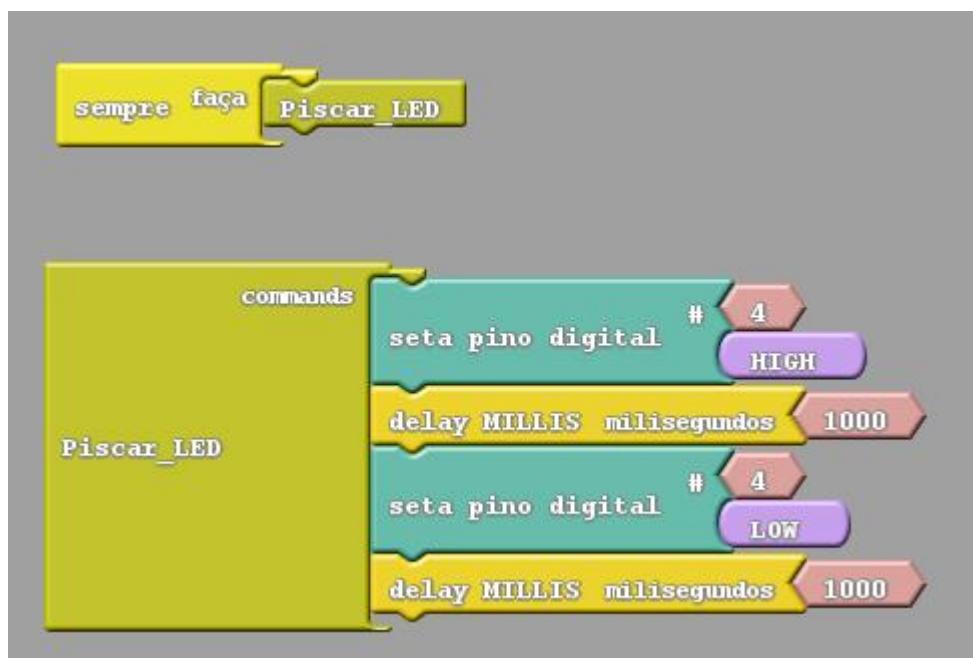
Para este código, iremos aprender um conceito novo: o de subrotina no Ardublock.

A P O S T I L A K I T  
**ARDUINO KIDS**

Os comandos de **subrotina** no Ardublock estão localizados na parte amarela chamada “Controle”.



Esses dois comandos servem para podermos “criar” um novo bloco a partir dos outros blocos já feitos. Para usar, basta colocar o bloco de comando inteiro dentro do “subrotina commands”, e você poderá renomear esse novo bloco criado pelo bloco que você quiser. Por exemplo, iremos criar um novo bloco que não existe, o de piscar um LED hipotético que está na porta 4.



Com isso, criamos o bloco “Piscar\_LED”, baseado nos blocos que já existem. No exemplo do carrinho isso fará com que nossa programação fique muito mais fácil.

A P O S T I L A K I T  
**ARDUINO KIDS**

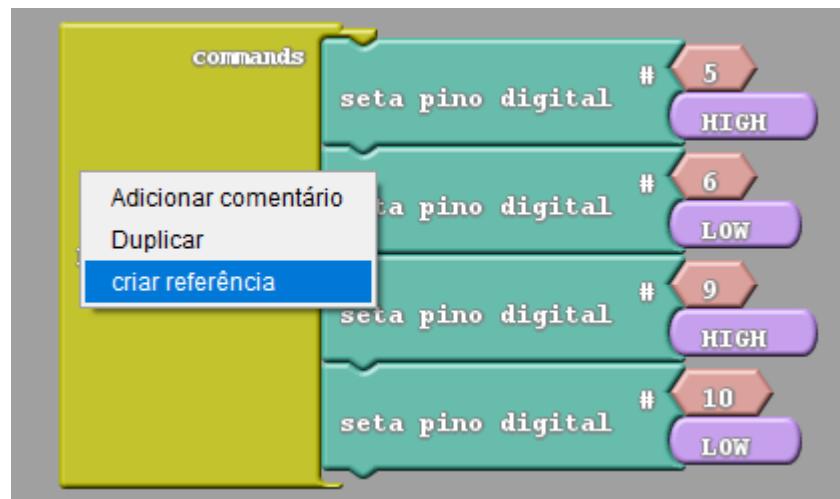
Para o carrinho, iremos criar alguns blocos novos, e eles são:



Onde, o “Giro1” é para girar em um sentido, e o “Giro2” no outro. Criando isso, ficará muito simples programar o carrinho para fazer o que queremos. É muito importante salvar estes comandos criados, porque pode ser útil posteriormente para automatizar o carrinho e fazer trajetos como você quiser.

Uma dica boa é: clique com o botão direito do mouse em cima do bloco, e vá em “criar referência”.

A POSTILA KIT  
**ARDUINO KIDS**



E então surgirá um novo bloco para ser usado. E então, para enfim testar nosso carrinho, faremos o seguinte código:



E o carrinho irá fazer um caminho. Aproveite para fazer novos desafios com o seu carrinho, o fazendo percorrer trilhas diferentes e caminhos diferentes.

## Considerações finais

Essa apostila tem por objetivo apresentar alguns exemplos básicos para crianças e de como fazer o manuseio dos produtos, com foco no Ardublock, dando exemplos mais em formas de projetos para poder aguçar o desejo do aprendizado de forma prática e divertida.

Esta apostila tem foco na linguagem de programação visual, em blocos, e é recomendado que ao familiarizar com este tipo de programação, comece a usar a linguagem escrita, pois, ela dá mais funcionalidades e maiores possibilidades de uso para a placa.

Para a aprendizagem e uso de linguagem de programação, a página oficial e o fórum do Arduino são muito indicados, pois o fórum oficial Arduino possui muitas discussões e exemplos muito bons. A comunidade de desenvolvedores é bastante ativa e certamente pode te ajudar em seus projetos. No Project Hub poderá encontrar milhares de projetos com Arduino.

- Fórum oficial Arduino: <https://forum.arduino.cc/>
- Project Hub Arduino: <https://create.arduino.cc/projecthub>

O Instructables é a ótima referência do mundo maker atual. Pessoas que buscam construir suas próprias coisas e projetos encontram referências e compartilham suas experiências no site.

- Instructables: <https://www.instructables.com/>

O Maker Pro é outro site referência no mundo em relação aos projetos com Arduino. Há uma infinidade de projetos, todos bem explicados e com bom conteúdo.

- Maker pro: <https://maker.pro/projects/arduino>

No mais, esperamos que essa apostila seja apenas o início de vários outros projetos e, quem sabe, a iniciação em outros kits com foco na programação escrita e que possuem uma gama muito maior de componentes, como o kit Maker e o kit Advanced. Qualquer dúvida, sugestão, correção ou crítica a esse material, fique à vontade para relatar em nosso blog oficial: <http://blog.eletrogate.com/>

A P O S T I L A   K I T

# ARDUINO KIDS

Esta apostila acompanha o **Kit ARDUINO KIDS**  
da Eletrogate, e contém conteúdos relacionados  
a todos os componentes do Kit.

**WWW.ELETROGATE.COM**



ELETROGATE

A P O S T I L A   K I T  
**ARDUINO KIDS**

**Sobre o autor:**

**Gustavo Nery**

Graduado de Engenharia de Controle e Automação pela UFMG. Apaixonado por eletrônica, computação e tecnologias na área de sistemas embarcados. Nos tempos livres se divide entre desenvolver pesquisa na universidade, adquirir novos conhecimentos e estar com a família.

Agradecimentos ao **Gustavo Murta** pela revisão e sugestões.