OFICINAS SYNESTHESIA VISION

ESP32 E KODULAR

Manual do Estudante

Aida A. ferreira

Lucas A. Barbosa

Paulo E. G. Vieira

Isaque D. S. Silva

Gilmar G. de Brito

Ioná M. B. Rameh

Vânia S. de Carvalho

Ronaldo R. B. Aquino

OFICINAS SYNESTHESIA VISION

Manual do Estudante

Sobre

Material didático desenvolvido pela equipe do Projeto Oficinas Synesthesia Vision do IFPE, *Campus* Recife

Coordenadora

Aida Araújo Ferreira

Equipe

Lucas Alves Barbosa
Paulo Emilio Gestosa Vieira
Isaque Domingos Santana Silva
Gilmar G. Brito
Ioná M. B. R. Barbosa
Vânia S. de Carvalho
Ronaldo B. B. de Aquino

2ª edição

2023

INSTITUTO FEDERAL
Pernambuco

CENTRO DE PESQUISA IFPE

GRENDES *Lab*Geo

Dados Internacionais de Catalogação na Publicação (CIP)
(Câmara Brasileira do Livro, SP, Brasil)

Oficinas synesthesia vision : ESP32 e kodular
[livro eletrônico] : manual do estudante /
coordenadora Aida Araújo Ferreira. -- 2. ed. --
Recife, PE : Ed. dos Autores, 2023.
PDF

Vários autores.
Bibliografia.
ISBN 978-65-00-77014-8

1. Ciência da computação 2. Dispositivos eletrônicos 3. Eletrônica 4. Programação (Computadores) 5. Prototipagem 6. Robótica - Estudo e ensino 7. Tecnologia Assistiva (TA)
I. Ferreira, Aida Araújo.

23-167617 CDD-372.358

Índices para catálogo sistemático:

1. Robótica : Estudo e ensino 372.358

Aline Graziele Benitez - Bibliotecária - CRB-1/3129

Sumário

Introdução

O objetivo desta apostila é a capacitação teórica e prática dos estudantes em conceitos básicos de eletrônica, programação, robótica e prototipagem de dispositivos tecnológicos, através da realização de Oficinas de Robótica Livre. Visamos também a conscientização dos participantes em relação à importância do desenvolvimento de tecnologias assistivas para auxiliar na independência de pessoas com necessidades especiais.

Com carinho, equipe Synesthesia Vision.

Tecnologias Assistivas

1 - Introdução

Desde os primórdios, a tecnologia sempre foi essencial para a evolução e ampliação da nossa sociedade. Atualmente, e mais do que nunca, a tecnologia está presente no nosso cotidiano. Seja para controlar a luz de um ambiente, para se comunicar ou até mesmo para se alimentar, o propósito da tecnologia é facilitar o nosso cotidiano, eliminando barreiras a fim de acelerar os processos de forma a gerar resultados mais rápidos. Ou seja, a tecnologia serve para ajudar a tornar o mundo mais prático para nós.

Mas não podemos esquecer que existem pessoas com necessidades especiais, por isso, devemos criar tecnologias visando atender às necessidades de todos, independentemente de sua condição física ou mental. Um teclado “comum” de computador, por exemplo, não poderia ser utilizado por uma pessoa com deficiência visual (visto que não enxergaria as letras), mas poderia usar um teclado em braile. Do mesmo modo, uma escada não poderia ser utilizada por um cadeirante, mas uma rampa poderia. O que esses dois exemplos citados têm em comum? Ambos promovem a inclusão através da Tecnologia Assistiva.

2 - O que é Tecnologia Assistiva?

Quando a palavra “Tecnologia” vem em nossas cabeças, pensamos imediatamente em algo eletrônico, como um computador, um celular, um videogame e até mesmo algo que custe muito caro. E, de fato, a tecnologia pode ter um custo elevado, mas quando quando falamos de Tecnologia Assistiva, como um aparelho auditivo ou uma cadeira de rodas motorizada, ela vai muito além disso. Um produto de tecnologia assistiva, por exemplo, pode ser um piso tátil que facilite a mobilidade de um deficiente visual, uma prótese que permita à pessoa com deficiência física caminhar, inclusive a linguagem de sinais pode ser considerada como tal por permitir a comunicação de pessoas com deficiência auditiva.

Para compreender melhor isso, vamos começar pensando na etimologia de “Tecnologia Assitiva”:

Tecnologia: é um produto da ciência e da engenharia que envolve conjunto de instrumentos, métodos e técnicas que visam à resolução de problemas.

Assistiva: vem de ajudar, acompanhar.

Entendido isso, pense em Tecnologia Assistiva como um produto projetado para ajudar pessoas com deficiência, de forma que promova maior conforto, segurança, mobilidade, autonomia, ampliação de sua comunicação e melhoria em seu aprendizado.

3 - Qual o objetivo da Tecnologia Assistiva?

Bom, até aqui você já deve ter uma noção de que tecnologia assistiva é um produto ou recurso que visa ajudar pessoas com deficiência. Vamos agora destrinchar isso melhor e de acordo com a lei nº 13.146/2015, conhecida como a Lei Brasileira de Inclusão da Pessoa com Deficiência, a tecnologia assistiva é definida como:

De acordo com a constituição, produtos, equipamentos, dispositivos, recursos, metodologias, estratégias, práticas e serviços que objetivem promover a funcionalidade, relacionada à atividade e à participação da pessoa com deficiência ou com mobilidade reduzida, visando à sua autonomia, independência, qualidade de vida e inclusão social.

Logo, o objetivo da Tecnologia Assistiva é a inclusão social, proporcionando à pessoa com deficiência maior independência, qualidade de vida, ampliação de sua comunicação, mobilidade, controle de seu ambiente, aprimoramento em suas habilidades de trabalho e aprendizado.

4 - Qual a importância das Tecnologias Assistivas?

Que existem diversas pessoas com diferentes deficiências, você já deve saber. Em números exatos, de acordo com o último senso realizado pelo IBGE em 2010, existem mais de 45 milhões de pessoas portadoras de alguma deficiência. Entretanto, não são apenas pessoas com deficiência que podem usufruir de um produto ou recurso de Tecnologia Assistiva. Qualquer pessoa, seja ela deficiente, com mobilidade reduzida ou até mesmo alguém que não possua nenhuma deficiência, pode se beneficiar com esse tipo de tecnologia. Uma grávida que precisa banhar-se em uma cadeira de banho, um idoso necessitando de uma bengala, ou até mesmo uma criança que torceu o tornozelo jogando bola e que venha precisar de uma órtese, qualquer pessoa pode vir a utilizar um produto ou recurso de tecnologia assistiva. Por isso, a criação e elaboração dessas tecnologias são tão importantes.

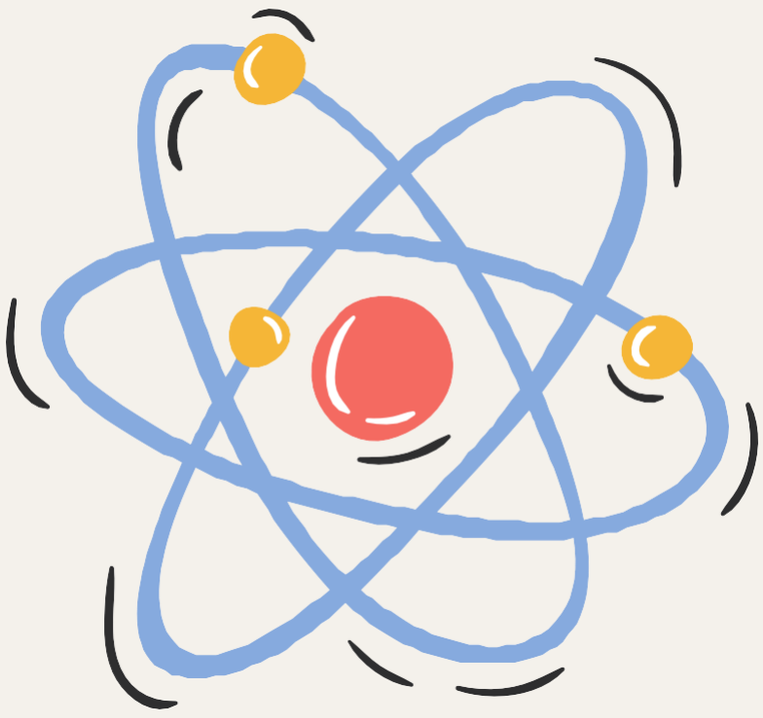
5 - E o que podemos concluir disso tudo?

Um produto ou recurso de tecnologia assistiva é um grande amigo da inclusão e possibilita amplo acesso a produtos e serviços, ampliando a independência, comunicação e mobilidade. Assim, proporcionando maior qualidade de vida a quem a utiliza. A tecnologia assistiva é sim uma área interdisciplinar do conhecimento que pode demandar meses ou até mesmo anos de estudos, além de testes, com um custo elevado. Porém, ela também pode ser algo de baixo custo, prático e simples, pois o

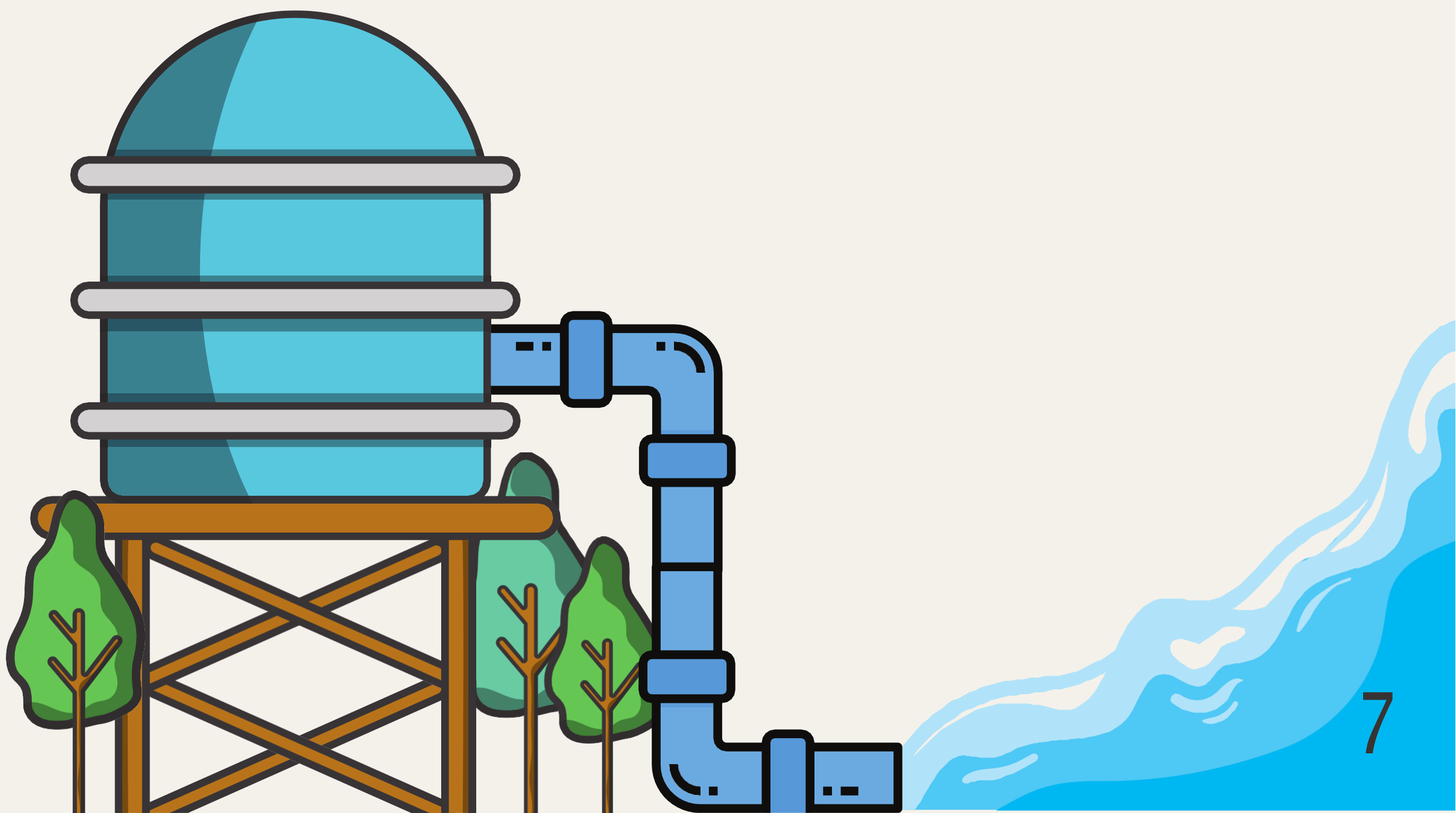
objetivo é o mesmo: ajudar pessoas com deficiência. Ficou curioso, com vontade de criar algum produto de tecnologia assistiva e não sabe como? Continue lendo essa apostila e participando ativamente das Oficinas Synesthesia Vision para saber como você pode contribuir para um mundo mais inclusivo.

Eletrônica

1 - O que é eletricidade?

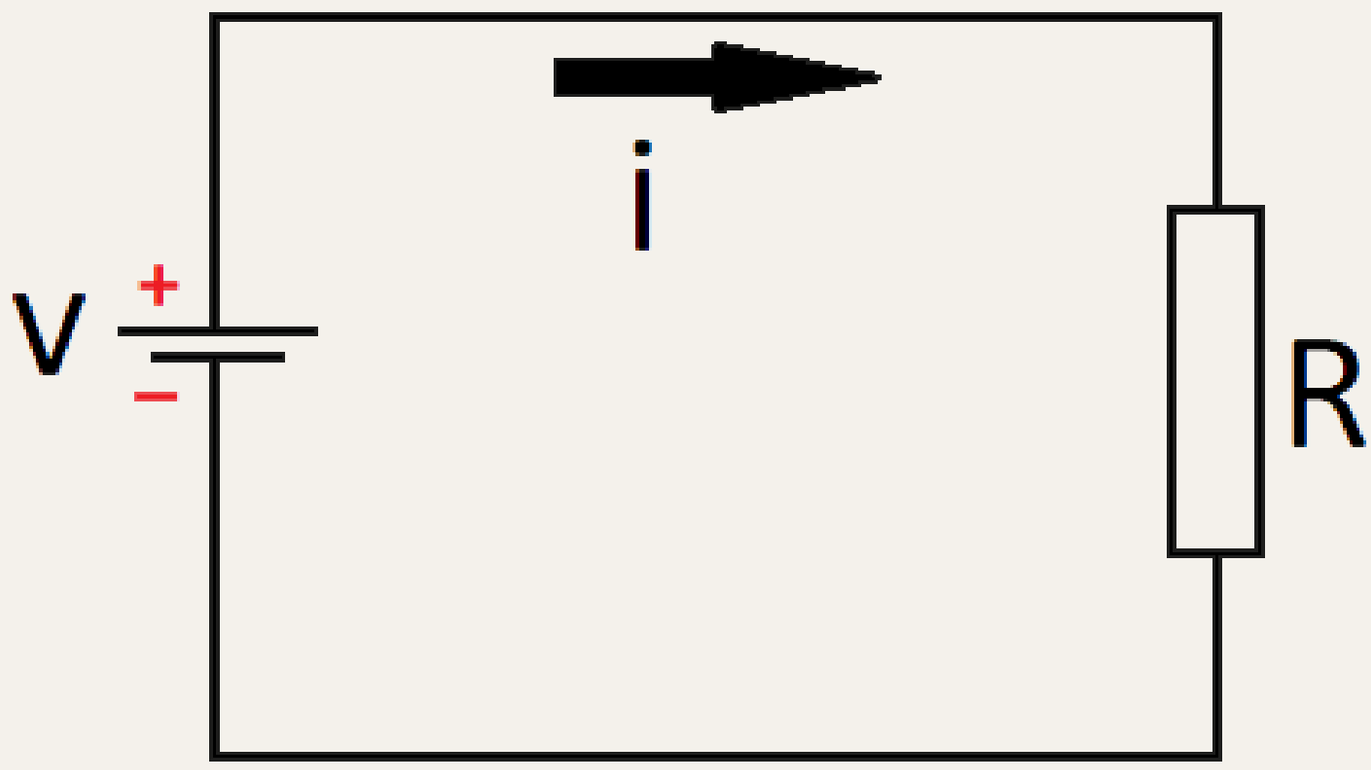


Tudo que existe no universo é formado por pequenas partículas chamadas átomos. Todos os átomos possuem elétrons que giram indeterminadamente ao seu redor e possuem carga negativa. Quando um átomo perde um de seus elétrons, ele passa a ter carga positiva e tenta atrair elétrons de outros átomos. Essa força que atrai elétrons é chamada de diferença de potencial ou tensão elétrica. Quando a diferença de potencial move elétrons de um átomo para outro, ocorre a formação de corrente elétrica. Já a capacidade de um material de se opor à passagem de corrente elétrica, é chamada de resistência elétrica. Essas são as principais medidas elétricas usadas na eletrônica. Para entender melhor esses conceitos observe o exemplo abaixo e a explicação desta imagem na próxima página.

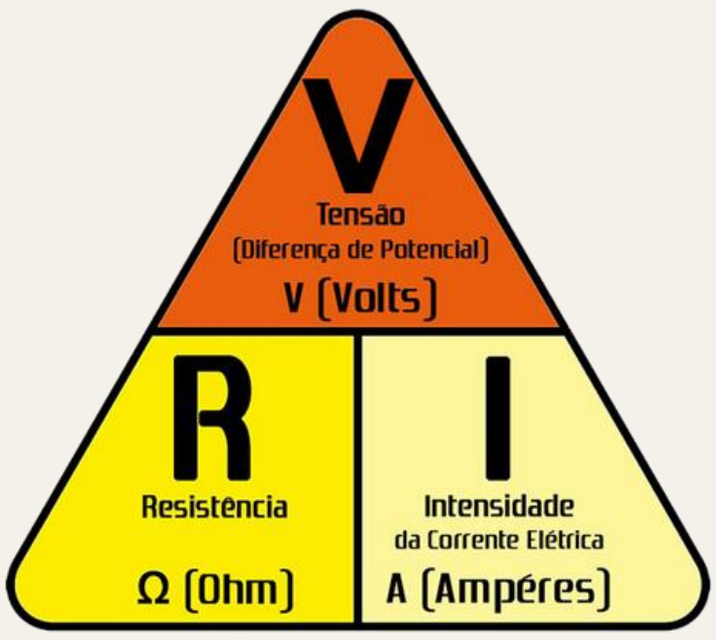


Imagine que nós tenhamos uma caixa d'água que está no telhado de uma casa a 5 metros do chão. Essa caixa está ligada em um cano que desce até o solo. O fato dessa caixa d'água estar em uma altura elevada faz com que a água possa descer por esses canos até embaixo. Nesse cenário, a altura de 5 metros pode ser comparada com a tensão elétrica - que é medida em Volts. A água que desce pelo cano pode ser comparada com a corrente elétrica - que é medida em Amperes. Se nós tivermos um cano mais grosso, mais água pode fluir, mas, se tivermos um cano mais fino, menos água pode fluir. Nesse caso, o cano é como a resistência elétrica - medida em Ohms.

2 - Circuitos



A imagem acima é a representação de um circuito no qual temos uma fonte de tensão (representada por V) conectada a um resistor (representado por R). A conexão desses dois componentes faz com que apareça uma corrente elétrica em um determinado sentido (representado por I). O resistor é o elemento responsável por limitar a corrente que circula nesse circuito. É possível calcular qual o valor dessa corrente através da Lei de Ohm, que utiliza a fórmula V = R x I, representada pelo esquema abaixo:



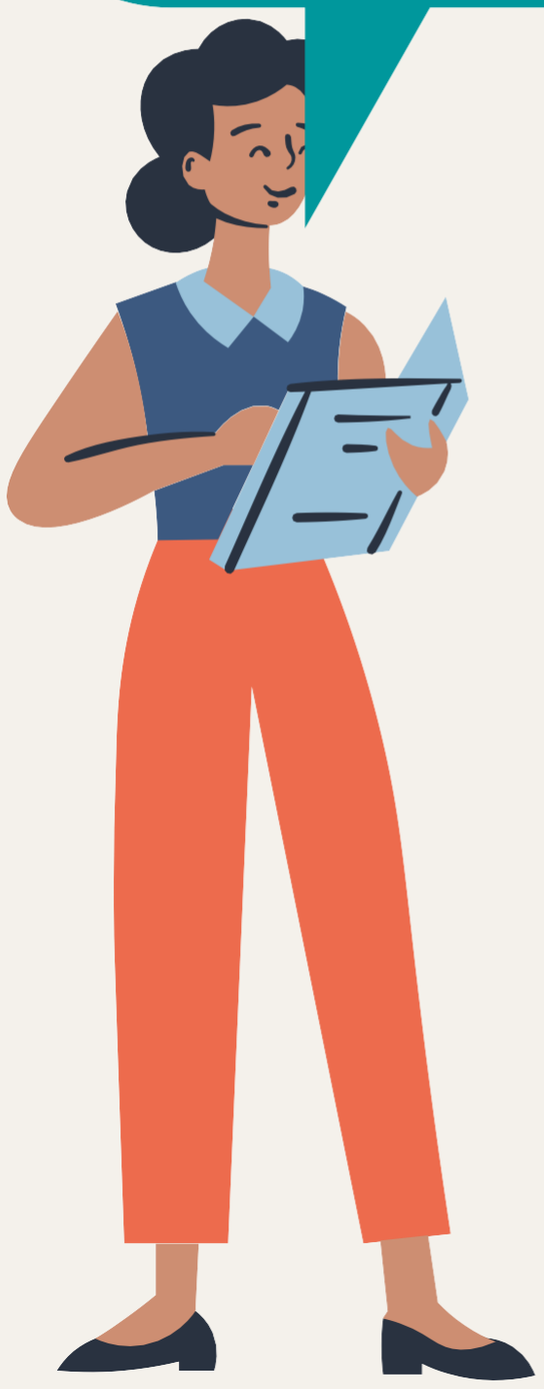
ESP32

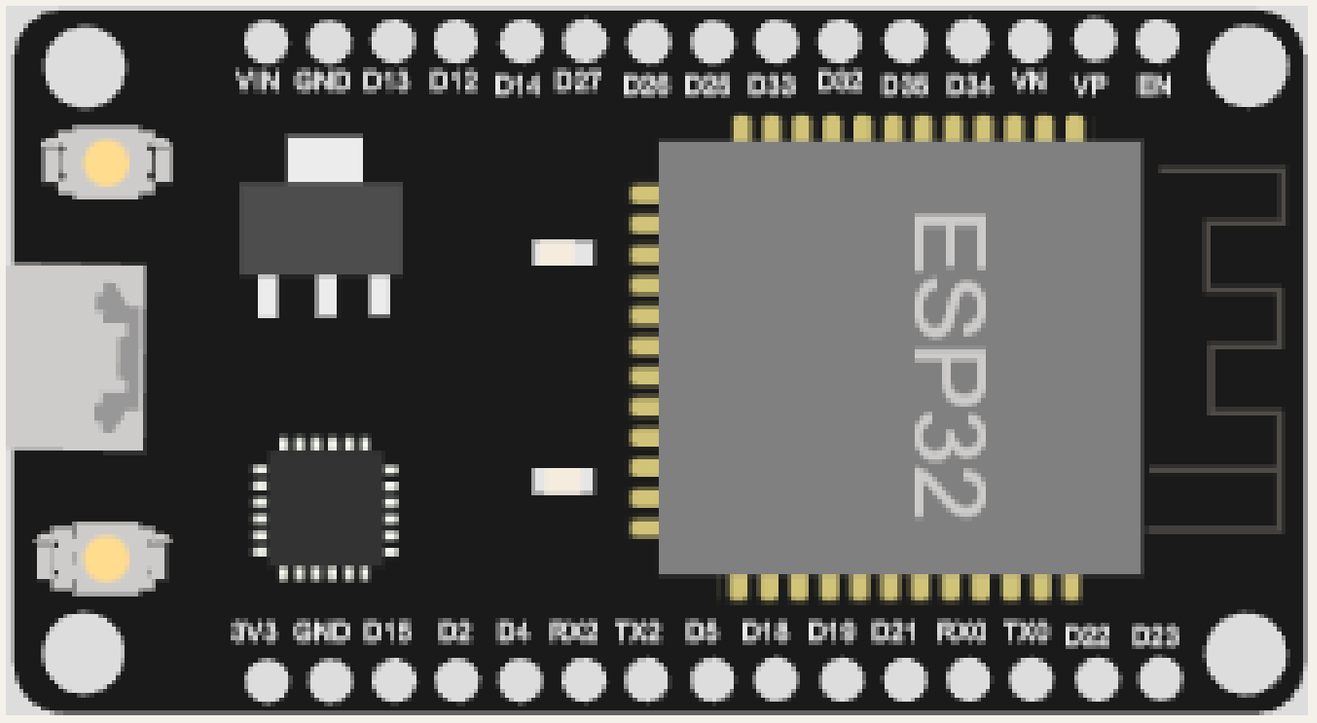
1 - Introdução

A ESP32 é uma plataforma de desenvolvimento, projetada pela Espressif Systems. Se você está começando com eletrônica e programação, a ESP32 é uma excelente opção. Com sua versatilidade, recursos avançados e ampla documentação, a ESP32 é uma das melhores placas para iniciantes.

Com a ESP32, você terá uma placa robusta e confiável para experimentar e aprender. Seus recursos incluem uma ampla gama de pinos de entrada e saída digital, entrada analógica, comunicação serial, PWM e muitos outros, que permitem uma ampla gama de projetos. Além disso, a ESP32S possui recursos avançados, como suporte a Wi-Fi e Bluetooth, o que a torna uma escolha poderosa para projetos de IoT (Internet das Coisas).

Esta é a ESP32, uma das melhores placas para começar com eletrônica e programação.



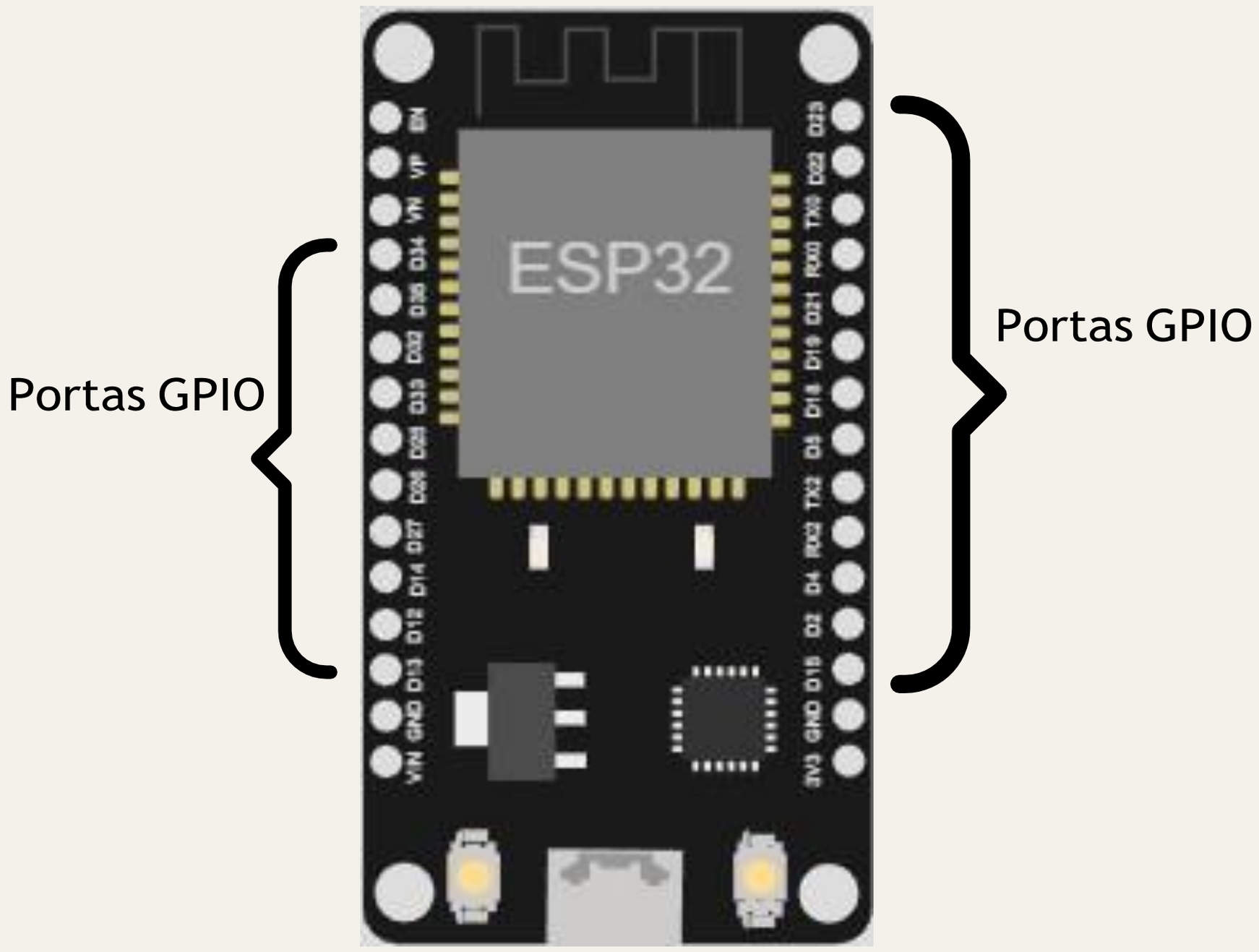


2 - Portas da ESP32

Os pinos da ESP32 são os pontos de conexão física presentes no módulo ESP32, uma plataforma de desenvolvimento baseada em microcontrolador da família ESP32, desenvolvida pela Espressif Systems. Os pinos são usados para conectar a ESP32 a outros dispositivos e componentes eletrônicos, como sensores, atuadores, displays, entre outros.

A ESP32 possui uma ampla variedade de pinos, que incluem pinos de entrada e saída digitais, pinos de entrada analógica, pinos de alimentação, pinos de comunicação serial (UART, SPI, I2C), pinos de controle de PWM (Pulse Width Modulation), pinos de interface de câmera, entre outros. Cada pino possui uma funcionalidade específica e pode ser configurado pelo programador para realizar diferentes tarefas, de acordo com os requisitos do projeto.

Os pinos da ESP32 são identificados por números e letras, e geralmente são dispostos em formatos de pinagem, como o padrão GPIO (General Purpose Input/Output), que permite a conexão de diferentes dispositivos eletrônicos. A documentação oficial da ESP32 fornece detalhes completos sobre os pinos e suas funções, facilitando o desenvolvimento de projetos com essa plataforma versátil e poderosa.

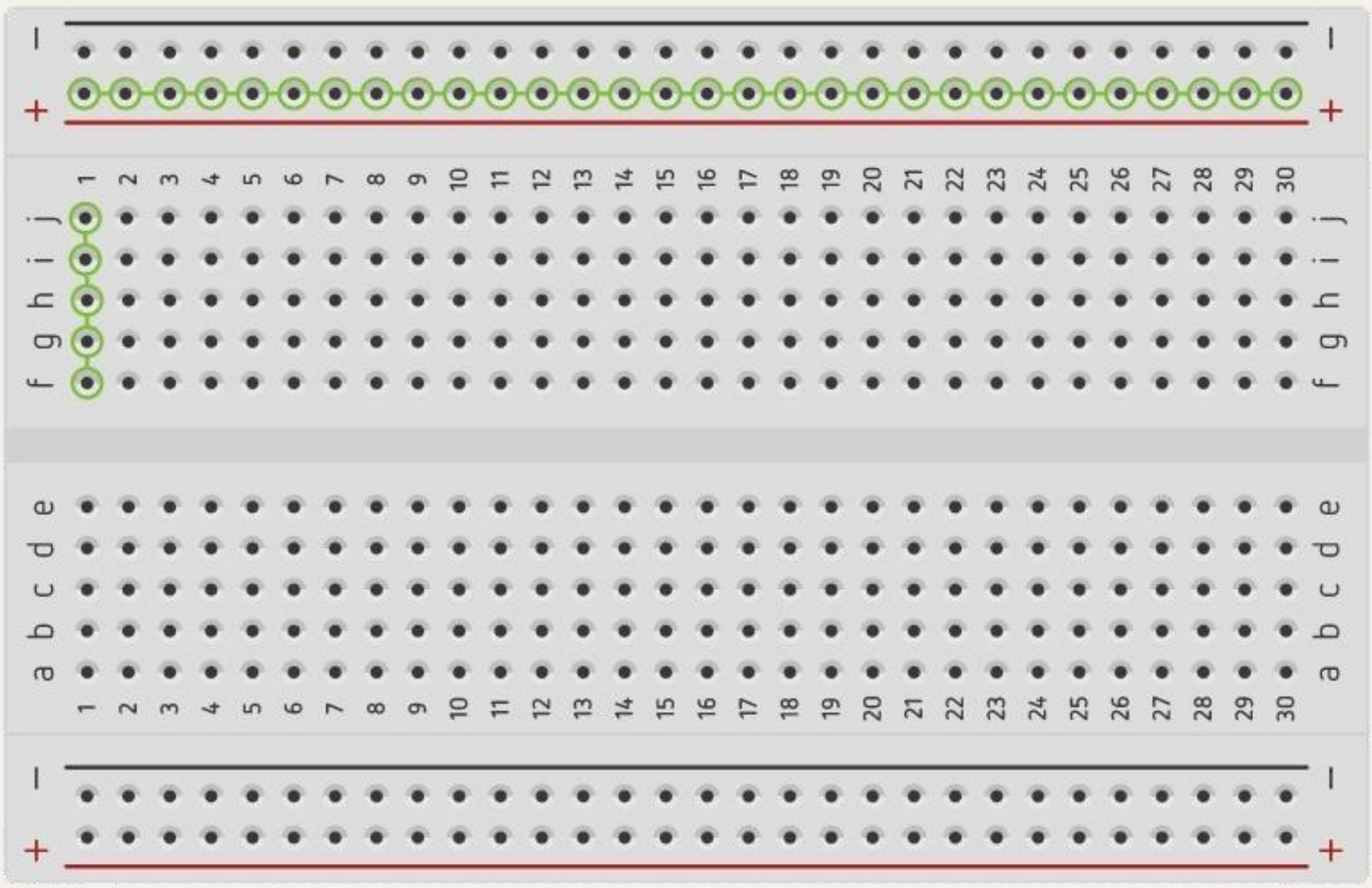


Componentes

Os componentes eletrônicos são as "peças" que utilizamos para montar circuitos com as mais diversas funções. Começaremos pelo mais fudamental deles:

1 - Protoboard

Protoboards são muito utilizadas no aprendizado da eletrônica para a montagem de circuitos de maneira rápida e fácil. Ela é essencial caso deseje montar qualquer tipo de circuito. A grande vantagem das protoboards é que as conexões não são permanentes e podem ser alteradas facilmente. Na imagem abaixo, podemos ver o padrão de conexão interna das protoboards, nos quais todos os pontos das linhas (marcadas com "+" e "-") e todos os pontos das colunas ('j' até 'f' ou 'e' até 'a') estão ligados internamente. Ou seja, as linhas laterais estão conectadas horizontalmente e as linhas enfileiradas por letras estão conectadas verticalmente

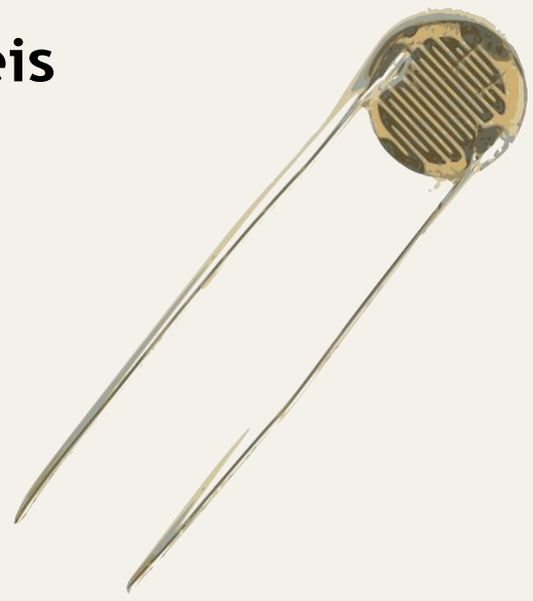


2 - Resistores

Resistores são componentes elétricos que oferecem resistência à passagem de corrente elétrica. Eles são muito utilizados para controlar o valor da corrente em um circuito.

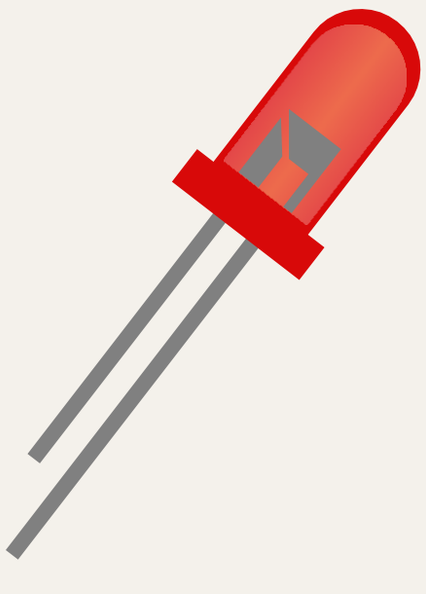
2.2 - LDR

A sigla LDR significa resistor dependente de luz (Light Dependent Resistor). Eles são chamados assim pois sua resistência varia de acordo com a quantidade de luz no ambiente. Esses resistores são muito úteis para desenvolver sensores de luz.



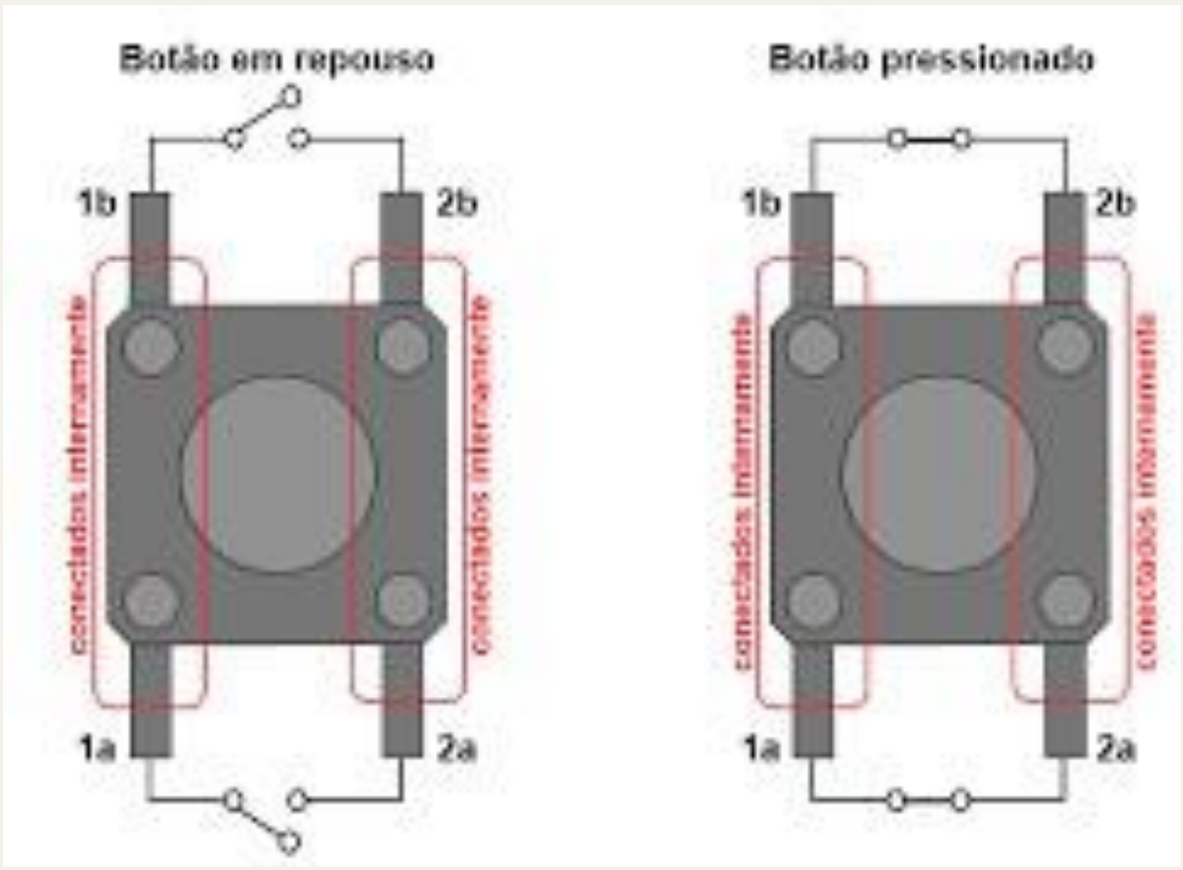
3 - LED

LEDs são componentes capazes emitir luz quando recebem energia elétrica. Estão presentes em quase todos os eletrônicos e geralmente servem como indicadores visuais de algo. Seus polos são identificados através das suas duas hastes (pernas), sendo a menor o polo negativo e a maior o polo positivo



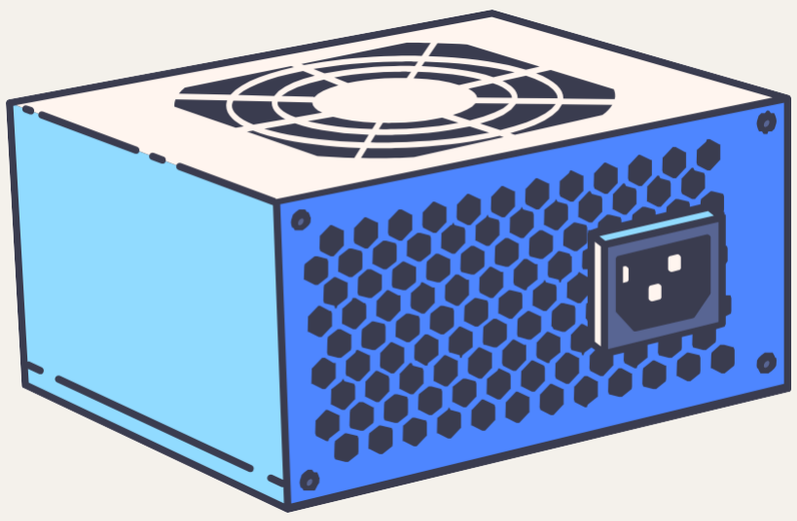
4 - Botão

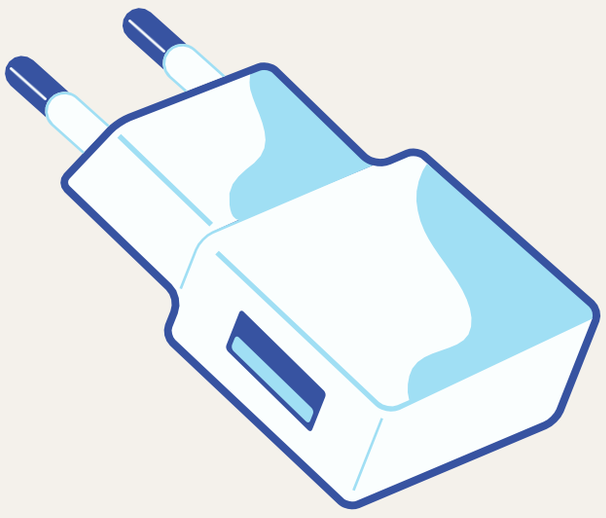
Botões são contatos mecânicos que têm como objetivo controlar a passagem de corrente em um circuito a partir de uma força externa. Geralmente eles tem um contato de ação momentânea, abrindo ou fechando o circuito apenas quando são pressionados.



4 - Fontes de alimentação

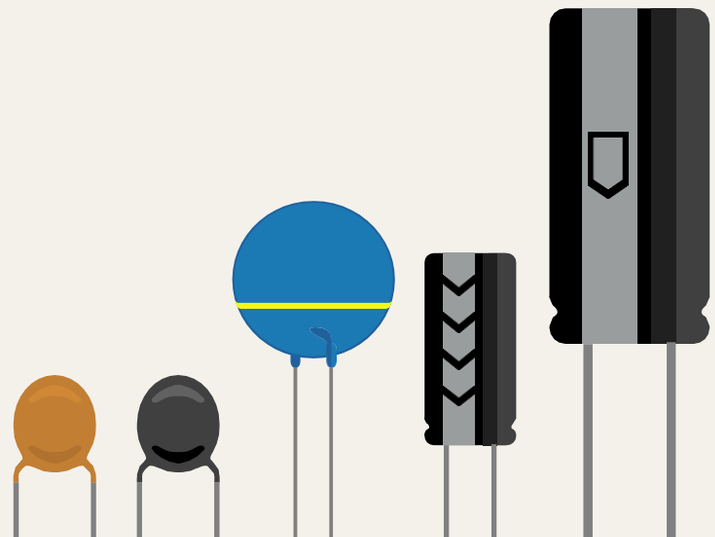
As fontes de alimentação são reponsáveis por prover energia constante aos circuitos. Elas podem ser encontradas em diversos formatos como, por exemplo: pilhas, fontes de computador, carregadores de celular, placas solares e fontes de laboratório que são muito úteis por terem tensão e corrente configuráveis.





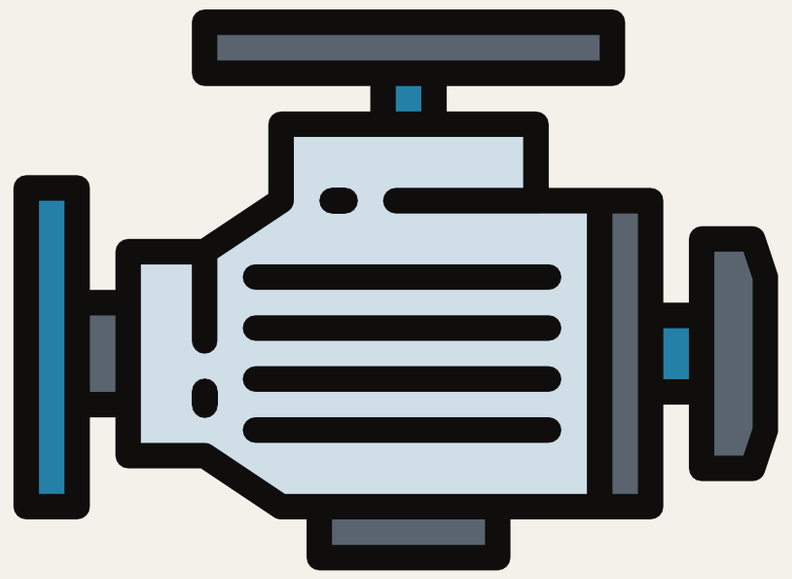
5 - Capacitores

Capacitores são componentes elétricos, isto é, são como baterias ligadas ao circuitos. Uma vez que há energia eles carregam lentamente. E quando a energia da fonte é cortada eles descarregam sua energia gradualmente no circuito. Normalmente são utilizados para garantir estabilidade.



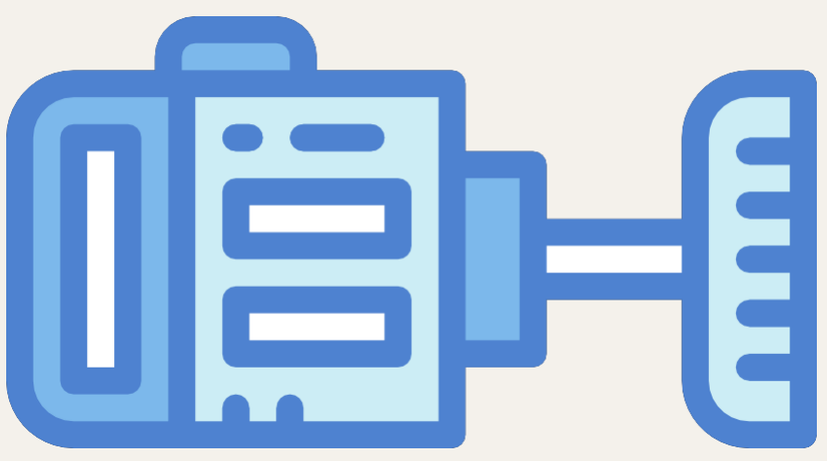
6 - Motores

Motores são, direta e indiretamente, responsáveis pelo movimento de todas as máquinas e robôs ao nosso redor. Seu funcionamento é simples e a velocidade/força de rotação depende da quantidade de energia fornecida pelo circuito.



7 - Servo motores

Assim como os motores, os servo motores são responsáveis por oferecer movimento. No entanto, eles são utilizados para atingir movimentos com angulações precisas e não apenas para rodar. Os servo motores são muito usados em operações que necessitam de precisão mas, que não precisam de velocidade.



9 - Buzzers

Buzzers são pequenos alto-falantes capazes de gerar apitos ou até mesmo notas musicais. Eles são utilizados para gerar alertas e reproduzir sons diversos.

9 - Potenciômetro

Um potenciômetro é um componente eletrônico que possui resistência elétrica ajustável. Ele possui três terminais, ou pontas, que são controlados a partir de um eixo giratório, responsável por ajustar a resistência do dispositivo.

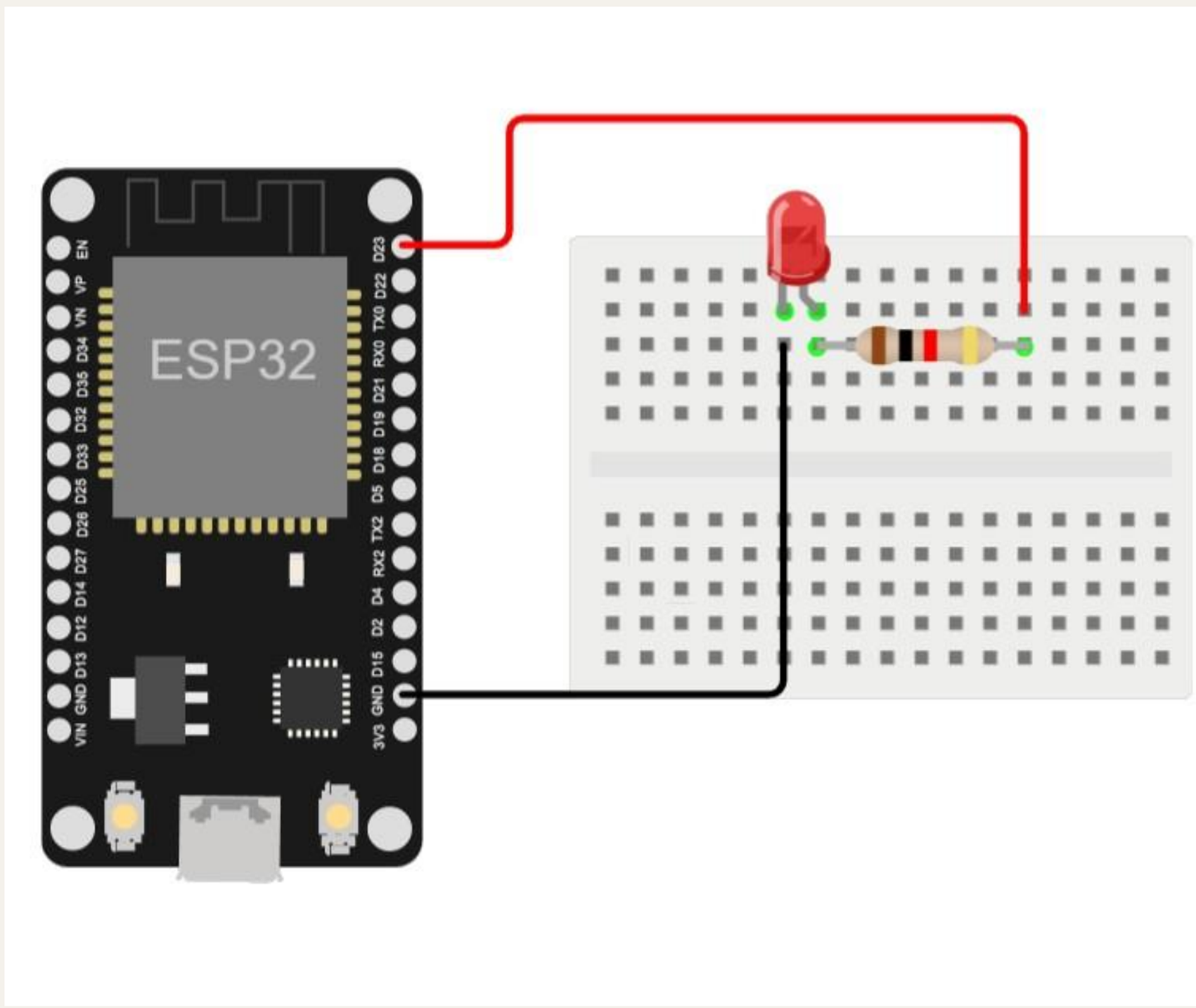
8 - Sensores

Existem diversos tipos de sensores que são capazes de detectar e medir grandezas naturais como distância, temperatura, umidade, pressão entre outras coisas.

Práticas básicas

1 - Acendendo um LED

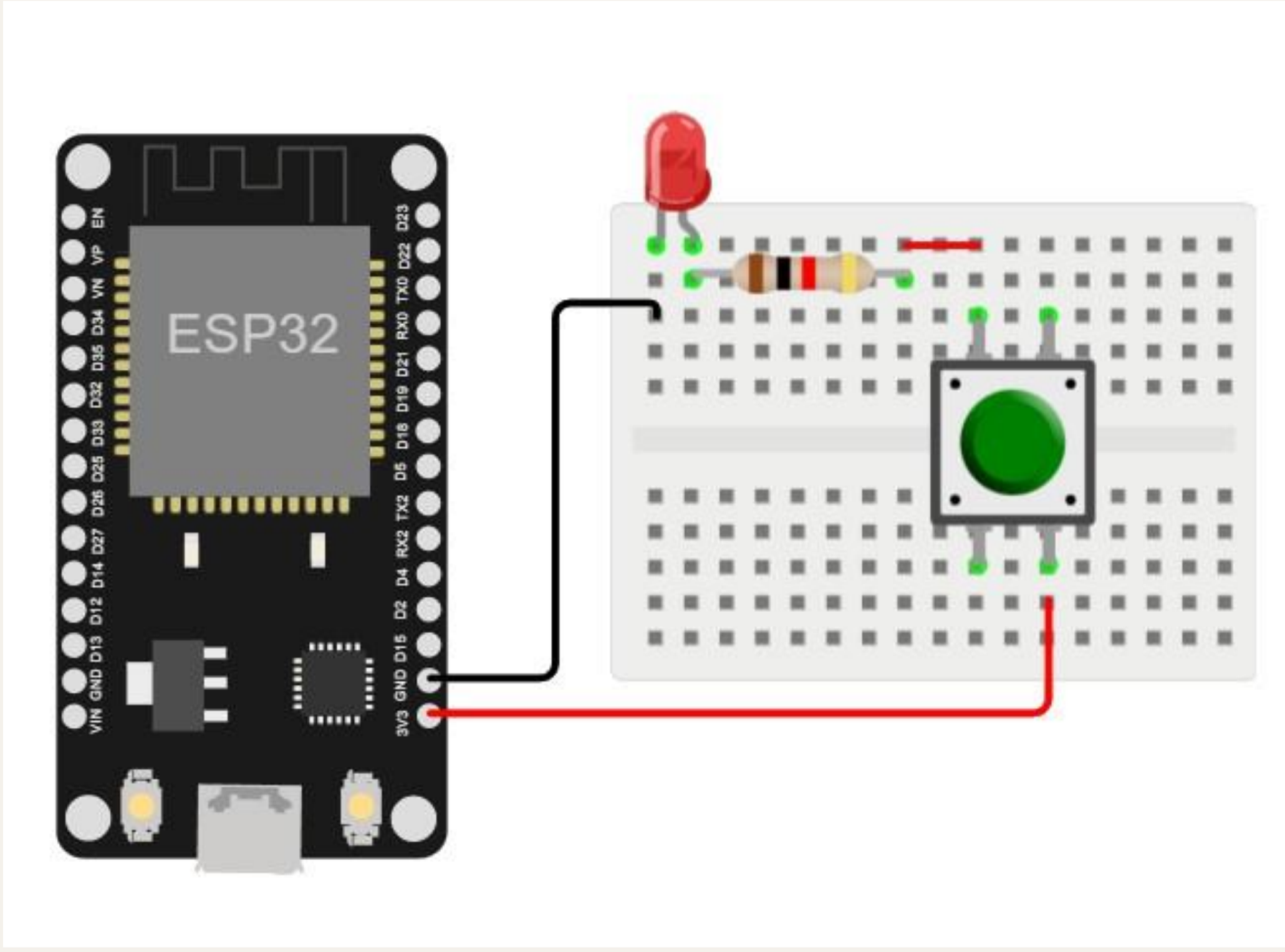
Para essa prática vamos precisar de uma fonte de alimentação (nesse caso usaremos a ESP32 como fonte), uma protoboard, um resistor e, logicamente, um LED. Utilizaremos a protoboard para criar um circuito onde uma fonte de alimentação fornecerá uma corrente para a protoboard, será direcionada para o resistor e através dele, chegará ao LED. Além disso, vale ressaltar que a fonte (3.3V) deve ficar no polo positivo e o GND (terra) no polo negativo.



Você deve estar se perguntando: Por quê a corrente deve passar pelo resistor ao invés de ir diretamente para o LED? A resposta é simples. A quantidade de energia fornecida quando usamos o Arduino como uma fonte, é de 3.3V, mas o LED tem um limite de corrente (energia) que pode receber. Caso passe do limite (3V), o mesmo irá queimar. Então, usaremos o resistor com o objetivo de limitar a corrente para uma quantidade suportada pelo LED.

2 - Acendendo um LED utilizando um botão

Dessa vez, utilizaremos os mesmos componentes de antes, mas iremos acrescentar um botão. Nesse circuito, o botão agirá como uma ponte de passagem. A corrente passará por ele desde que esteja sendo pressionado, caso contrario, a corrente não passará e ficará estagnada até o botão ser apertado novamente.

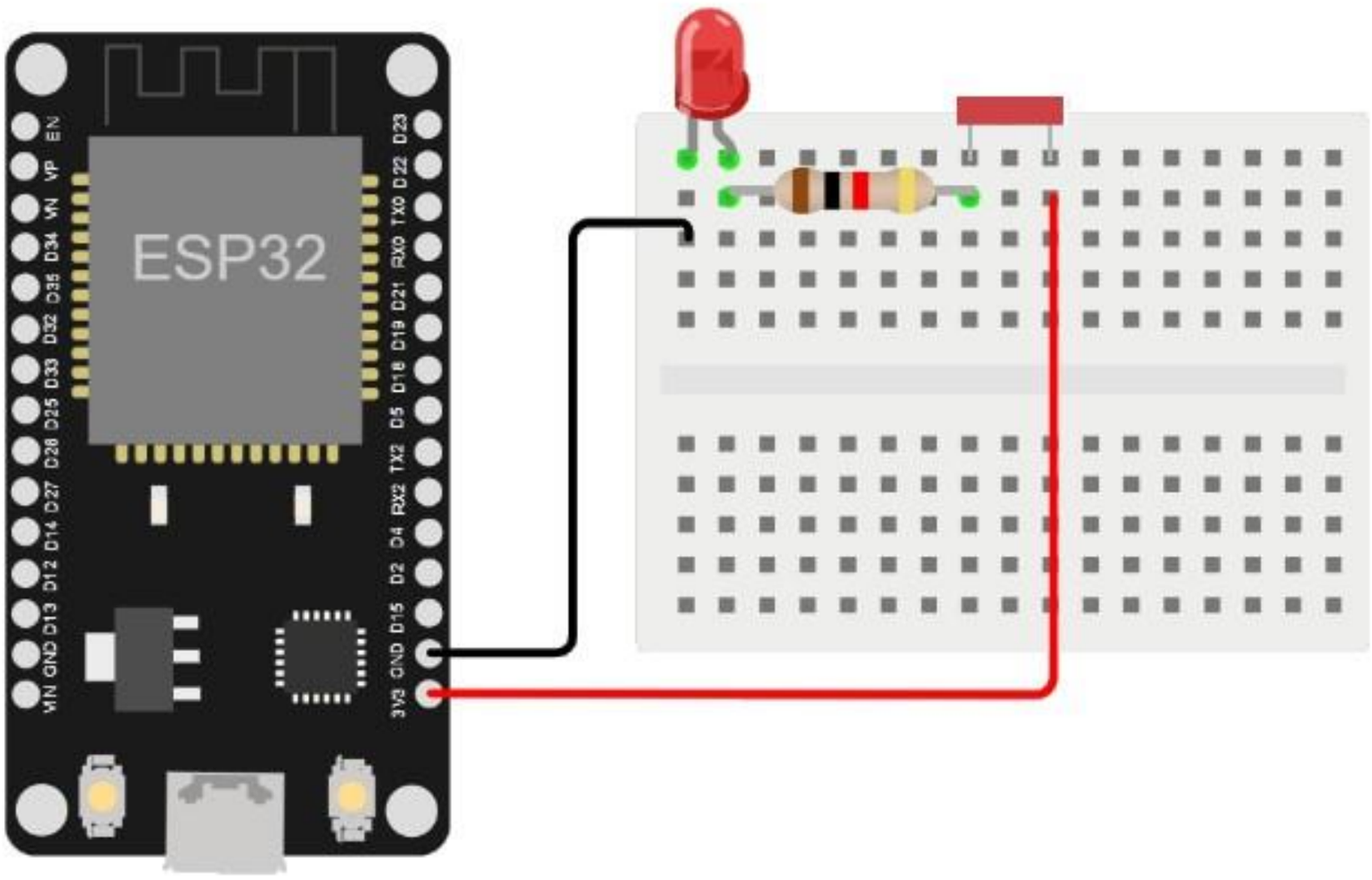


As ligações internas do botão, mostradas anteriormente, fazem com que a ligação precise ser realizada de forma diagonal. Então, a energia terá que passar primeiro pelo botão, seguirá para o resistor e logo em seguida chegará ao LED.

DESAFIO: Tente fazer dois botões acenderem um único LED.

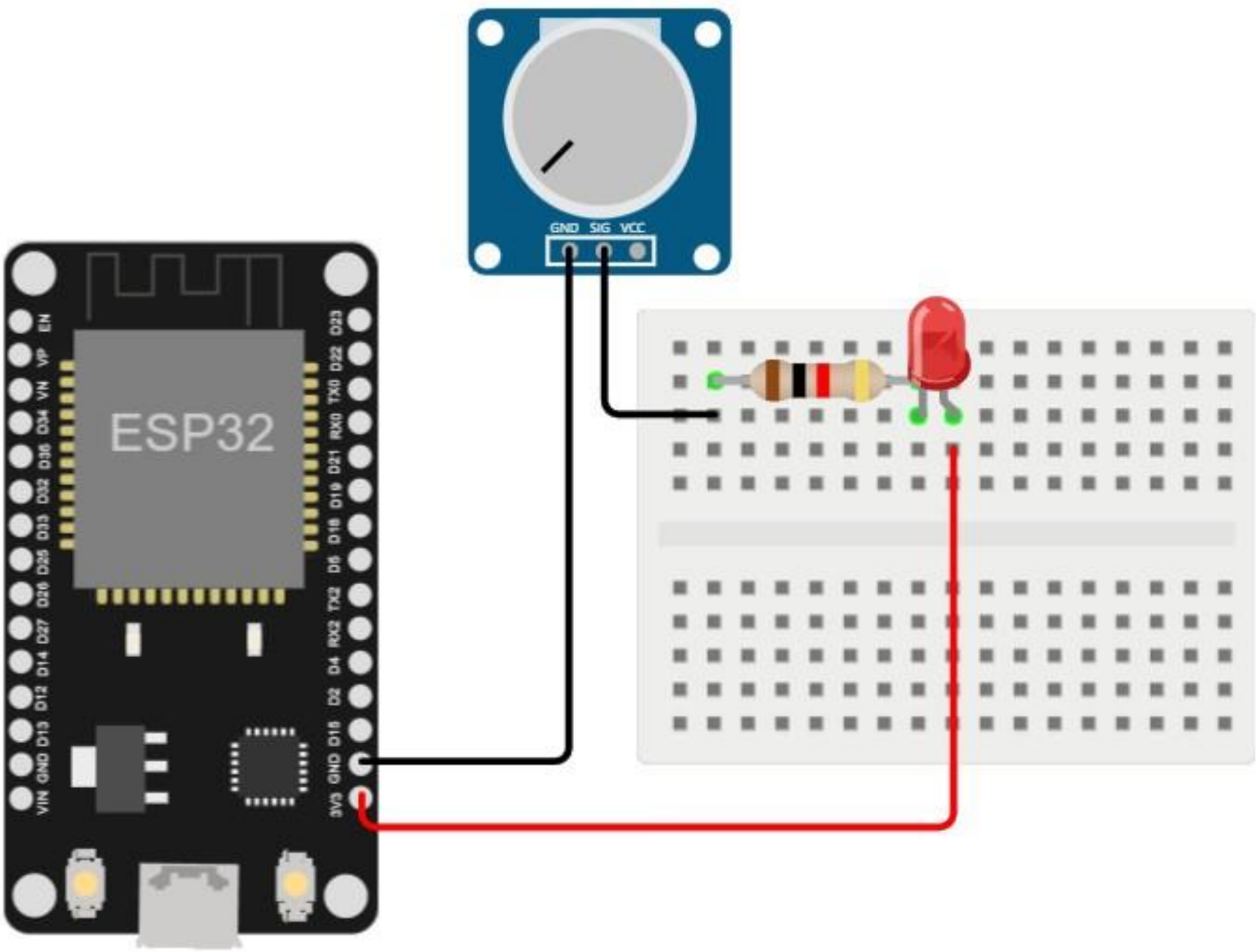
3 - Acendendo um LED com um LDR

Nesta prática, iremos retirar o botão e substituí-lo por um resistor LDR. Como dito anteriormente, o LDR é um resistor sensível a luz, então ele irá oferecer uma resistência que irá variar de acordo com a intensidade da luz. Quando houver ausência de luz, a resistência será maior, já quando houver muita luminosidade, a resistência será menor.



4 - Acendendo o LED com um Potenciômetro

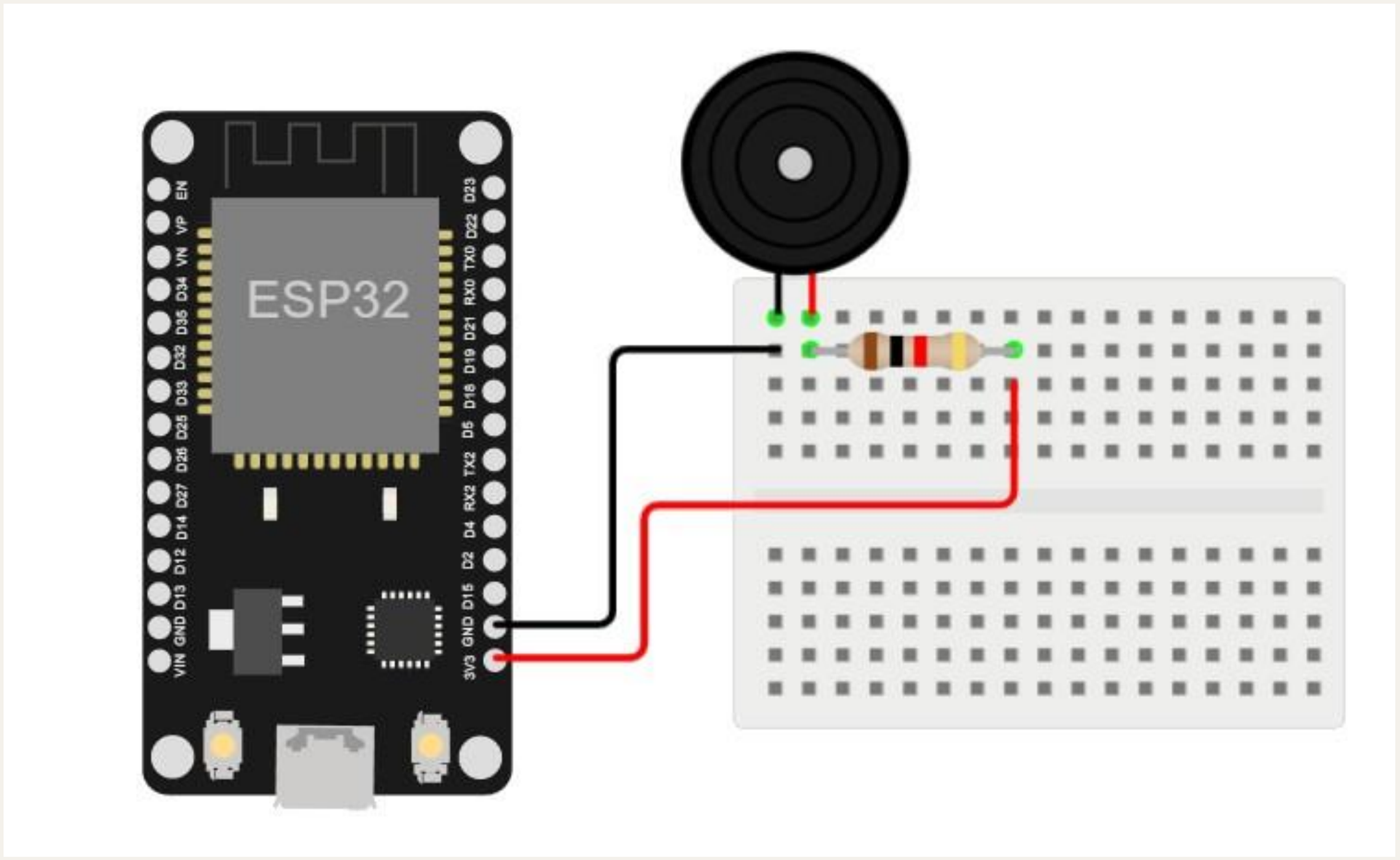
Agora iremos substituir o LDR por um potenciômetro, esse componente irá controlar a luminosidade do LED variando sua própria resistência através da rotação de sua chave.



É necessário colocar o fio GND conectado no pino em que você prefira que tenha menos resistência, assim o LED brilhará mais nesse lado. É importante sempre colocar um resistor conectado ao LED, pois quando o potenciômetro estiver no valor mínimo de resistência, pode acabar queimando o LED.

5 - Ativando um buzzer

Nesta pratica iremos continuar usando a ESP32 como uma fonte de alimentação, mas iremos retirar todos os outros componentes com exceção do resistor. Iremos adicionar somente um buzzer, que é como um pequeno alto-falante, e faremos com que ele apite. Para isso, iremos conectar o fio 3.3V na haste indicada como positiva pelo buzzer, e logo em seguida conectaremos o fio GND na outra perna.



Se seu buzzer apitou, então parabéns! Aqui vai mais um desafio! Que tal tentar ativar o buzzer utilizando um botão?
Dica: observe o exemplo 2.

OBS: O Tinkercad pode acabar dando um pequeno erro em que o som não saia em forma de apito, e sim de estalos, mesmo que o circuito esteja correto.

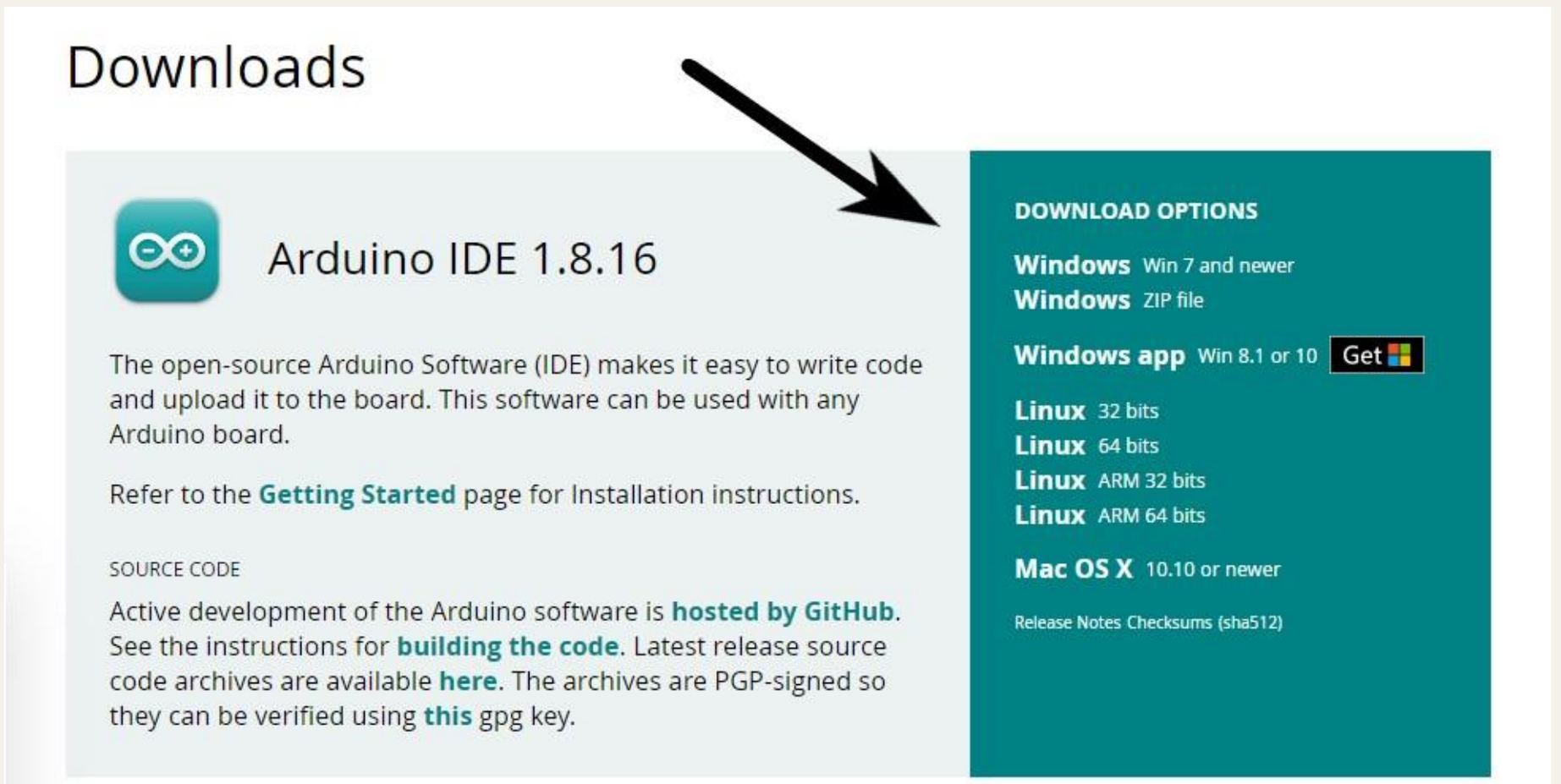
Programação básica

Agora que aprendemos alguns conceitos e praticas basicas, vamos seguir para a programação. A habilidade de programar é muito importante e com ela, você será mais lógico e terá uma maneira nova de pensar que irá lhe ajudar bastante em algumas situações na sua vida. Além disso, estamos aprendendo a usar o Arduino por aqui, e é a programação que vai definir qual tipo de tarefa especifica ele fará.

Antes de tudo, vamos entender o que é uma IDE:
Uma IDE é basicamente um ambiente de desenvolvimento integrado. É nela onde existe as principais ferramentas utilizadas para programação. Seja para programar uma placa (Arduino) ou para utilizar uma linguagem de programação, usaremos uma IDE ou um programa similar a uma.

1 - Instalando a Arduino IDE

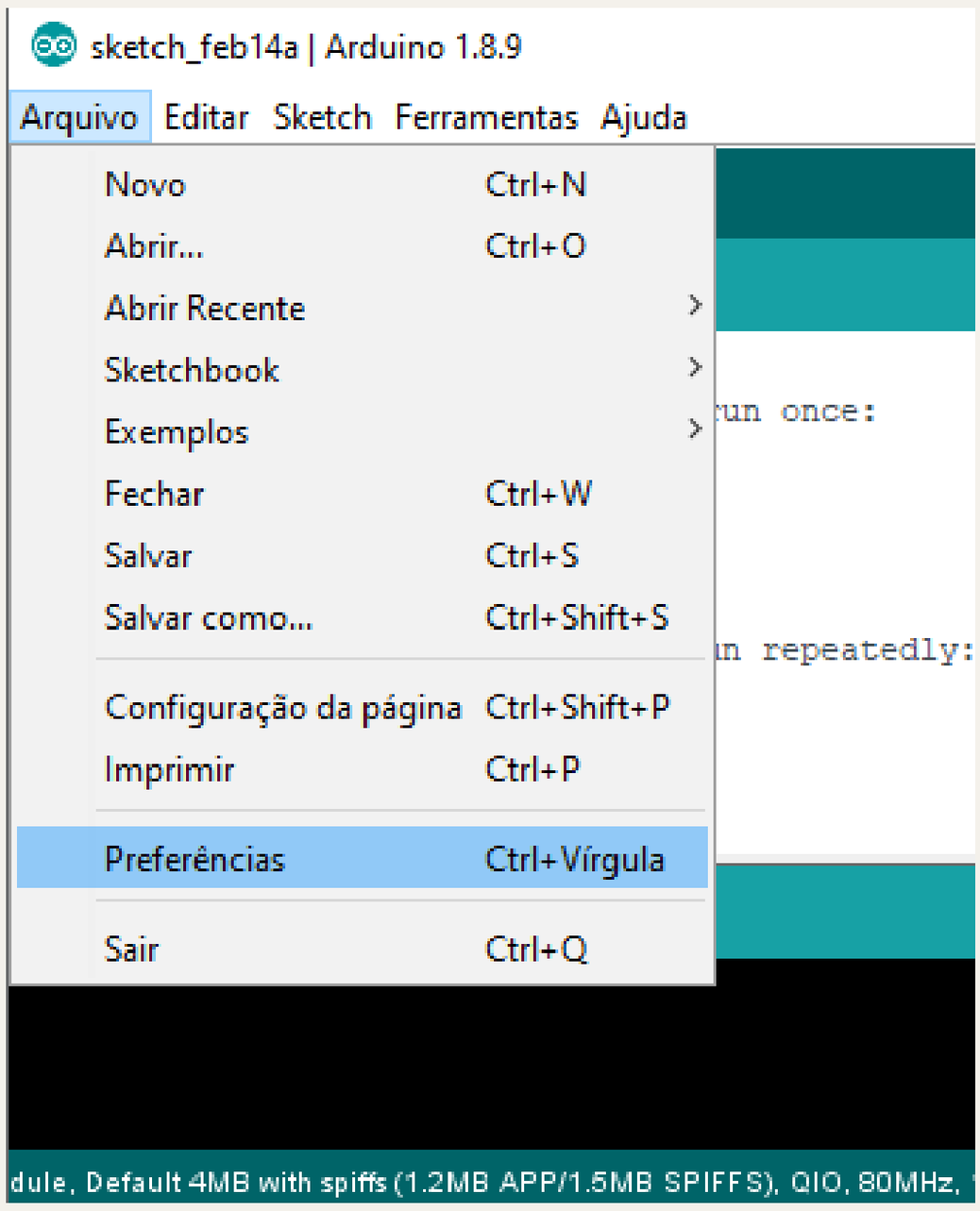
Primeiramente, será necessario acessar esse site:
https://www.arduino.cc/en/software
Logo em seguida selecionará a opção compativel com seu sistema operacional:

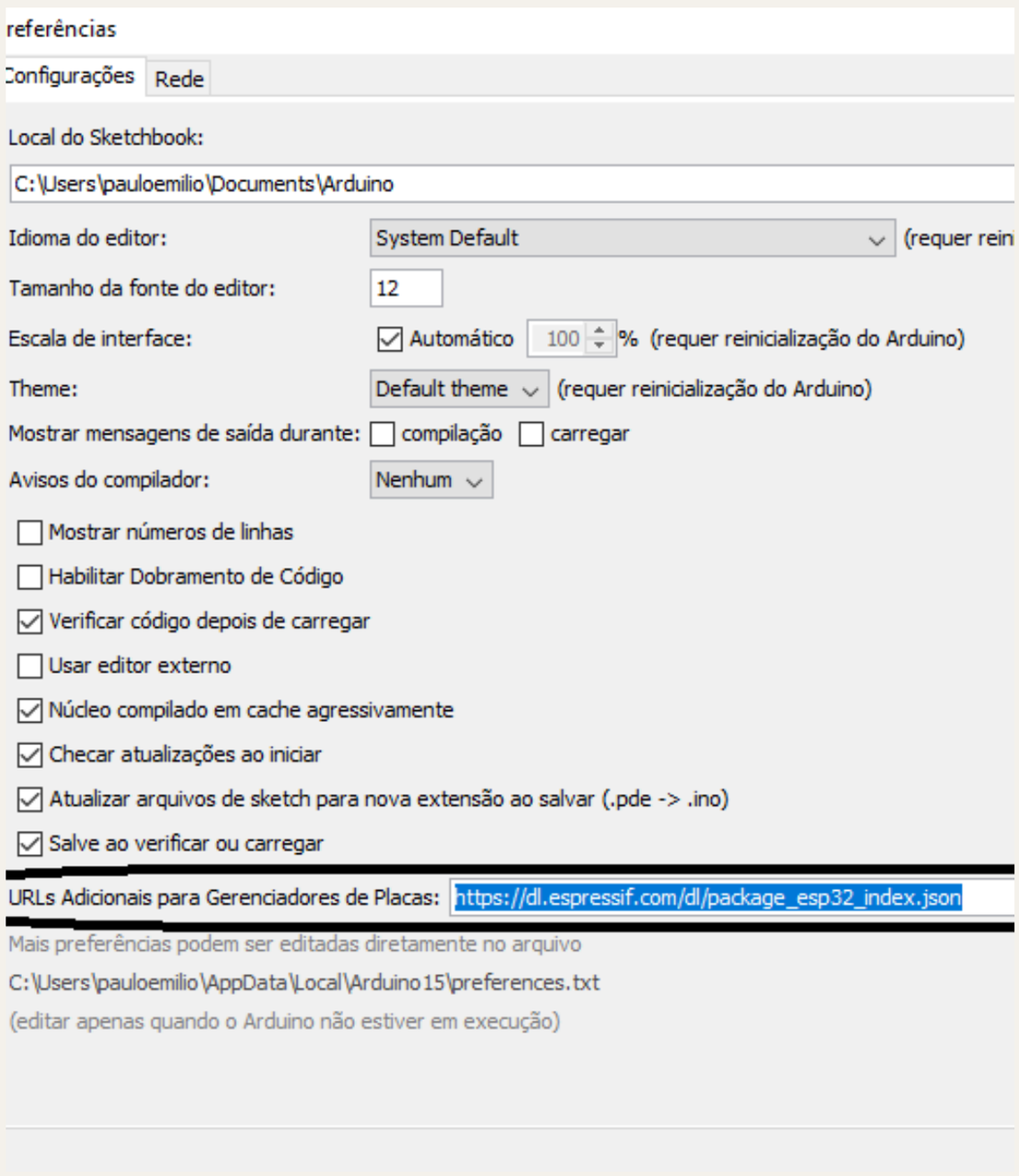


1.1 - Instalação da ESP32 na IDE

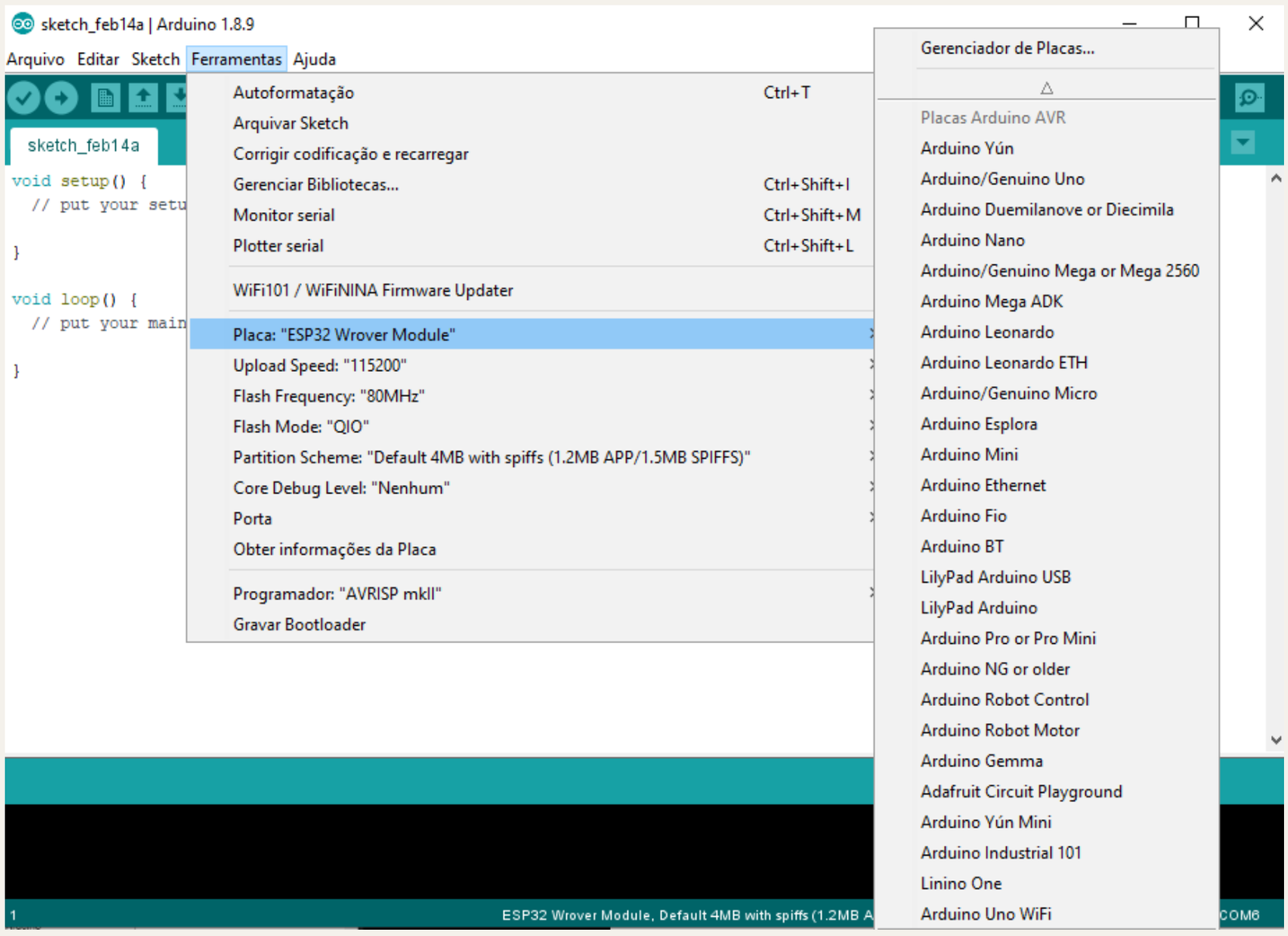
Agora que já instalamos a IDE do Arduino, precisamos ir para a próxima etapa que é adaptar a nossa IDE para a placa ESP32, que utilizaremos para fazer toda a programação e controle dos componentes eletrônicos escolhidos.

Primeiramente, é necessário abrir Arquivos e logo depois Preferências e colar na URL de gerenciador de placas a URL da placa ESP32:
"https://dl.espressif.com/dl/package_esp32_index.json"

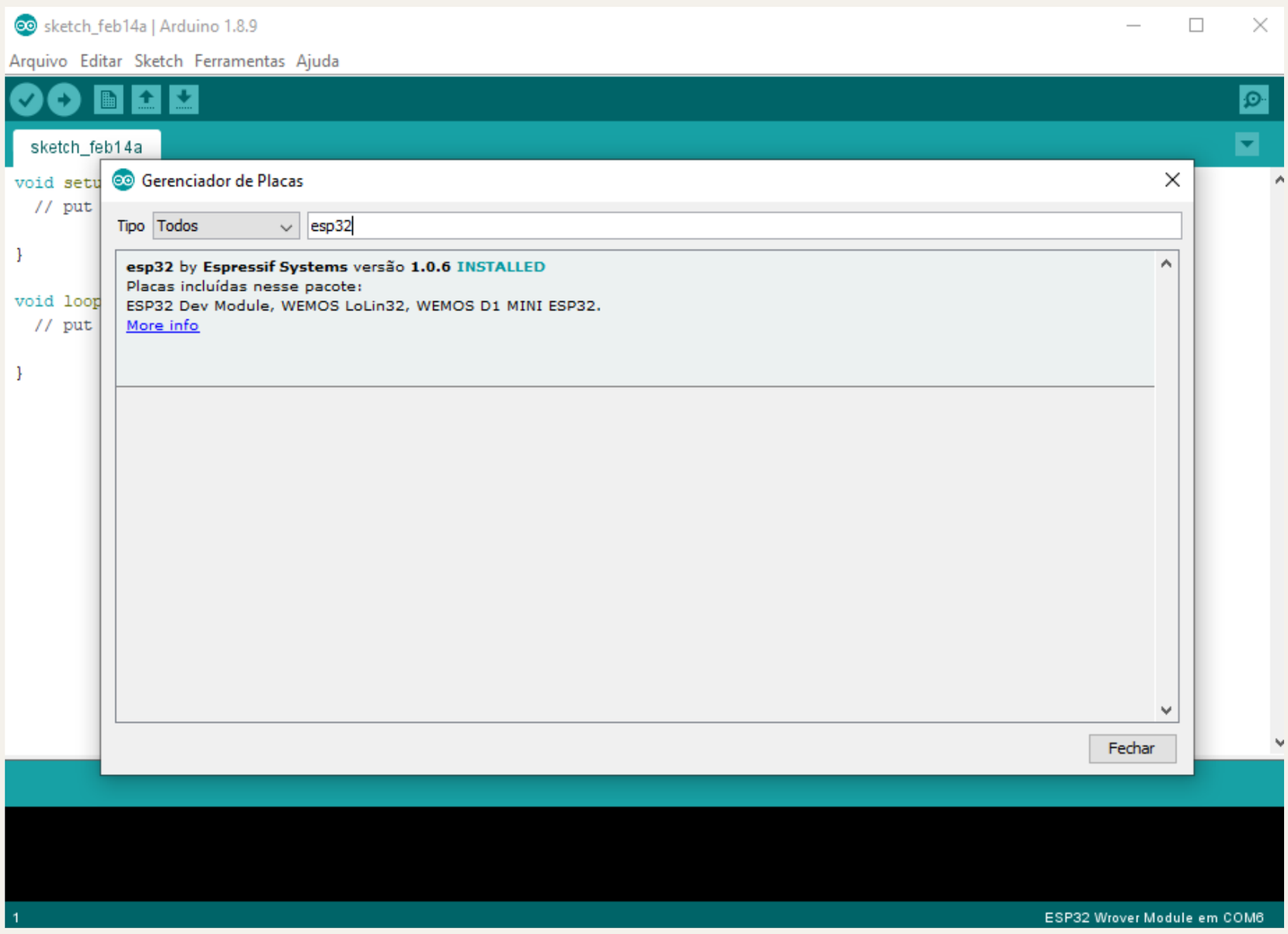




Logo após, deve-se instalar a placa ESP32 dentro do gerenciador de placas da IDE do Arduino, para fazer isso, entra-se em Ferramentas, passa-se o mouse por cima da placa que já está selecionada e clica-se em Gerenciador de Placas, como indica a imagem abaixo.



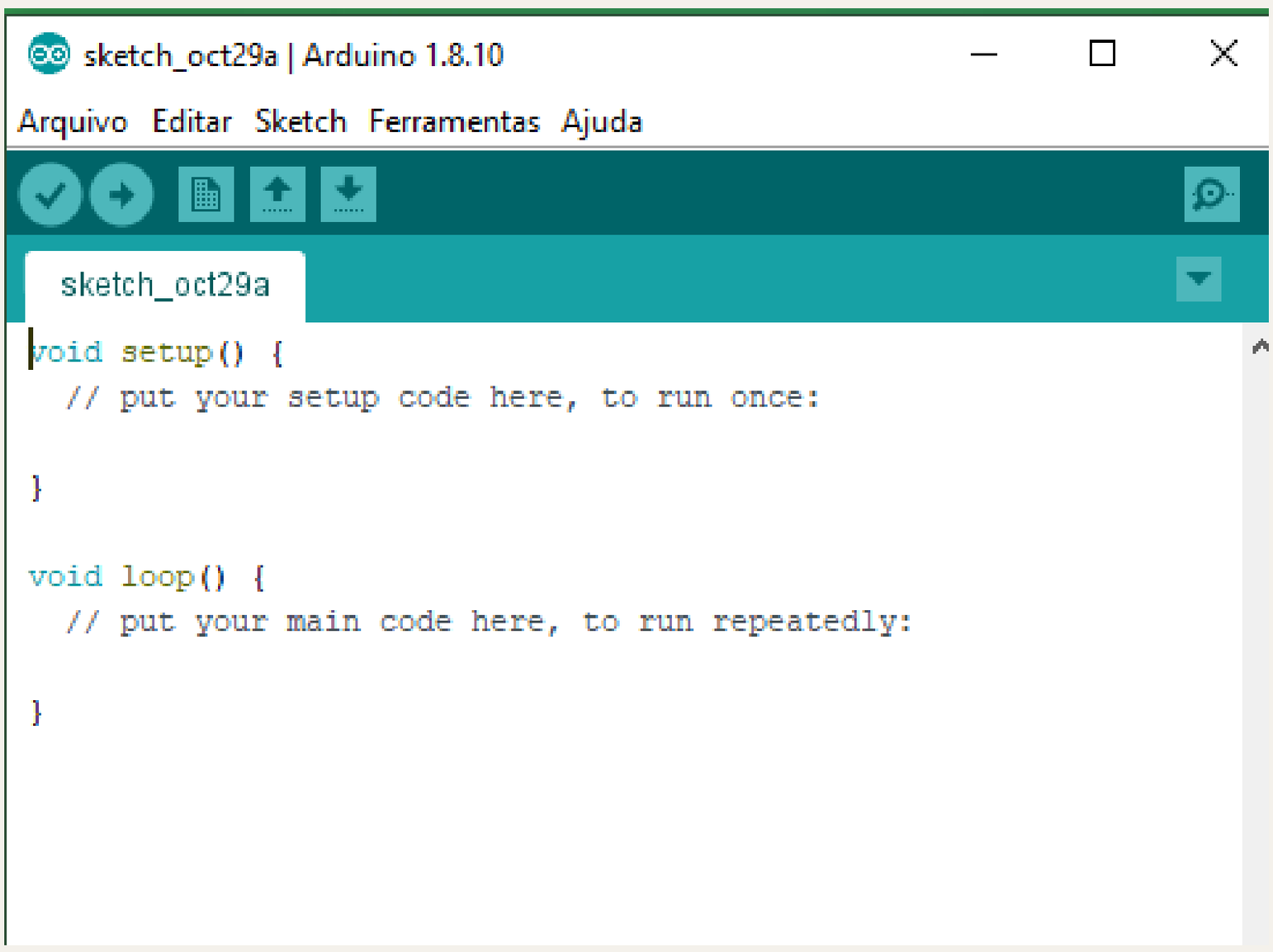
Com o gerenciador de placas aberta, procure pela ESP32 feita pela "Espressif Systems" na versão mais atualizada e a baixe.



Com a placa já instalada, o próximo passo é selecionar a "ESP32 Wrover Module" como placa selecionada para programação, selecionar a porta correta e você poderá começar a controlar a placa através da IDE do Arduino.

2 - Mãos á obra!

Agora que está tudo preparado, podemos ir à programação de fato. No mundo real, usamos a programação para controlar os computadores e fazer com que eles realizem tarefas que facilitem e melhorem nossas vidas. Assim que abrir o programa, essa será a primeira tela que verá:



Como provavelmente é seu primeiro contato com programação, vamos começar com o primeiro conceito basico sobre a linguagem de programação Arduino.

2.1- Void Setup e Void Loop

Esse é lugar onde voce irá digitar suas linhas de códigos! Primeiro iremos começar com o *Setup:* Nessa seção, iremos digitar as linhas de códigos que serão executadas somente uma vez no inicio do seu programa. Ou seja, é nele onde serão colocadas as configurações inicias para o Arduino. Como por exemplo, o código:

pinMode (porta escolhida no Arduino, OUTPUT ou INPUT);

Esse código serve para que você defina uma porta da sua escolha no Arduino como uma saída de informação (OUTPUT), como os LEDs e os Buzzers, ou como uma entrada informação (INPUT), como o botão e o potenciômetro. Exemplo: pinMode(12, OUTPUT);

Exemplo na prática:

```
void setup() {

pinMode (12, OUTPUT); // Aqui estamos determinando a porta 12 como uma saida de energ
}
```

Já o *Loop* será executado logo após os códigos do Void Setup. É a seção onde o código digitado será executado infinitamente (Loop) e só irá parar quando for ordenado.

```
void loop()
{
 digitalWrite(12,HIGH);
 delay(1000);
 digitalWrite(12,LOW);
 delay(1000);

}
```

Esse é um código que será explicado mais a frente. Assim que for executado e chegar na ultima linha de código, ele irá "resetar" e executar a primeira linha de codigo novamente.

3 - Variáveis

As variáveis são um dos elementos mais básicos na programação. Elas servem para guardar e representar um determinado valor ou expressão dentro do programa. É através dela que pegamos um espaço da memória do seu computador para armazenar uma determinada informação. Imagine como se estivéssemos dando um valor para o X de uma equação, assim X pode ser o valor que você escolher, como por exemplo: X = 12, a partir de agora toda vez que você escrever "X" no programa, ele irá interpretar como um 12.

```
LED = 10 // A partir de agora, toda vez que eu escrever "LED",
         // o programa interpretará como "10".


void setup()
{
  pinMode(LED, OUTPUT);
}

void loop()
{
  digitalWrite(LED, HIGH);
  delay(1000);
  digitalWrite(LED, LOW);
  delay(1000);
}
```

É importante ressaltar que antes de criar uma variável, É necessário determinar o tipo dela. Os tipos de variáveis determinam quais informações serão armazenadas. Como visto na imagem abaixo, esses serão os principais tipos de variáveis que utilizaremos.

Tipo	Tamanho	Valores Possíveis
bool	1 byte	true e false
byte	1 byte	0 a 255
sbyte	1 byte	-128 a 127
short	2 bytes	-32768 a 32767
ushort	2 bytes	0 a 65535
int	4 bytes	-2147483648 a 2147483647
uint	4 bytes	0 to 4294967295
long	8 bytes	–9223372036854775808L to 9223372036854775807L
ulong	8 bytes	0 a 18446744073709551615
float	4 bytes	Números até 10 elevado a 38. Exemplo: 10.0f, 12.5f
double	8 bytes	Números até 10 elevado a 308. Exemplo: 10.0, 12.33
decimal	16 bytes	números com até 28 casas decimais. Exemplo 10.991m, 33.333m
char	2 bytes	Caracteres delimitados por aspas simples. Exemplo: 'a', 'ç', 'o'

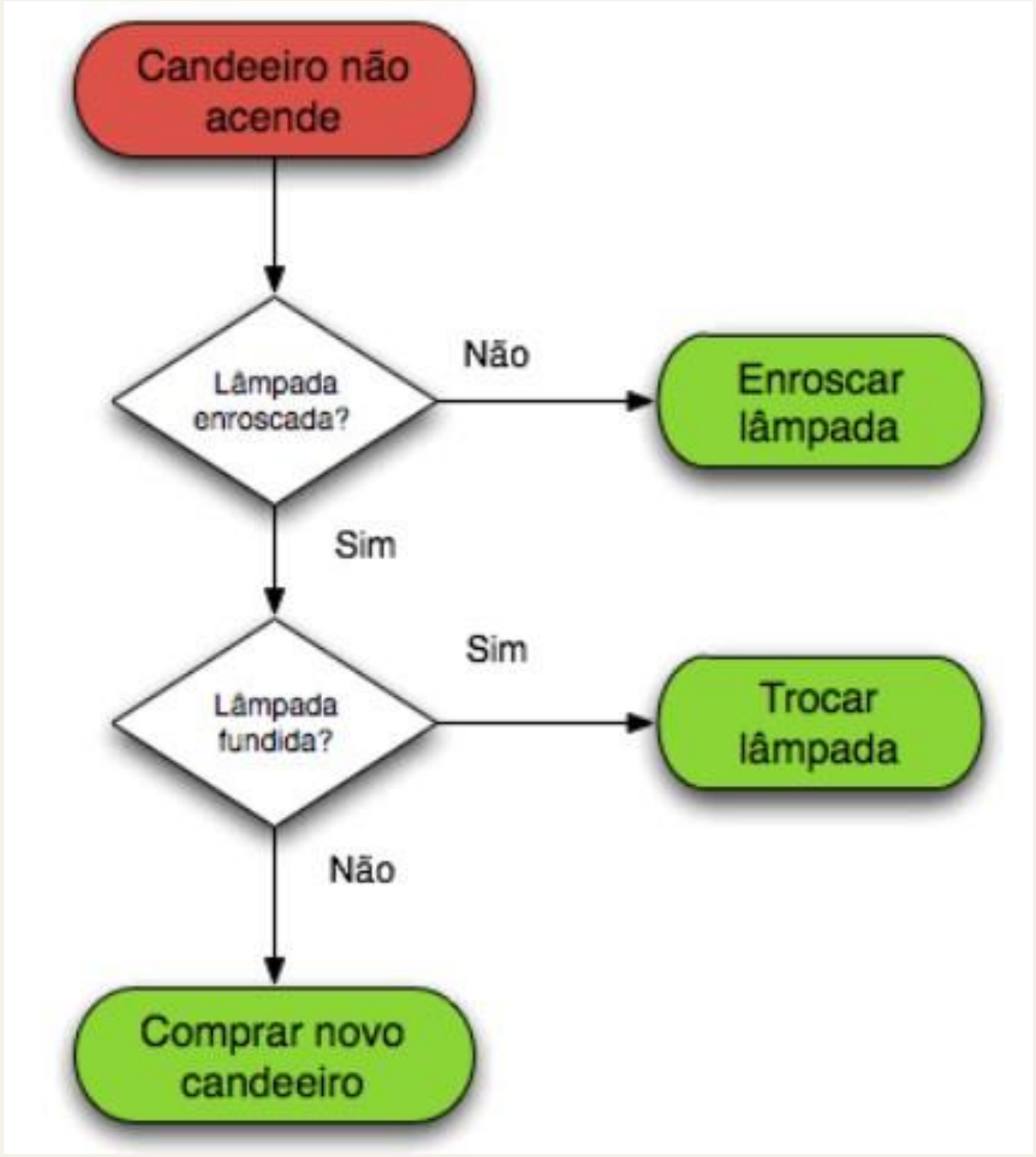
```
int LED = 10 // Após colocar "int" antes de "LED", só poderei
             // utilizar números inteiros, como mostra na imagem acima.
```

Vale ressaltar que as variáveis precisam ser criadas num espaço a parte, acima/antes dos lugares onde iremos digitar nossos códigos (Void setup e Void Loop). Para entender melhor, veja imagem abaixo.

```
int idade = 18
float numero = 18,0
char nome = Alenxadre
boolean nada = true
```

4 - Condicionais

Outro conceito básico presente na programação são as Condicionais. Este consiste em criar uma condição que determinará o fluxo que o programa irá seguir. Sendo assim, quando uma condição imposta no programa for atingida, uma série de códigos serão executados para agir de acordo.



Considere as condições propostas na imagem acima. Caso o candeeiro não acenda, o programa irá verificar se a lâmpada estará enroscada e caso não esteja, alguns códigos serão executados para a lâmpada ser enroscada. Funcionará do mesmo jeito caso a lâmpada esteja enroscada, o programa irá analisar a lâmpada para confirmar se está fundida e caso esteja, uma série de códigos serão executados para trocar a lâmpada. Basicamente é como se fosse o "se" e o "senão" da língua portuguesa. Exemplo: Se o cachorro estiver triste, brinque com ele, senão, durma com ele.

4.1 - Códigos

Existem determinados códigos necessários para impor condições ao programa. Os principais quando tratamos de programação em Arduino são: if, else e "else if".

```
int A = 1; //Utilizei variáveis que aceitam só números inteiros, e as chamei de A e B.
int B = 2;

void setup()
{
  Serial.begin(9600); //Esse código serve para ativar um local chamado "serial"
                      //onde o programa irá escrever as frases que escolherei abaixo.
}

void loop()
{
  if (A > B){          // Aqui coloquei a primeira condição, "Se A for maior que B".
    Serial.print("A é maior que B"); //Caso A seja maior que B, então ele escreverá isso.
  }
  else if (A < B){     //Adicionei mais uma condição, como já havia usado "if" então
                       //preciso usar o "else if", mas tem literalmente o mesmo significado.
  Serial.print("A é menor que B"); //
  }
  else {    //Caso nenhuma das condições anteriores seja verdadeira, o programa usará o "else".
    Serial.print("A é igual a B");
}
}
// Como resultado, é esperado que o programa esreva que B é maior que A, já que B = 2 e A = 1.
```

Como mostrado no exemplo acima, caso A seja maior que B, o programa escreverá: "A é maior que B", caso A seja menor que B, o programa escreverá: "A é menor que B", e em ultimo caso, se A for igual a B, o programa escreverá: "A é igual a B".

5 - Laços de repetição

Como o próprio nome diz, laços de repetição são comandos que se repetem enquanto uma determinada condição seja atendida. Imagine que está com fome, você comerá uma quantidade X de comida enquanto não estiver satisfeito e só parará quando sua barriga estiver cheia ou a comida acabar.

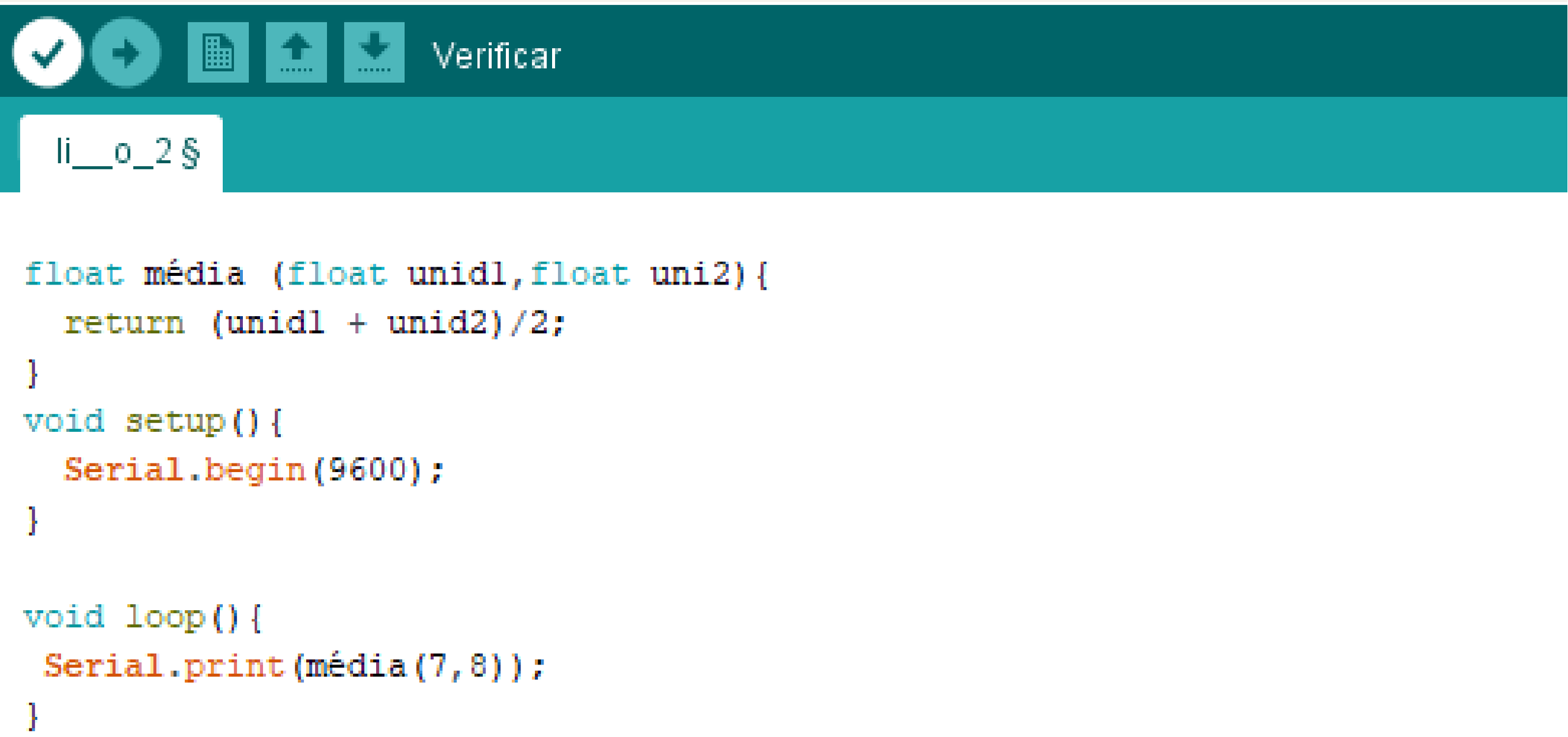
li__o_2 §

```
int A = 1;
int B = 10;
void setup()
{
  Serial.begin(9600);

}

void loop()
{
while (A<B){ //Essa linha de código implica que, enquanto A for menor que B, será somado +1 até que A se torne igual a B
  Serial.print(A);
  A++;
  delay(10000);
}
}
```

Um dos principais comandos para ativar o laço de repetição é o comando *while*. Ele implica que, enquanto aquela determinada condição estiver sendo atendida (A<B), alguma ação será executada (somar +1 no valor de A). Quando o limite for atingido (A=B), o programa parará.

6 - Funções

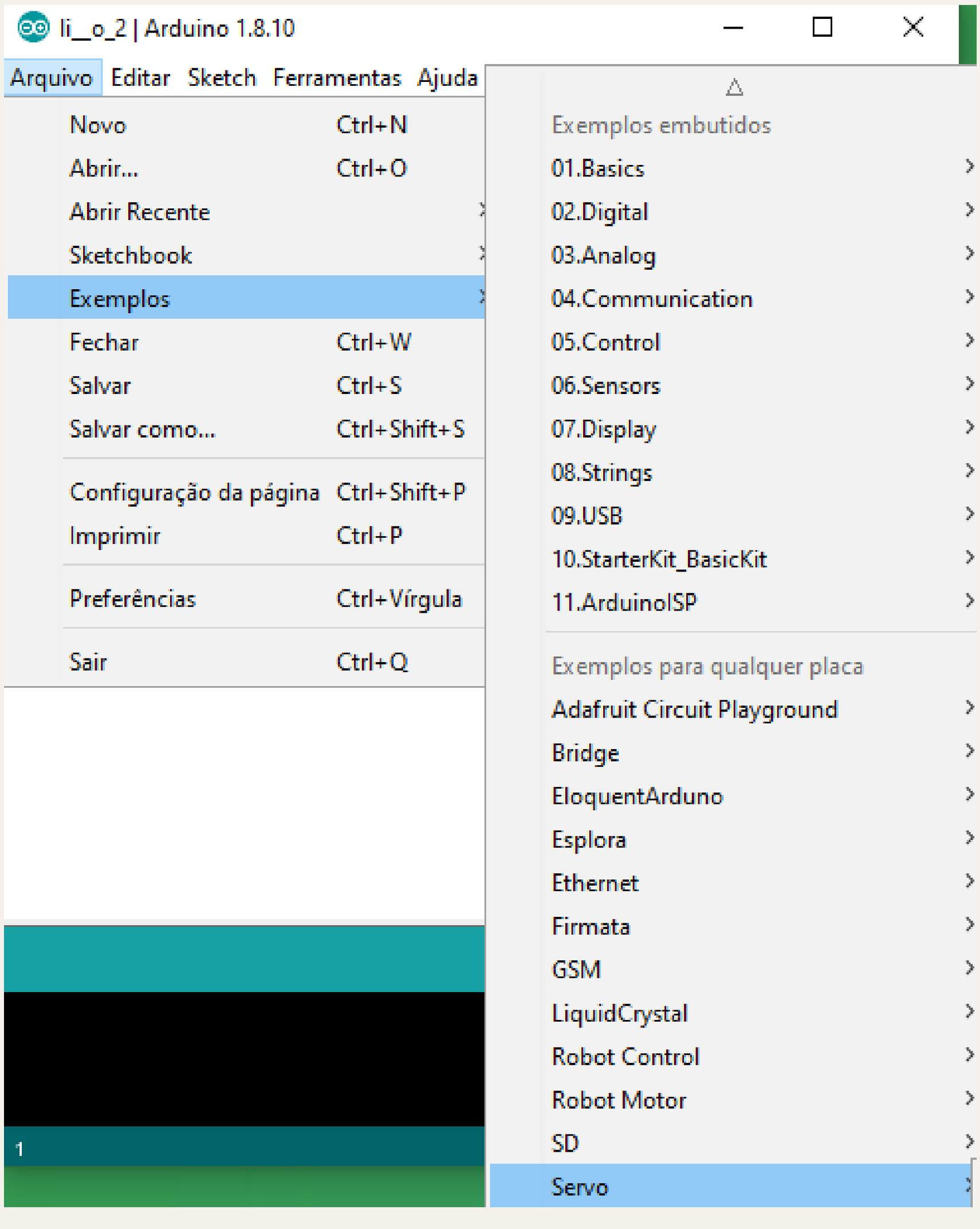
Esse conceito é bem parecido com o utilizado na matemática. No caso da programação, as funções são uma série de comandos utilizados para realizar uma determinada tarefa de acordo com os parâmetros definidos pelo programador.



Como mostrado no código acima, executamos uma função para calcular a nota de um aluno. Repare que utilizamos conceitos já mostrados anteriormente para montar esta função.

7 - Bibliotecas

No geral, as bibliotecas são nada mais do que subprogramas com códigos pré-estabelecidos para realizar diversas ações no seu programa. Contudo, no caso da programação com Arduino, utilizamos elas para ativar as funcionalidades de um ou mais componentes.



No exemplo acima, para acessar a biblioteca do programa Arduino, foi necessário clicarmos na aba arquivo e seguidamente em exemplos (ambos destacados em azul). Na aba aberta a direta, vemos as bibliotecas pré-instaladas e prontas para uso. Para um melhor esclarecimento do assunto, abriremos a biblioteca servo (destacada em azul).

Sweep

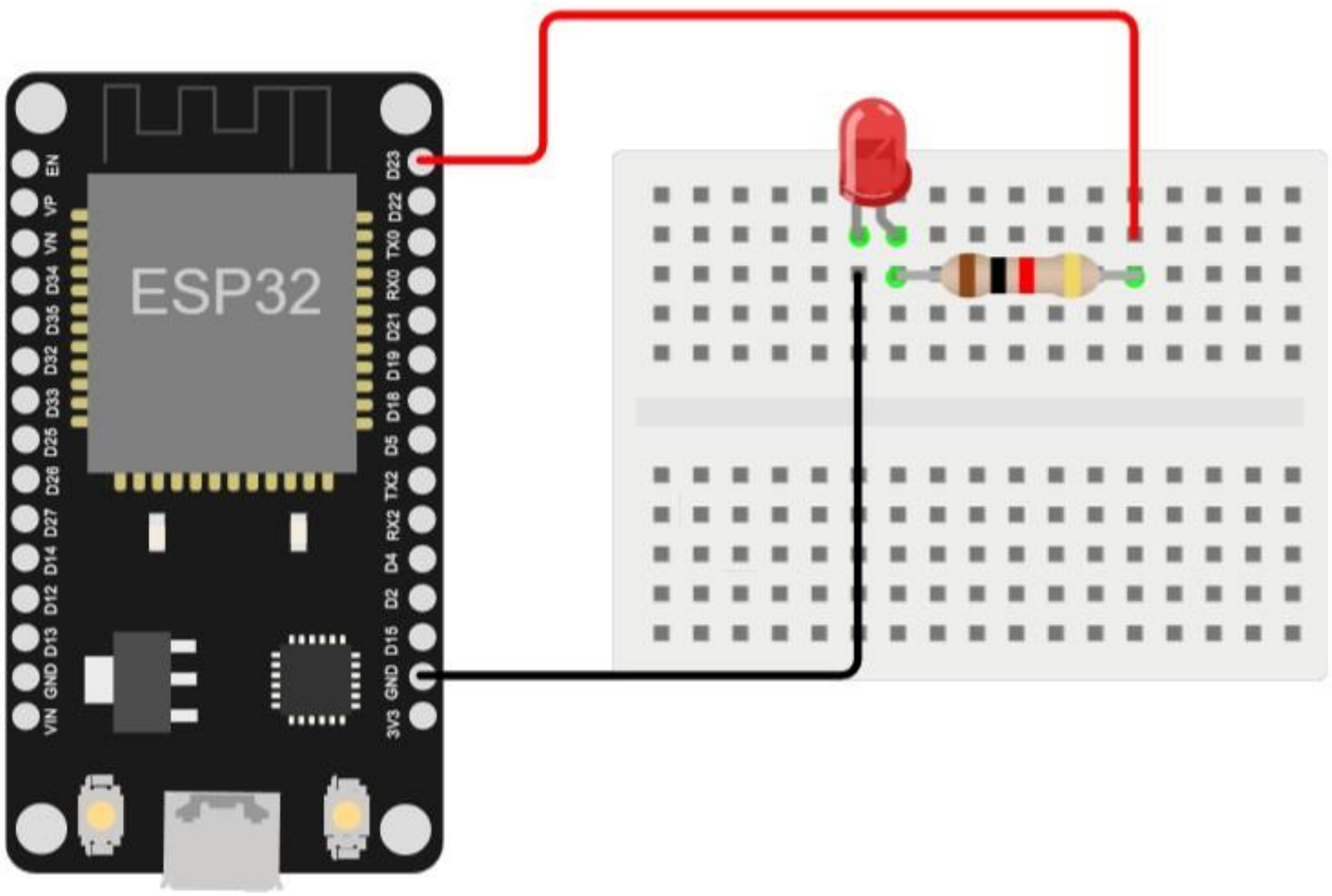
```
/* Sweep
 by BARRAGAN <http://barraganstudio.com>
 This example code is in the public domain.

 modified 8 Nov 2013
 by Scott Fitzgerald
 http://www.arduino.cc/en/Tutorial/Sweep
*/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15ms for the servo to reach the position
  }
}
```

Na imagem acima, vemos os códigos da biblioteca servo prontos para uso. Como dito anteriormente, essa biblioteca possui uma função específica que é, como o próprio nome sugere, controlar um componente chamado servo motor. Através de algumas alterações nessas linhas de código, podemos comandá-lo para realizar determinadas funções.

Praticas com programação

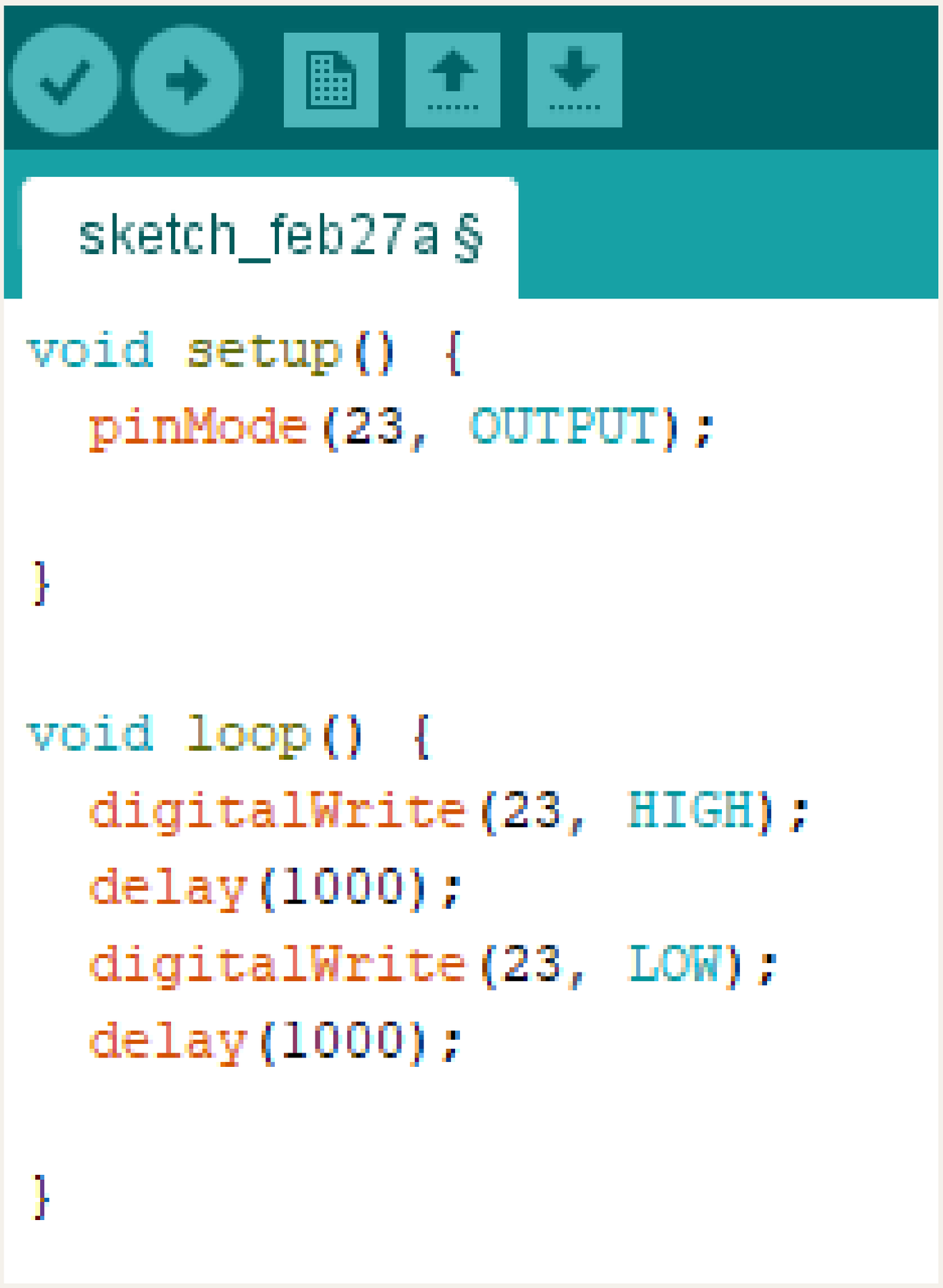
1 - Piscando um Led.

É nesta etapa que iremos pôr em prática todos os assuntos ensinados anteriormente. Para começarmos, iremos fazer uso da protoboard, um resistor, um LED e dos pinos digitais da ESP32. Primeiramente, escolheremos um dos pinos digitais, numerados de 0 a 35, para ser a nossa saída de energia (que no nosso caso será a porta D23) e a ligaremos na haste positiva do LED. Em seguida, utilizaremos uma porta GND e a conectaremos na haste negativa do LED, o resistor colocamos conectado em qualquer um dos lados do LED, por tanto que esteja no meio do caminho entre a ESP32 e o LED. Seguindo os passos acima, seu circuito ficará assim:



Feito isso, escreveremos nosso primeiro código! Antes de tudo, não se esqueça de abrir a IDE do Arduino, afinal, sem ela não teremos como executar a programação. Digitaremos as seguintes linhas de código:

Ps: Caso deseje fazer sozinho, não veja a imagem abaixo até que tenha terminado o código!

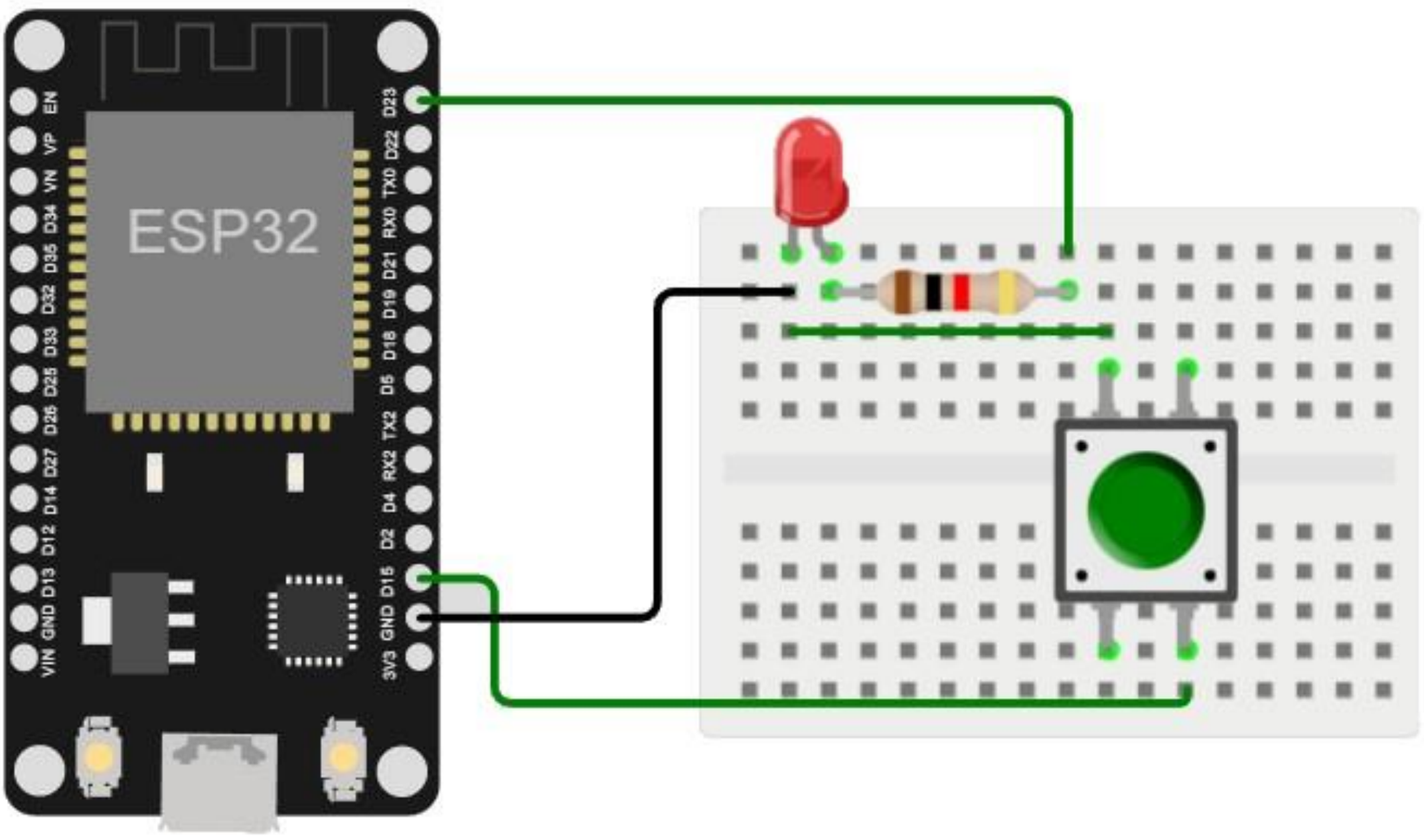


No primeiro passo (Void setup) usamos o comando *pinMode()* para declaramos a porta digital 23 como uma saída de energia (OUTPUT). Em seguida, na outra seção (void loop) utilizamos os comandos digitalWrite(), que determina se a porta 23 liberará energia de 3.3V (HIGH) ou uma energia de 0V (LOW). E o delay tem o proposito de pôr um intervalo de tempo entre cada ação do código, ou seja, ele adicionará o intervalo de 1000 milissegundos (1 segundo) para que o pino 23 libere energia e deixe de liberar.

Ps: Caso tenha feito tudo certo, o LED acenderá e apagará a cada 1 segundo.

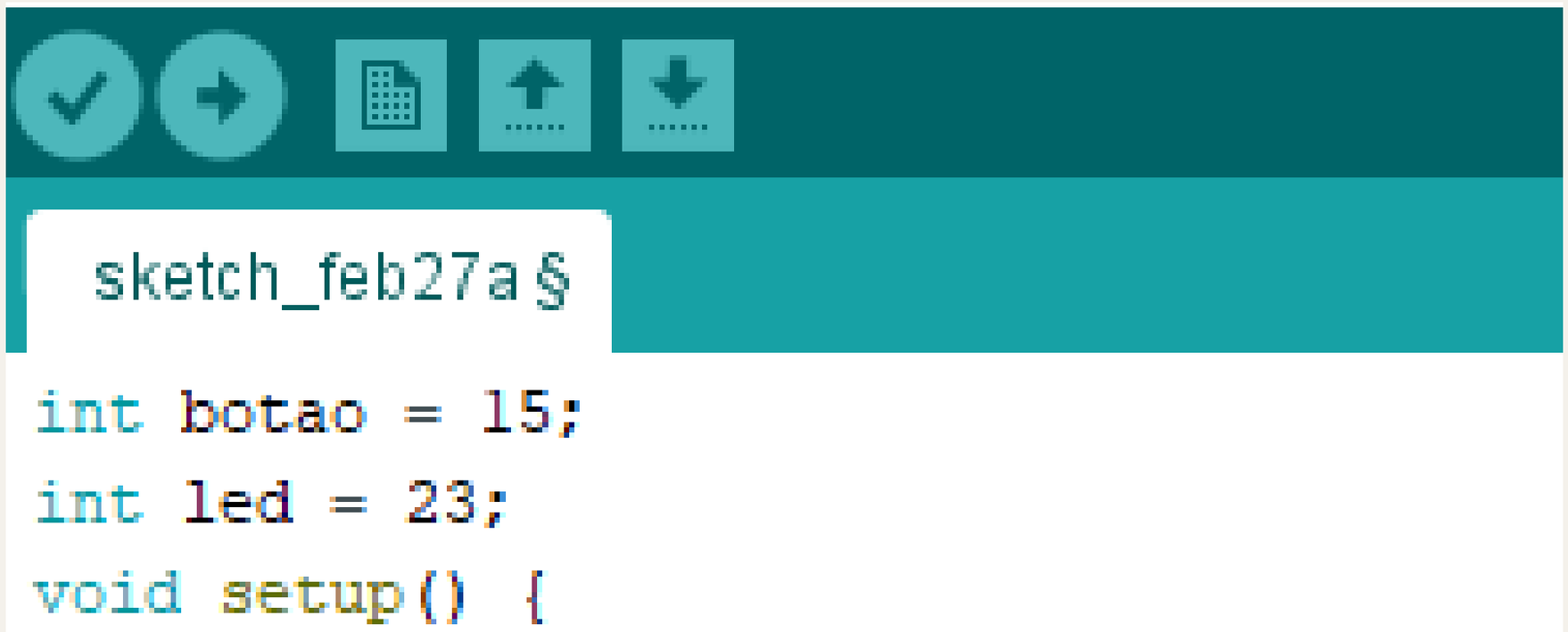
2 - Configurando um botão.

Está pratica é um pouco parecida com a ultima, porém iremos adicionar um botão e mais um pino. De forma similar a primeira vez que o usamos, ele agirá como um interruptor que pode parar imediatamente a passagem de energia, mas dessa vez, faremos um código que fará o botão executar essa tarefa "conscientemente".



Como mostrado acima, o circuito será montado da seguinte forma: A porta digital 15 será conectada ao botão, a porta digital 23 será conectada ao LED na haste positiva, o GND será ligado na haste negativa do LED e na outra porta do botão, o resistor vai estar em uma das portas do LED.

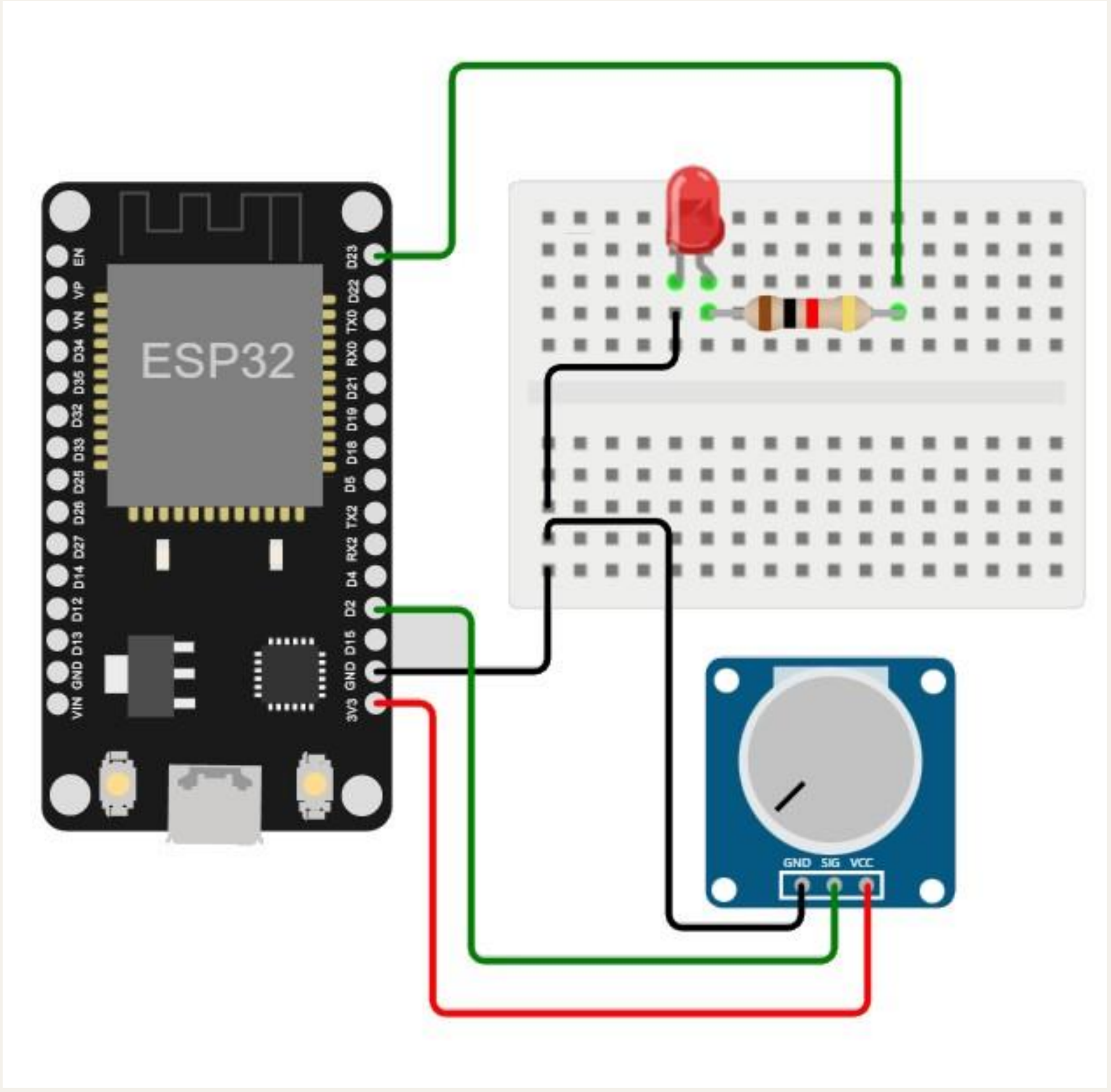
Agora chegamos a parte dos códigos para comandarmos a ESP32. É nesta etapa onde vamos mandar um comando para que a ESP32 pare de enviar energia quando o botão for pressionado.



Detalhando melhor os passos seguidos acima, começaremos criando duas variáveis, uma representando a porta digital do LED (led) e outra do botão (botao). Em seguida, declaramos o pino do LED como uma saída de energia e faremos a ESP32 reconhecer a variável "botao" como uma entrada de energia, já que é um botão. Já no V*oid Loop*, estamos ordenando que a ESP32 leia o estado em que a variável "botao" (botão) estará (pressionado ou não pressionado, 0 ou 1, 0V ou 3.3V). Dessa forma, caso o botão esteja sendo pressionado, a ESP32 irá ler essa alteração e parará imediatamente o envio de energia (no caso desta programação).

3 - Configurando um potenciômetro

Este exemplo é bem parecido com a nossa última prática com potenciômetro. Contudo, assim como a anterior, faremos a ESP32 trabalhar com o potenciômetro de forma "consciente".



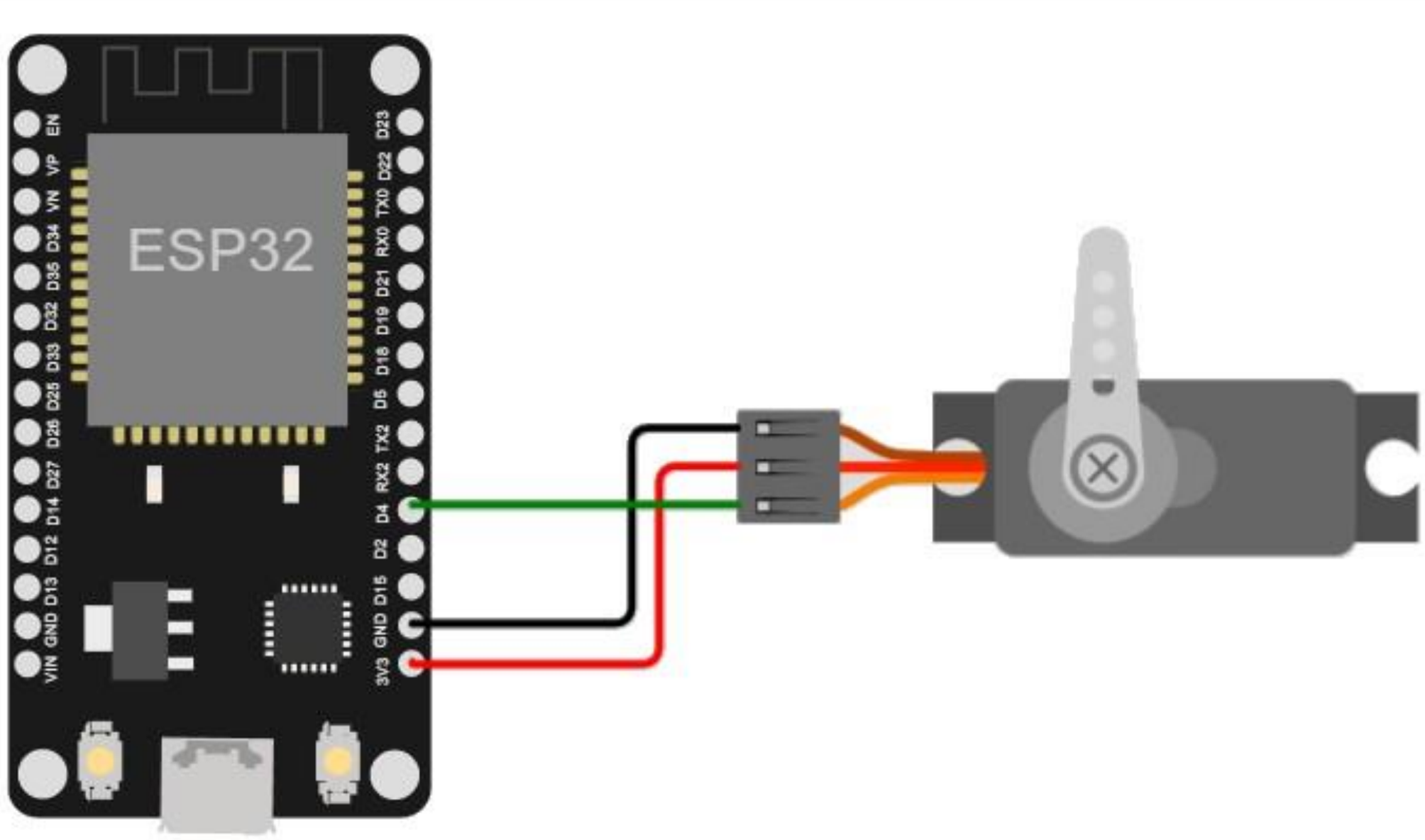
Basicamente esse é o circuito que montaremos. Seguindo alguns passos já citados anteriormente, conectaremos uma porta digital à haste positiva do LED, o GND na haste negativa e o resistor em qualquer um desses caminhos. Quanto ao potenciômetro, conectaremos o pino do meio da porta digital D2, e as outras duas extremidades (de sua preferência) ao GND e 3.3V.

Agora partiremos para o código. Faremos com que o potenciômetro controle a intensidade do brilho do LED, ou melhor, a resistência que ele proporcionará para o circuito.

Primeiramente, criaremos as variáveis para as entradas do LED (23) e do potenciômetro (2). Depois, uma variável representando a leitura e outro para a intensidade do LED. No Void Setup, definiremos a porta do LED como saída de energia. Já no Void Loop, faremos com que a variável leitura analise a porta analógica na qual o potenciômetro está ligado e ordenaremos que a variável "brilholed" faça um mapeamento, relacionando os ângulos do potenciômetro com o limite de tensão da ESP32 através de uma regra de três. Em seguida, mandaremos o LED entrar no estado "brilholed", que vai ser o valor em que o potenciômetro se encontra no mapeamento feito anteriormente.

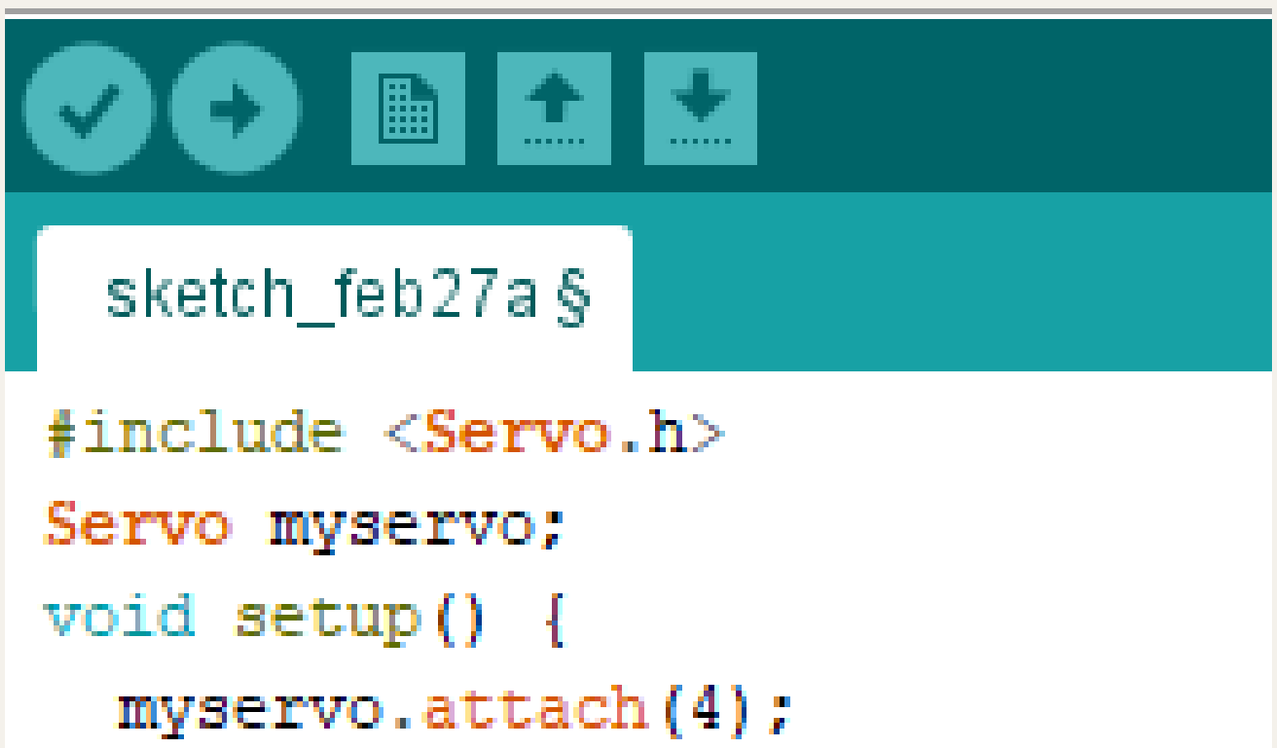
4 - Configurando um Servo motor

Chegamos na penúltima prática e também uma das mais "complexas" dentre elas. Iremos estrear o servo motor, como visto anteriormente, os servo motores são responsáveis por oferecerem movimento. No entanto, eles são utilizados para atingir movimentos com angulações precisas e não apenas para realizar giros de 360°.



Dessa vez, não usaremos uma protoboard, então ligaremos diretamente ao Arduino. Repare bem na coloração dos jumpers (fios) que estão no motor, pois é através deles que você se guiará. O marrom será conectado ao GND, o vermelho ligado no 3.3V e o laranja numa porta digital de sua preferência, no exemplo usaremos o D4.

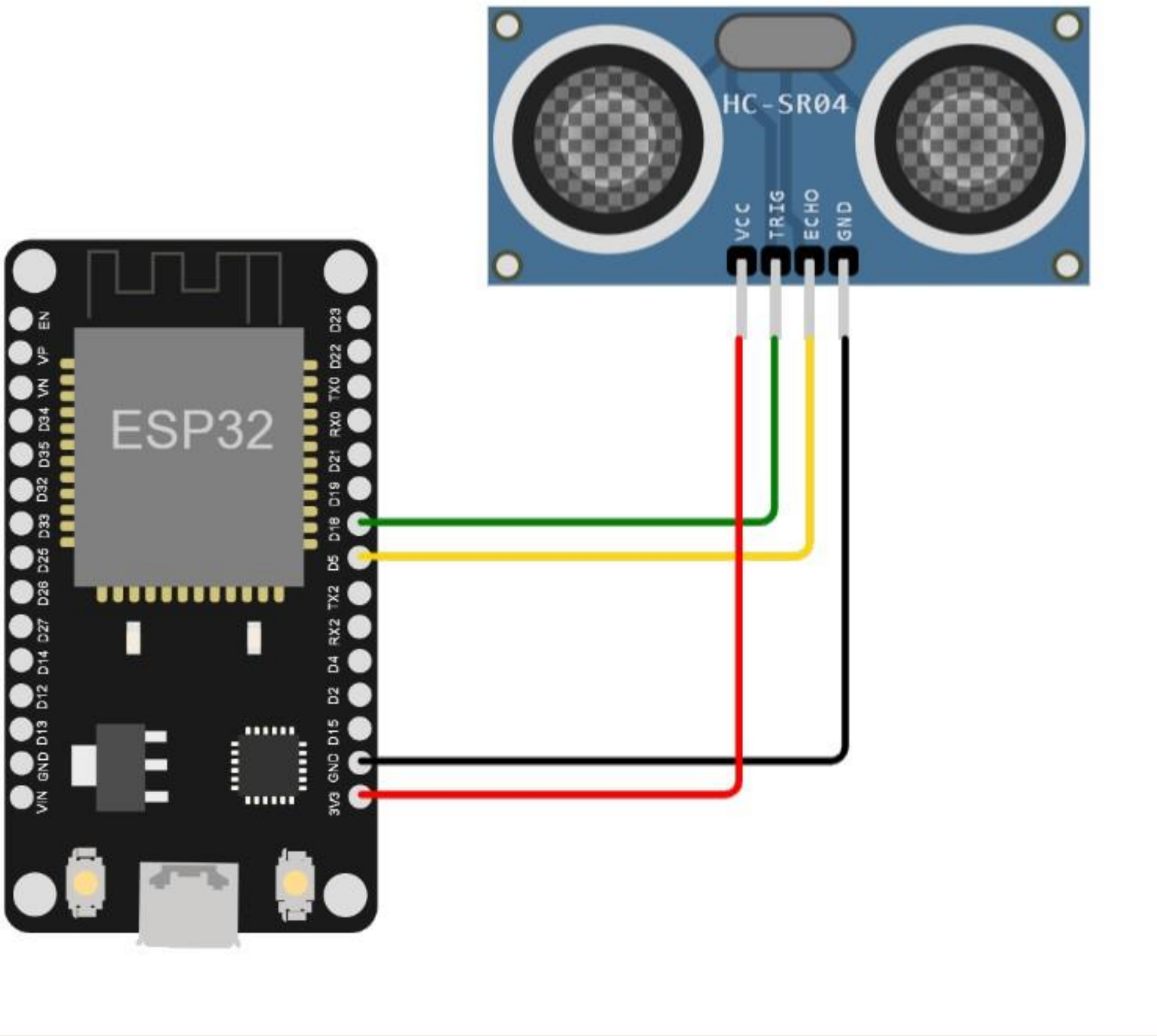
Agora sim digitaremos os comandos. Primeiro de tudo, usaremos uma função da IDE chamada "biblioteca". Como já ensinado, alguns componentes necessitam da instalação de uma biblioteca feita para o mesmo.



Aquela linha de código no topo indica que a biblioteca do servo já está inclusa. A partir daí, seguimos para a 2ª linha, que é onde nomeamos o nosso servo utilizando uma função da biblioteca incluída, nesse caso chamamos de "myservo".
No Void Setup nós definimos que a porta de controle do servo é a porta digital 4, que por observação é uma porta PWM, afinal o motor tem ângulos intermediários entre 1 e 180°.
No Void Loop colocamos as ações que o servo deve tomar, primeiro ele vai girar sua hélice para o ângulo 180°, depois vai aguardar um intervalo de 1000 milissegundos, em seguida vai para o ângulo de 50°, aguarda 1000 milissegundos e então o programa reinicia.

5 - Utilizando um sensor ultrassônico

Como já citado anteriormente, existem diversos tipos de sensores com determinadas funções, sejam elas medir temperatura ambiente, medir batimentos, etc. Dessa vez, usaremos um sensor ultrassônico capaz de medir curtas distancias através de ondas sonoras.



Os únicos componentes que utilizaremos serão a ESP32 e o sensor ultrassônico. Preste bastante atenção as portas nas quais o sensor está conectado, pois em alguns casos é possível queimá-lo simplesmente por ligar numa porta que não deveria. Na maioria dos casos, o 3.3V será representado como VCC e a sigla SIG será substituída por ECHO e TRIG, onde as mesmas serão ligadas as portas digitais.

Dessa vez, usaremos uma função chamada 'Serial' para podermos visualizar os resultados obtidos pelo sensor.

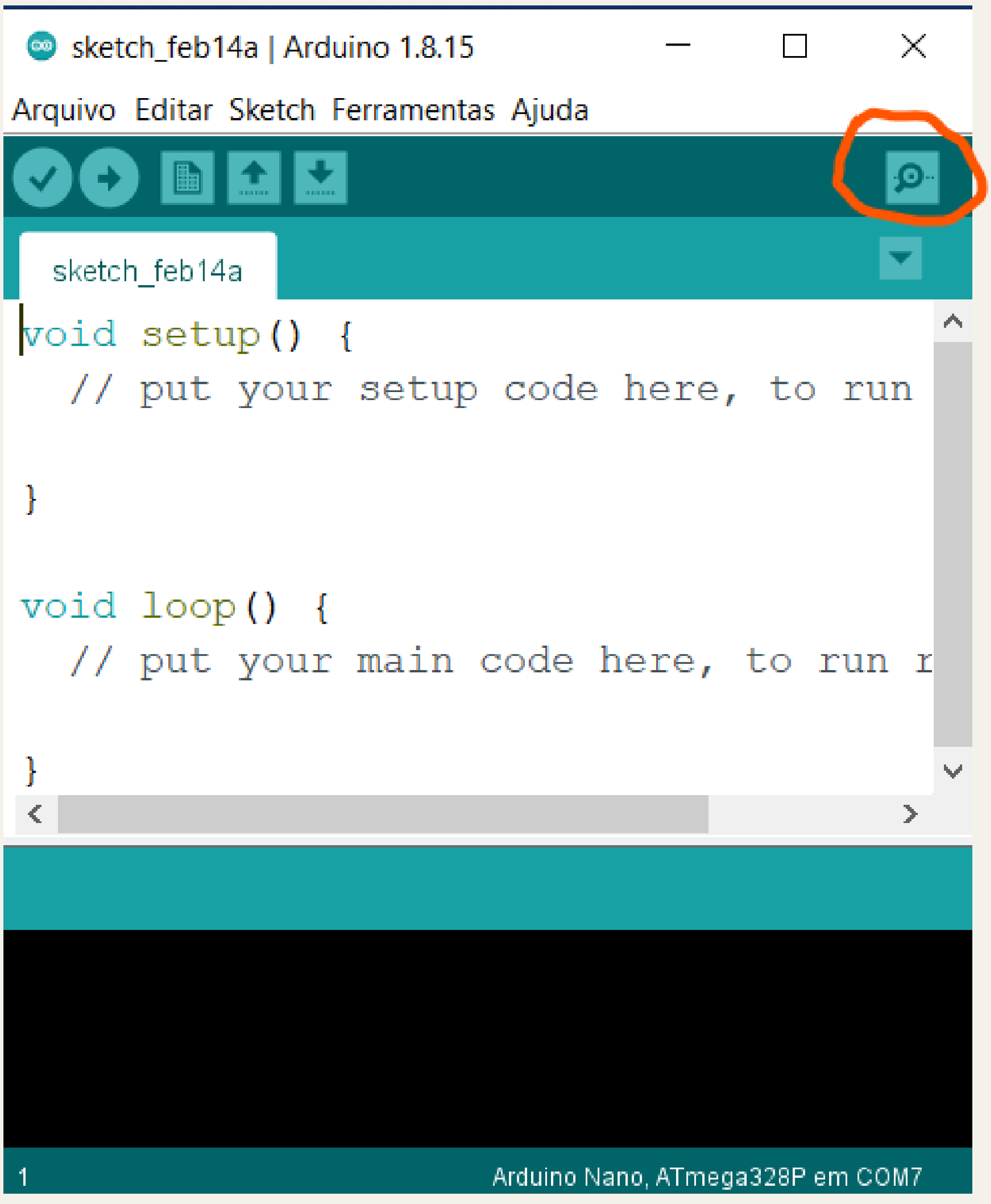
sketch_feb27a §

```
#include <Ultrasonic.h>

#define TRIGGER_PIN 18
#define ECHO_PIN 5

Ultrassonic ultrassonic(TRIGGER_PIN, ECHO_PIN);

void setup() {
  Serial.begin(9600);

}

void loop() {
  float cmMsec, inMsec;
  long microsec = ultrassonic.timing();

  cmMsec = ultrassonic.convert(microsec, Ultrassonic::CM);
  inMsec = ultrassonic.convert(microsec, Ultrassonic::IN);
  Serial.print("MS: ");
  Serial.print(microsec);
  Serial.print(", CM: ");
  Serial.print(cmMsec);
  Serial.print(", IN: ");
  delay(1000);

}
```

Como mostrado no exemplo anterior, importaremos a biblioteca do sensor ultrassônico, em seguida definiremos as portas nas quais os pinos ECHO e TRIGGER estarão conectados e declararemos o sensor utilizado. No void setup, somente ativaremos o Serial, que é usado como um monitor para visualizarmos dados obtidos ou utilizados pelos componentes.

Ao chegarmos no void loop, declararemos 2 variáveis, a primeira representará as unidades de medida: centímetros (cm) e polegadas (in), fazendo-as serem escritas com mais de uma casa decimal. Já a segunda representará o tempo utilizado, que no caso é a unidade segundos, fazendo com que um novo valor seja registrado a cada segundo. Em seguida, faremos com que as variáveis sejam representadas pelas siglas CM e IN. Por último, faremos com que os valores dos segundos passados, centímetros medidos e polegadas registradas sejam escritos no Serial a cada 1 segundo.

Para os estudantes em dúvida sobre como acessar o Serial, basta clicar no icone de lupa localizado no canto superior direito da tela.

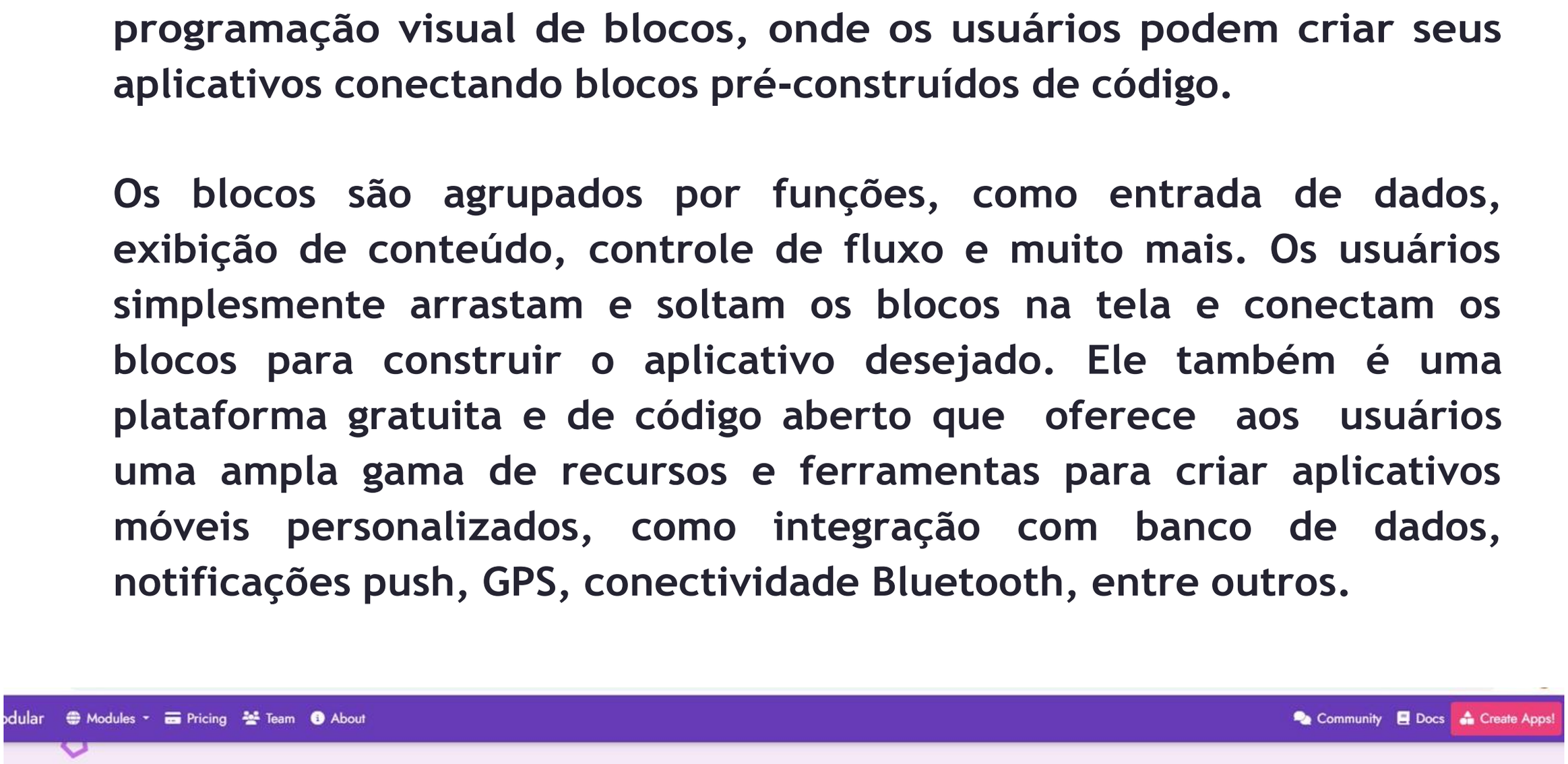


Kodular

1 - Introdução

Kodular é uma plataforma de desenvolvimento de aplicativos móveis baseada em nuvem que permite que usuários sem conhecimento de programação criem facilmente seus próprios aplicativos para Android. A plataforma é baseada na linguagem de programação visual de blocos, onde os usuários podem criar seus aplicativos conectando blocos pré-construídos de código.

Os blocos são agrupados por funções, como entrada de dados, exibição de conteúdo, controle de fluxo e muito mais. Os usuários simplesmente arrastam e soltam os blocos na tela e conectam os blocos para construir o aplicativo desejado. Ele também é uma plataforma gratuita e de código aberto que oferece aos usuários uma ampla gama de recursos e ferramentas para criar aplicativos móveis personalizados, como integração com banco de dados, notificações push, GPS, conectividade Bluetooth, entre outros.



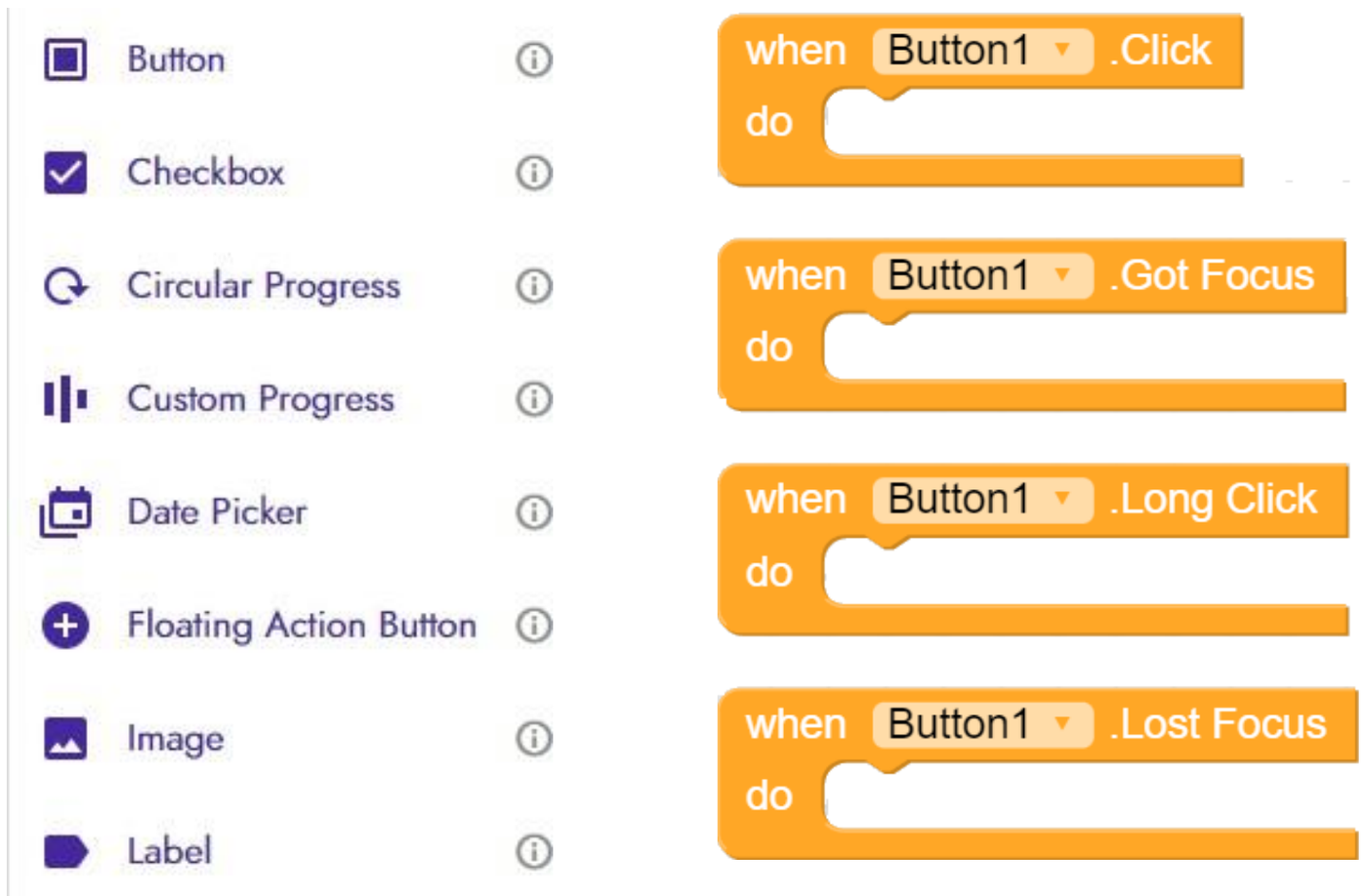
Página inicial do site, clique na imagem para acessa-lo.

2 - Principais Blocos

Kodular oferece uma ampla variedade de blocos que os usuários podem usar para construir aplicativos móveis personalizados. Aqui estão alguns dos principais blocos do Kodular:

2.1 - Bloco de Interface do Usuário

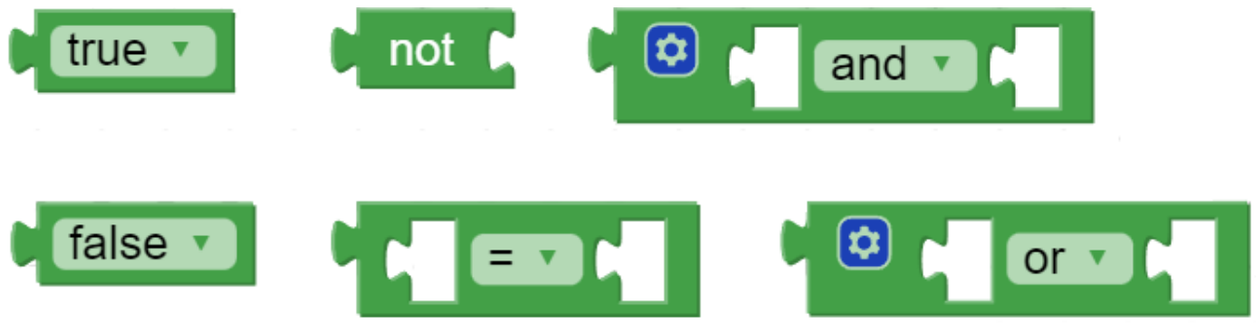
Esses blocos permitem que os usuários criem a interface do usuário de seus aplicativos. Eles incluem blocos para adicionar botões, caixas de texto, imagens, listas, menus e muito mais.



Como, nesse caso, temos alguns dos blocos que podem ser utilizados para um botão, com intuito de se comunicar com o usuário. O botão é somente um exemplo de bloco de interface do usuário, no Kodular, temos a disposição.

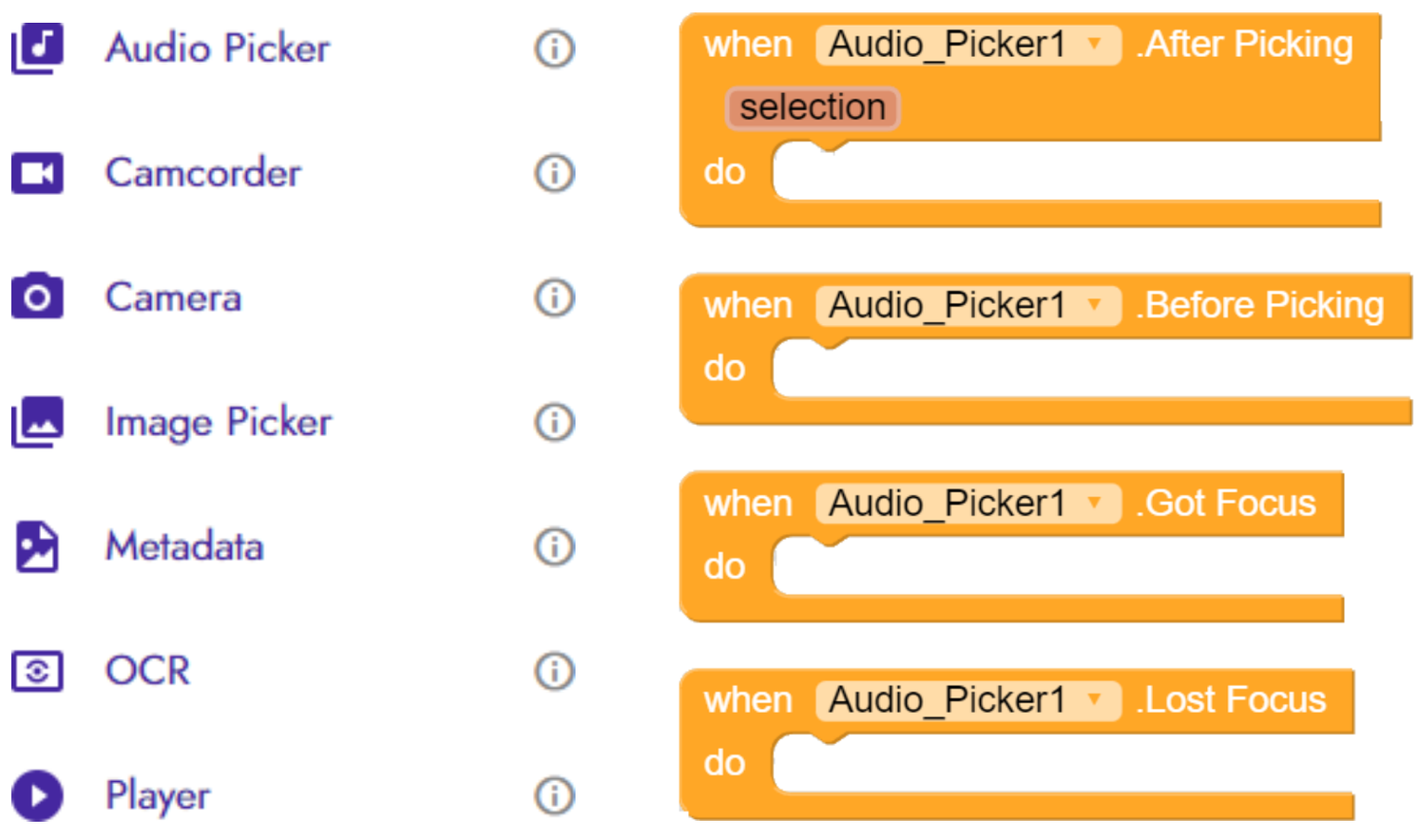
2.2 - Bloco de Lógica

Esses blocos permitem que os usuários controlem o fluxo de seus aplicativos. Eles incluem blocos para tomadas de decisão, loops, variáveis, funções matemáticas e muito mais.



2.3 - Bloco de Mídia

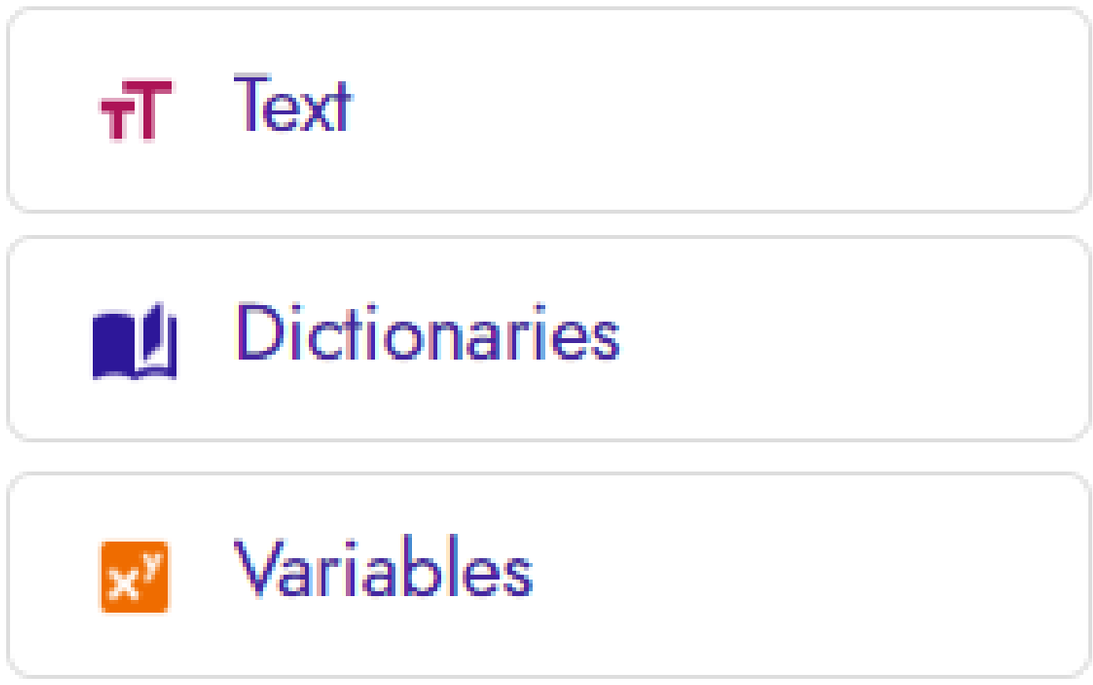
Esses blocos permitem que os usuários adicionem recursos de áudio e vídeo aos seus aplicativos. Eles incluem blocos para reproduzir música, gravar som, reproduzir vídeo e muito mais.



Como, nesse caso, temos alguns dos blocos que podem ser utilizados para um botão, com intuito de se comunicar com o usuário. O botão é somente um exemplo de bloco de interface do usuário, no Kodular, temos a disposição.

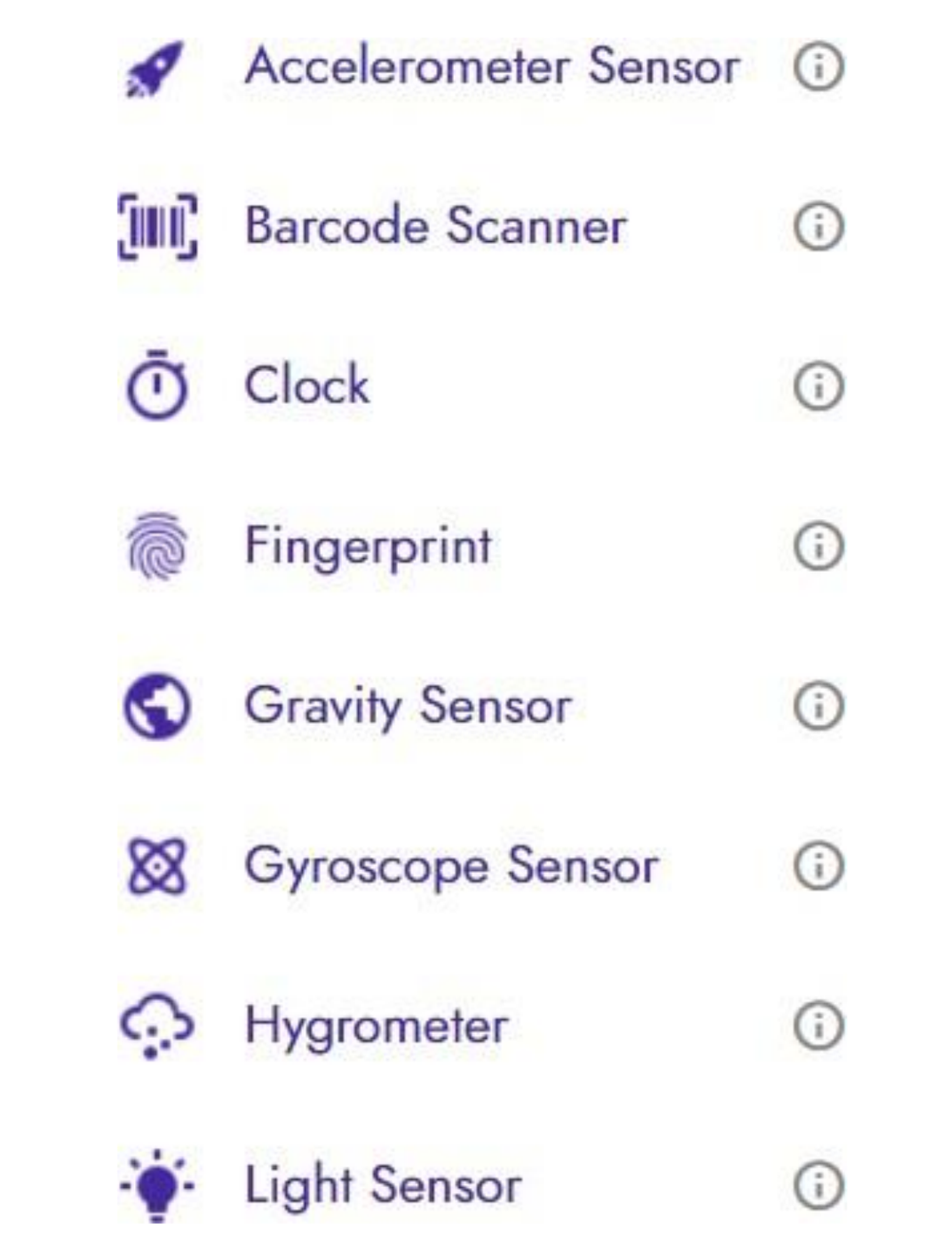
2.4 - Blocos de Dados

Esses blocos permitem que os usuários armazenem e gerenciem dados em seus aplicativos. Eles incluem blocos para salvar e carregar dados em um banco de dados, enviar e receber dados por meio de APIs da web e muito mais.



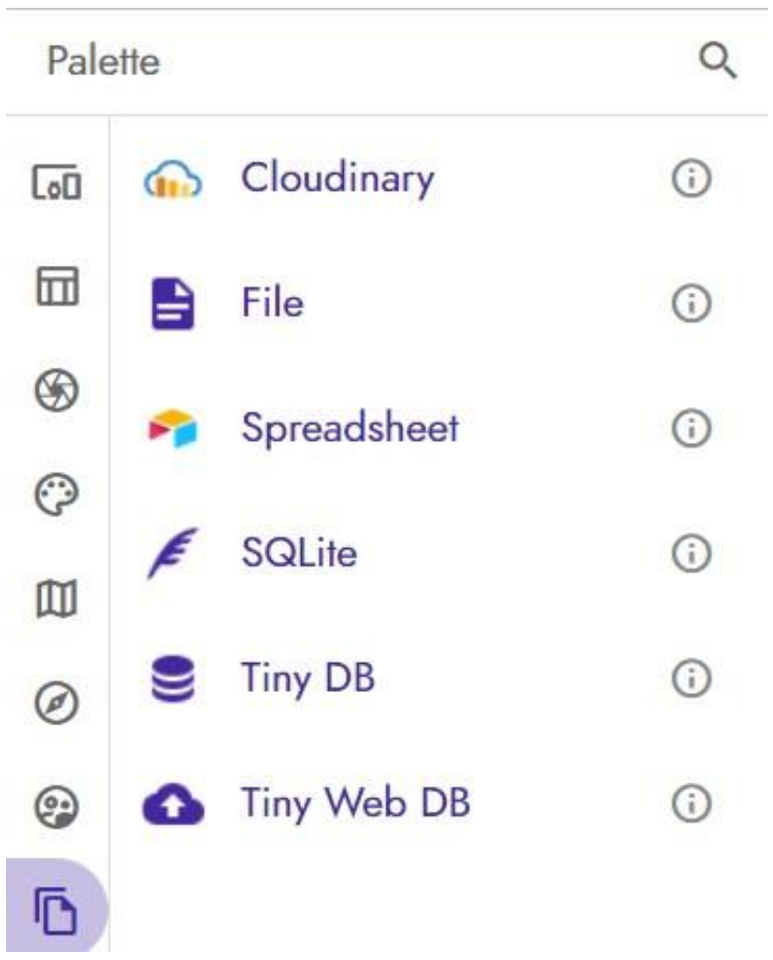
2.5 - Blocos de Sensores

Esses blocos permitem que os usuários acessem os sensores do dispositivo, como GPS, acelerômetro e giroscópio. Eles incluem blocos para obter a localização atual do dispositivo, detectar movimento e muito mais.



2.6 - Blocos de Rede

Esses blocos permitem que os usuários se conectem à Internet e se comuniquem com outros dispositivos. Eles incluem blocos para fazer solicitações HTTP, enviar e-mails, enviar mensagens de texto e muito mais.



2.6 - Blocos de Extensões

Esses blocos permitem que os usuários adicionem funcionalidades personalizadas aos seus aplicativos. Eles incluem blocos de extensões desenvolvidos pela comunidade Kodular, como blocos para reconhecimento de fala, reconhecimento de imagem e muito mais.

O passo a passo para importar blocos feitos pela comunidade:

1. **No Kodular Creator, você verá a tela "Designer". Aqui, você pode arrastar e soltar componentes na tela para criar a interface do usuário do seu aplicativo.**

2. **Para acessar o "Editor de Blocos", clique na aba "Blocos" no canto superior direito do Kodular Creator.**

3. **No "Editor de Blocos", você pode ver os blocos nativos disponíveis no Kodular e começar a criar a lógica do seu aplicativo.**

4. **Se você quiser adicionar uma extensão ao seu projeto, você pode fazer isso usando a funcionalidade "Importar Extensão" no "Editor de Blocos". Clique no ícone "Importar Extensão" (é o ícone com um bloco de quebra-cabeça verde e um sinal de "+" no canto superior direito do "Editor de Blocos").**

5. Na janela que se abre, você pode pesquisar e selecionar as extensões que deseja adicionar ao seu projeto. Lembre-se de verificar a fonte da extensão e verificar se ela é segura e confiável.

Depois de importar uma extensão, os novos blocos fornecidos pela extensão estarão disponíveis no "Editor de Blocos" e você poderá utilizá-los junto com os blocos nativos do Kodular para expandir as funcionalidades do seu aplicativo.

Práticas com Kodular

1 - Criação de conta

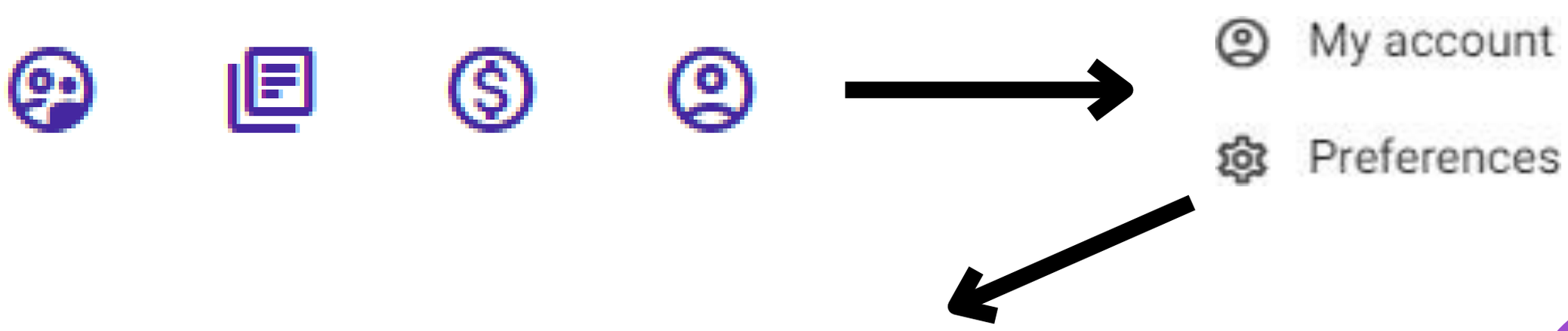
Agora vamos colocar em prática os conhecimentos que adquirimos nas explicações anteriores. Primeiramente, é necessário criar uma conta no site para que possamos criar nossos aplicativos livremente. Ao acessar o site, clique no botão "Create Apps" e você será direcionado para a tela de login. Existem várias formas de criar ou entrar em uma conta no site, mas é recomendável criar uma conta diretamente vinculada ao site, selecionando a opção "Create Account". Isso facilitará caso você queira criar aplicativos em conjunto, pois basta fornecer o usuário e a senha para acessar a conta. Não é necessário vinculá-la a um e-mail ou outras contas.

Além disso, é importante lembrar que ao criar uma conta no Kodular, você deve escolher um nome de usuário e senha seguros e fáceis de lembrar para evitar problemas.

1.1 - Alterando a linguagem

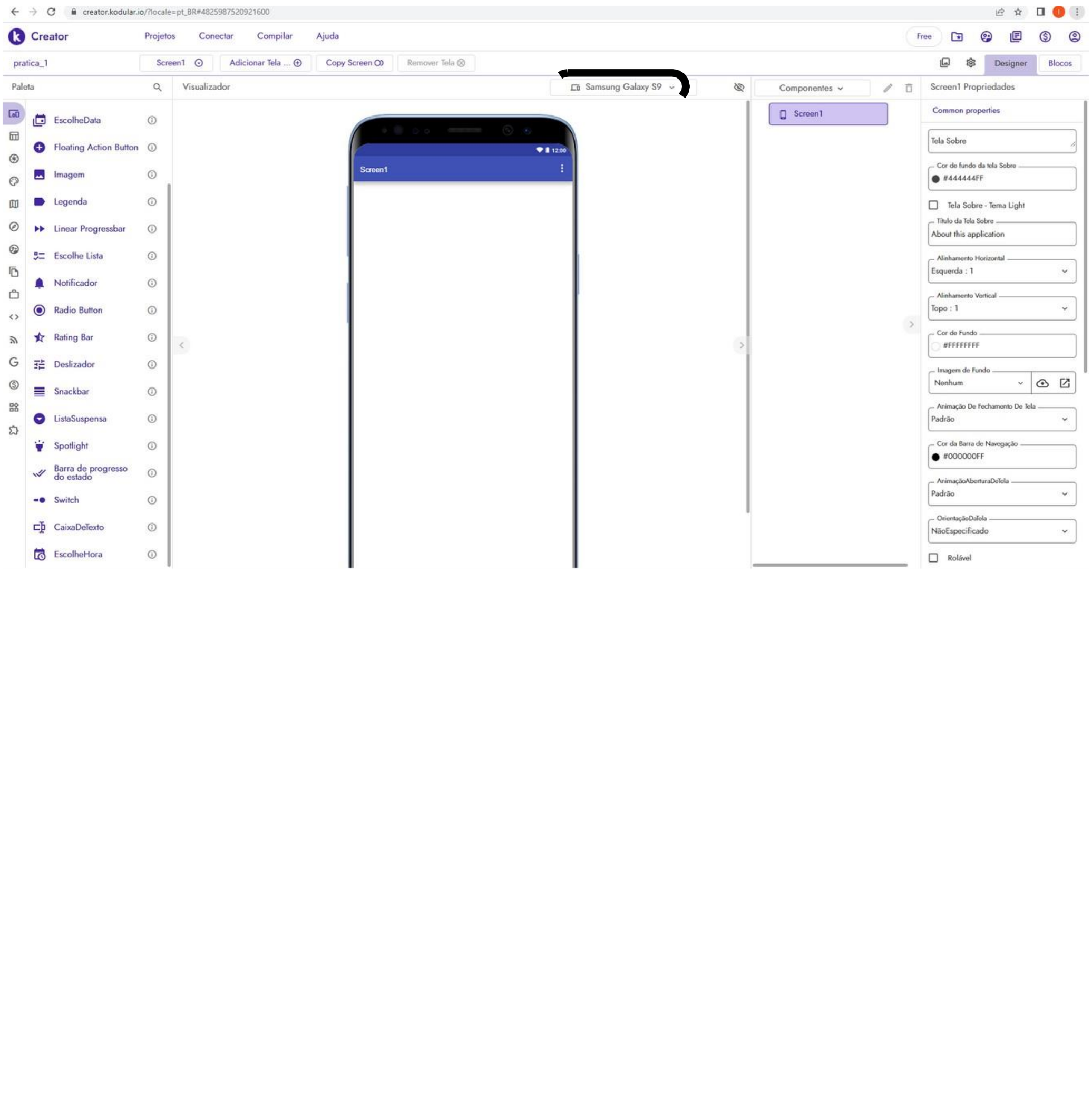
Após fazer o login em sua conta, você será redirecionado para uma nova página onde poderá fazer algumas configurações, caso deseje. No entanto, o foco desta introdução é ensinar como criar um aplicativo no Kodular. Para isso, basta retornar para a tela em que estava antes de fazer o login e clicar novamente em "Create Apps", o que o levará à página do Kodular Creator.

Se você preferir usar o Kodular Creator em português, pode facilmente alterar a linguagem. Basta clicar no último ícone na parte superior direita da tela, selecionar a ferramenta "Preferences" e, na seção "Languages", escolher "Português do Brasil". Assim, todos os textos e menus do Kodular Creator serão exibidos em português.



1.1 - Criando Projeto

Esta etapa é bastante simples. Primeiramente, clique em "Criar projeto", dê um nome a ele e clique em "Avançar". Em seguida, atribua um nome ao seu aplicativo e clique em "Finish". Isso irá direcioná-lo para a página de layout da tela do seu aplicativo, onde vamos começar a desenvolvê-lo. Antes de começar a trabalhar no layout, é recomendável alterar o modelo de smartphone para o "Samsung Galaxy S9", pois sua tela é maior e mais adequada para trabalhar. Para fazer isso, basta selecionar o modelo na aba abaixo.



2 - Calculadora

Devido as práticas com o kodular serem um pouco extensas, ensinaremos elas através de vídeo aulas. O primeiro aplicativo que faremos será uma calculadora básica capaz de somar, subtrair, multiplicar e dividir.

Fim

Esperamos que você tenha gostado. Nos esforçamos muito para fazer um texto divertido e de fácil digestão. Obrigada pela paciência e é isso aí! Nos vemos na próxima!

Com carinho, equipe Synesthesia Vision.

Quer saber mais sobre o projeto Synesthesia Vision? Entre no nosso site e fique por dentro!

https://synesthesiavision.com/

Ou entre em contato conosco através do email (aidaferreira@recife.ifpe.edu.br)