

# Guia de Projetos e Experimentos nas Plataformas Arduino e ESP32

Ricardo Gomes Lopes

Pedro Fernando Teixeira Dorneles

Luiz Antônio de Quadros Dworakowski

Januário Dias Ribeiro



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

Guia de projetos e experimentos nas plataformas  
Arduino e ESP32 [livro eletrônico] / Ricardo  
Gomes Lopes ... [et al.]. -- Bagé, RS :  
Ed. dos Autores, 2024.  
PDF

Outros autores: Pedro Fernando Teixeira Dorneles,  
Luiz Antônio de Quadros Dworakowski, Januário Dias  
Ribeiro.

Bibliografia.  
ISBN 978-65-00-99483-4

1. Arduino (Linguagem de computador) 2. Arduino -  
Plataforma 3. Ciência da computação 4. Plataforma  
digital I. Lopes, Ricardo Gomes. II. Dorneles, Pedro  
Fernando Teixeira. III. Dworakowski, Luiz Antônio de  
Quadros. IV. Ribeiro, Januário Dias.

24-201483

CDD-005.133

**Índices para catálogo sistemático:**

1. Arduino : Linguagem de programação : Computadores  
: Processamento de dados 005.133

Aline Grazielle Benitez - Bibliotecária - CRB-1/3129

## SUMÁRIO

Prefácio .....	4
FICHA DE IDENTIFICAÇÃO DO PRODUTO EDUCACIONAL.....	5
Parte I – Projeto no Arduino-Ardublock .....	7
Capítulo 1 – Revoluções Industriais .....	7
Capítulo 2 – Introdução à plataforma Arduino-Ardublock.....	16
Capítulo 3 – Construção de cronômetro utilizando a plataforma Arduino e a utilização do LDR .....	23
Capítulo 4 – Correção de um código e Construção de um Projeto (Sistema de irrigação automático de uma horta escolar) Parte I.....	38
Capítulo 5 – Construção de um sistema de irrigação automatizado (Parte II)...	47
Capítulo 6 – Sistema de irrigação automatizado Parte Final.....	50
Parte II – Tecnologia aplicada em outros contextos .....	57
Capítulo 7– Energia Eólica .....	58
Capítulo 8 – Meio Ambiente.....	63
Capítulo 9 – Saúde .....	71
Capítulo 10 – Conectividade.....	82
REFÊRENCIAS.....	121
Anexo A – Guia de Instalação do ARDUBLOCK.....	122
Apêndice A – Repositório Ardublock Modificado .....	125
Apêndice B – Guia de Instalação da Placa ESP32 .....	126
Apêndice C – Licença Creative Commons.....	128

## Prefácio

Este ebook é fruto da minha dissertação de mestrado, mais especificamente um dos produtos educacionais. A temática aqui foi desenvolvida para a criação de projetos voltados para a plataforma Arduino, sendo a primeira parte voltada para a construção de um projeto que envolvesse o interesse dos alunos, sendo a escolha destes a construção de um sistema de irrigação automatizado e o segunda parte a aplicação da tecnologia em diferentes contextos, sendo eles: energia eólica, meio ambiente, saúde e conectividade.

Também é necessário deixar claro que o guia é um dos produtos, sendo o outro as modificações no software Ardublock, ou seja, o guia é apenas um instrumento para ser utilizado juntamente com o software. O software modificado do Ardublock consta no Apêndice A, juntamente com as bibliotecas necessárias para o funcionamento do código.

As modificações do Ardublock feitas permitem a programação da placa ESP32 utilizando a estruturas de blocos. Inicialmente não existem registros de modificações para a programação do ESP32 no Ardublock, sendo este trabalho, o primeiro a fazer uma estruturação e com a finalidade de auxiliar estudantes e professores.

Gostaria neste prefácio de agradecer ao professor Luiz Antônio de Quadros Dworakowski por ter cedido suas aulas e o espaço em sua escola para a aplicação do meu projeto de mestrado.



## **FICHA DE IDENTIFICAÇÃO DO PRODUTO EDUCACIONAL**

### **GUIA DE PROJETOS E EXPERIMENTOS NAS PLATAFORMAS ARDUINO E ESP32**

**Ricardo Gomes Lopes**

**Pedro Fernando Teixeira Dorneles**

**Luiz Antônio de Quadros Dworakowski**

**Januário Dias Ribeiro**

#### **RESUMO**

Esse Produto Educacional é resultado da pesquisa de Mestrado Profissional em ensino de ciências que buscou responder as seguintes questões de pesquisa: o referencial teórico pensamento computacional pode ser aplicado na plataforma Arduino? Os pilares do pensamento computacional eles podem ser intercambiáveis? Quais os efeitos da construção de uma aplicação levando os interesses dos estudantes? Qual o efeito da contextualização de uma aplicação no Arduino em relação ao interesse do estudante? Para sua elaboração utilizou-se o seguinte referencial teórico: pensamento computacional e a análise dos dados foi feita usando metodologia qualitativa baseada na pesquisa de uma intervenção pedagógica proposta por Damiani (2013). A intervenção foi aplicada no clube de ciências com voluntários e bolsistas júnior do projeto feira /de ciências do campus Bagé na Escola Estadual de Ensino Médio Jerônimo Mércio da Silveira no período de 27 de outubro de 2021 até 09 de dezembro de 2021 a aplicação do primeiro capítulo do produto e de 09 de setembro de 2022 até 24 de novembro de 2022 o segundo capítulo, tendo um total de 11 horas de aplicação.

Informações adicionais.

- Público ao qual o PE se destina: Professores, alunos do ensino médio e pesquisadores.
- Nível de ensino (escolaridade) ao qual o PE se destina: Ensino Médio
- Modalidade de ensino a qual o PE se destina: Regular
- Conteúdo curricular abordado no PE: Ensino de programação na plataforma Arduino, sensores e atuadores, variáveis inteiras e decimais, projeto de

irrigação automatizado, energia eólica aplicada a tecnologia, sensores de gás, batimento cardíaco e conectividade na placa ESP32.

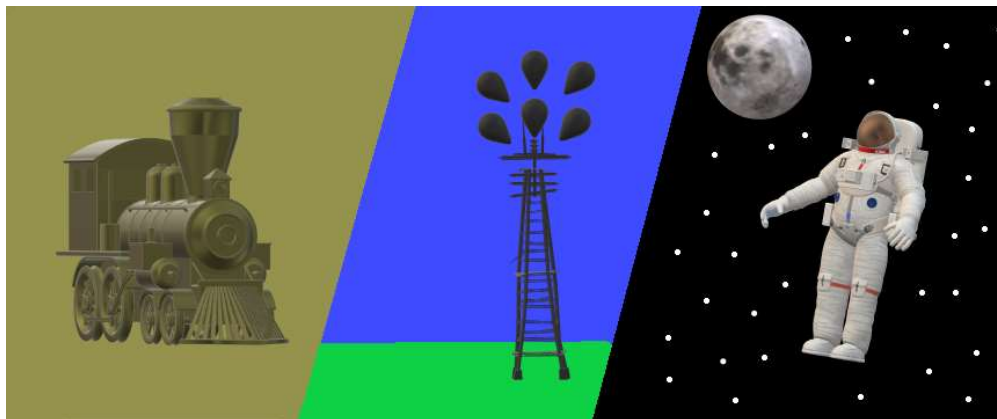
- O PE disponibiliza manual/instruções: Possui em apêndices os guias de instalação do Ardublock e do ESP32. O produto educacional como todo é uma guia para projetos
- O produto envolve conhecimentos de trabalhos prévios do próprio autor.
- Licença *creative commons* no PE – Apêndice C
- Guia de projetos e experimentos nas plataformas Arduino e ESP32 © 2023 by Ricardo Gomes Lopes is licensed under Attribution-NonCommercial 4.0 International

# Parte I – Projeto no Arduino-Ardublock

## Capítulo I – Revoluções Industriais

Você pode estar se perguntando por que iremos falar sobre história em um guia de experimentos envolvendo novas tecnologias? A história e as ciências caminham lado a lado. Para entendermos o que o futuro pode nos reservar, precisamos compreender o passado e os impactos que tiveram na sociedade. Outro ponto que devemos levar em conta é que ciência e tecnologia avançam juntas: a cada nova descoberta no âmbito científico pode, por consequência, trazer avanços tecnológicos. Da mesma forma, avanços tecnológicos podem nos levar a novas descobertas científicas. Até agora, tivemos três revoluções industriais. O que elas podem nos mostrar?

Figura 1 – Representação das três primeiras revoluções industriais



Fonte: Produzido pelo autor via *software paint 3D* (2021)

Na imagem acima, podemos ver a representação de três eventos marcantes das primeiras três revoluções tecnológicas.

A primeira é a imagem de um trem, uma máquina a vapor amplamente utilizada para locomoção de uma localidade para outra. A tecnologia envolvida se dá pela queima de carvão. Nosso ponto de partida começa na terra do Rei: a Inglaterra, mais especificamente no século XVIII.

A primeira revolução industrial ocorreu por volta dos anos 1760 a 1850, tendo seu início na Inglaterra. Nas décadas seguintes, espalhou-se pela Europa Ocidental e Estados Unidos.

A criação de máquinas termodinâmicas movidas a vapor pela queima de carvão, a manipulação de minérios de ferro e as mudanças nos processos de trabalho modificaram completamente a política, a sociedade e a economia, passando de um regime econômico feudal que predominava desde o século III para um sistema capitalista. Durante este período, houve o início de um processo migratório das regiões rurais para grandes centros urbanos, que se intensificou cada vez mais durante as próximas revoluções industriais. Com o surgimento das primeiras indústrias e as mudanças no sistema econômico e nos processos migratórios, começaram a surgir novas divisões na sociedade (Figura 2): a classe operária, que trabalhava majoritariamente nas indústrias, e a burguesia, que eram comerciantes de produtos e os donos ou financiadores das indústrias.

Figura 2 – Representação da Classe Trabalhadora e da Burguesia



Fonte: Produzido pelo autor via *software paint 3D* (2021)

Essa mudança também modificou o ensino, levando ao surgimento de escolas para educar os filhos da classe burguesa. A educação, que anteriormente era realizada pela igreja e restrita a apenas algumas pessoas da alta classe ligada à monarquia, passou a ser feita por estas instituições denominadas "escolas", e ocorreu o surgimento da figura chamada "professor". Entretanto, é necessário salientar que o ensino não era universal. A classe operária teve acesso a uma educação mais técnica e profissional, pois era necessário um grau mínimo de instrução para operar as máquinas. Muitas vezes, os filhos dos trabalhadores da classe operária eram empregados ainda na infância nas indústrias, em condições de trabalho que se assemelhavam à escravidão, conforme termo utilizado atualmente. Toda a classe operária realizava trabalhos de forma mecanizada (repetitiva), com jornadas exaustivas e baixa remuneração.



A segunda revolução industrial ocorreu de 1850 até 1945 (encerrando-se após a Segunda Guerra Mundial). Apesar de ter iniciado na Inglaterra, rapidamente se espalhou para outras partes do mundo, como a Europa Ocidental, Estados Unidos, Rússia e a modernização do Japão. A descoberta e utilização da eletricidade, a manipulação do aço, o petróleo como fonte de energia para máquinas de combustão interna, a automatização de vários processos industriais através de máquinas movidas a engrenagens que levaram à produção em massa de produtos, a indústria química na produção de remédios e adubos, e modelos de trabalho com o "*fordismo*"<sup>1</sup>, acentuaram as modificações da primeira revolução nos aspectos econômicos, sociais e políticos. Neste período, houve um aumento ainda maior da migração das regiões rurais para os grandes centros urbanos, conhecido historicamente como o "êxodo rural". Existiram mudanças sociais significativas, inspiradas pelas revoluções burguesas como a Revolução Francesa (1789-1799). Durante este período histórico, a classe operária se organizou por melhores condições de trabalho e pela luta por direitos iguais entre homens e mulheres. Houve também tensões políticas que resultaram em duas guerras mundiais (com as maiores potências mundiais envolvidas), a primeira tendo seu estopim com a morte do Arquiduque Austro-Húngaro Francisco Fernando e a segunda com a invasão da Alemanha Nazista à Polônia. Nos avanços tecnológicos desenvolvidos pela sociedade destacam-se: o desenvolvimento do avião, da eletricidade e iluminação de grandes cidades, o rádio, o telefone, os carros e as bombas nucleares (que serão um dos fatores que levarão a humanidade à terceira revolução industrial). O ensino sofreu mudanças significativas, como o início da ideia da educação universal e o fortalecimento da educação acadêmica (universidades).

---

<sup>1</sup> Modelo de produção criado por *Henry Ford* que visava a diminuição dos custos de produção e o aumento de produtividade. Cada indivíduo realizava uma única tarefa, dividindo-se em estações de trabalho.

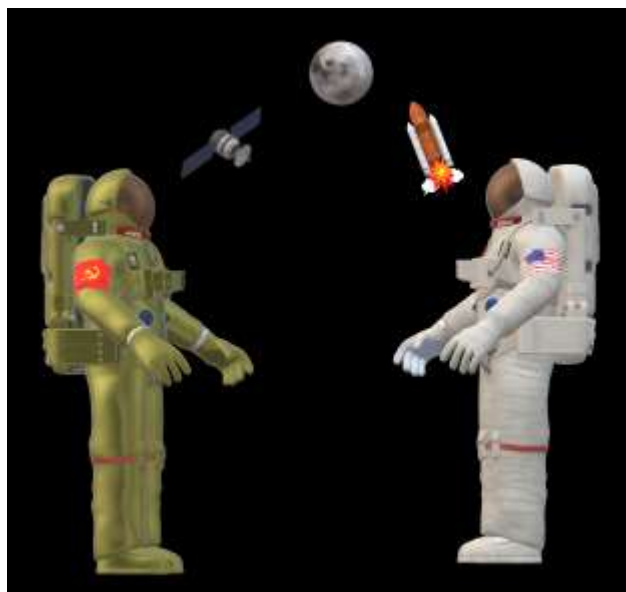
Figura 3 – Algumas das criações da segunda revolução industrial



Fonte: Produzido pelo autor via *software paint 3D* (2021)

A terceira revolução industrial começou entre 1945 e 1950, e historicamente ainda não está definido quando será o seu encerramento. Com o fim da Segunda Guerra Mundial, os Estados Unidos tornaram-se a maior potência econômica do mundo. Esta revolução teve início nos Estados Unidos e na União Soviética, espalhando-se por todo o mundo, inclusive no Brasil, com a modernização da indústria que teve seu início na era de Getúlio Vargas. A criação e o desenvolvimento da computação, genética, robótica, telecomunicações e o aprimoramento da indústria como um todo e da farmacêutica levaram a humanidade a um novo patamar tecnológico. O início dessa revolução é marcado pela Guerra Fria entre Estados Unidos (capitalista) e União Soviética (socialista). Ambas as nações desenvolveram e/ou aperfeiçoaram armas nucleares, havendo o risco de uma Terceira Guerra Mundial (que não ocorreu), na qual ambas as nações poderiam utilizar tal armamento. No contexto dessa disputa de poder, ocorreu a corrida espacial (Figura 4), que tinha como objetivo levar o homem ao espaço e, posteriormente, à Lua. O objetivo era demonstrar à outra nação quem tinha mais poder. O primeiro homem a sair da atmosfera terrestre foi Yuri Gagarin (russo) e o primeiro homem a pisar na Lua foi Neil Armstrong (estadunidense).

Figura 4 – Representação da Corrida Espacial



Fonte: Produzido pelo autor via *software paint 3D* (2021)

Durante esse período houve a evolução tecnológica nunca antes vista (Figura 5), como o envio de satélites artificiais que propiciaram a evolução das tecnologias de comunicação e ao desenvolvimento da computação (que ajudaram nos cálculos que levaram o homem à lua). A criação da televisão, os computadores domésticos (e sua constante evolução), da internet, tecnologias digitais, de vacinas e remédios e a evolução da indústria com a computação, mudaram aspectos sociais profundos, por exemplo de como o ser humano se relaciona com seu meio e ao acesso a um grande número de informações. A universalização do ensino se torna mais presente em todo o mundo e é uma necessidade para se viver em uma sociedade moderna. A Organização das Nações Unidas (ONU) que teve sua criação logo após o fim da segunda guerra mundial e tem como uma de suas principais metas a “educação de qualidade”, de modo que, surgiram indicadores de ensino como o Programa Internacional de Avaliação de Alunos (PISA), que avaliou os estudantes nas principais áreas do conhecimento de mais de 79 países (incluindo o Brasil) no ano de 2018

Figura 5 – Algumas criações da terceira revolução industrial

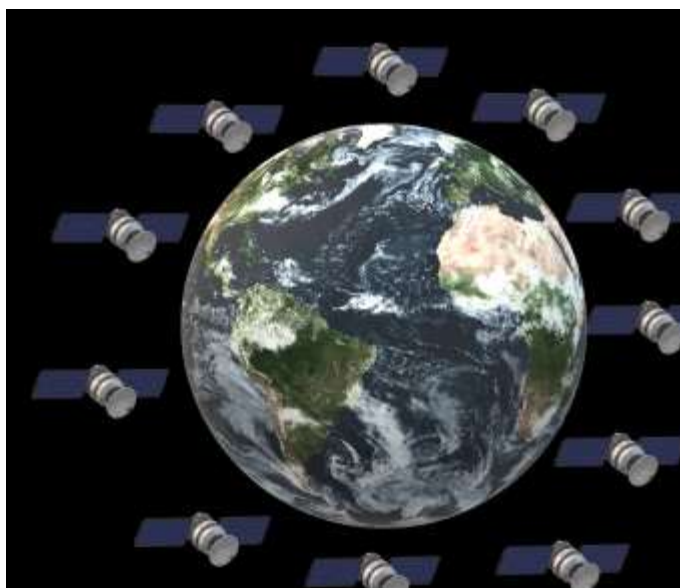


Fonte: Produzido pelo autor via *software paint 3D* (2021)

Agora vamos ao questionamento final, quando começa a próxima revolução industrial?

Na verdade, a Quarta Revolução Industrial já teve seu início histórico, situado por volta do ano de 2010. Não é algo que está por vir, mas sim algo que já está acontecendo. Ao analisarmos três fatores bastante característicos da Indústria 4.0, destacamos o seguinte: 1. Ela se caracteriza por conexões de alta velocidade à internet, como a rede móvel 5G (5ª geração) e a internet via satélite fornecida por empresas como a SpaceX (Figura 6); 2. Essa revolução impulsiona automações tanto nas indústrias quanto nos serviços, substituindo gradualmente a mão de obra humana por máquinas automatizadas; 3. Informações coletadas principalmente por meio de redes sociais são utilizadas para a venda de produtos direcionados diretamente ao usuário, potencializando cada vez mais o uso desses dados.

Figura 6 – Representação do que seria a rede de satélite da starlink



Fonte: Produzido pelo autor via *software paint 3D* (2021)

As informações coletadas pelas redes sociais já foram utilizadas no aspecto político. Como exemplo, podemos citar as eleições americanas de 2016 e a utilização de informações de 50 milhões de usuários da rede social Facebook pela empresa Cambridge Analytica. Essa empresa direcionava a publicidade e propagava notícias falsas ("Fake News") para determinados públicos-alvo, com o objetivo de eleger o candidato republicano Donald Trump. Ainda podemos mencionar o exemplo do Brasil nas eleições de 2018, com o disparo de mensagens em massa pelo aplicativo WhatsApp, novamente com notícias falsas ("Fake News") com o propósito de desacreditar um dos candidatos presidenciais. Este fato está sendo investigado atualmente por uma Comissão Parlamentar Mista de Inquérito (CPMI) na Câmara dos Deputados e pelo Tribunal Superior Eleitoral (TSE). Em ambos os casos, foram utilizados algoritmos complexos para o disparo de mensagens e publicidade. No caso das eleições americanas, essa complexidade foi ainda mais longe, traçando o perfil psicológico dos usuários. O empresário, CEO e proprietário das empresas Facebook, WhatsApp e Instagram, Mark Zuckerberg, foi convidado a prestar esclarecimentos no Senado Americano em 10 de abril de 2018. Uma das pautas discutidas foi o impacto de suas redes sociais nos regimes democráticos ao redor do mundo. Ainda enfrentamos o problema do uso excessivo dessas tecnologias e dos algoritmos construídos para causar dependência nos usuários<sup>2</sup>, estimulando principalmente a liberação de dopamina no cérebro. Essas programações são desenvolvidas de forma intencional para prender a atenção e coletar informações para publicidade direcionada especificamente para cada indivíduo, conforme sua personalidade e gostos pessoais. A ética envolvida nesse tipo de tecnologia é altamente questionável. Existem mais exemplos dos impactos das redes sociais e das informações coletadas, no entanto, o objetivo aqui não é fazer um levantamento total, mas sim fornece uma descrição de um pequeno recorte da realidade.

Um dos principais problemas sociais é, inicialmente, a precarização da mão de obra. Serviços oferecidos por aplicativos de celular, como o serviço de entregas, compras em supermercados e transporte de pessoas, estão atualmente levando os segmentos sociais mais vulneráveis à informalidade e sem os direitos trabalhistas básicos. No entanto, à medida que a automação evolui, esses serviços tendem a se

---

<sup>2</sup> Recomenda-se a leitura da reportagem pelo acadêmico Salcedo da UFSM no ano de 2021: <https://www.ufsm.br/midias/arco/como-redes-sociais-hackeiam-sua-mente>

tornar cada vez menos dependentes da mão de obra humana. Por exemplo, os veículos auto dirigíveis (Figura 7) estão substituindo os seres humanos nessas funções, o que nos leva ao segundo problema social: a mudança na demanda de trabalho. Inicialmente, com a falta de mão de obra qualificada para esse novo paradigma, ocorrerá um aumento nos índices de desemprego, deixando uma parcela maior da sociedade em vulnerabilidade social. Essa última análise nos leva ao ponto de discussão sobre uma mudança de paradigma também no ensino, o que será explorado no próximo parágrafo. Além disso, existem conflitos éticos relacionados ao uso das tecnologias e como todas as informações coletadas pelas empresas serão utilizadas.

Figura 7 – Representação de Veículos auto dirigíveis



Fonte: Produzido pelo autor via *software paint 3D* (2021)

Durante esses aproximadamente 260 anos desde a primeira revolução industrial, tivemos uma evolução tecnológica em crescimento exponencial e diversas mudanças de paradigmas, inclusive no âmbito do ensino. Esses últimos pontos discutidos sobre a quarta revolução industrial têm como objetivo levantar um questionamento chave: é senso comum que a função da escola está atrelada à formação humana e profissional do indivíduo. No entanto, como a escola deve atuar diante dessa mudança no paradigma tecnológico e social?

## **ATIVIDADES DO CAPÍTULO 1**

Sugere-se ao professor a discussão dos dois tópicos a seguir com seus alunos, com a finalidade de promover uma reflexão do papel da tecnologia no mundo e como os alunos imaginam a escola no futuro.

**1 – Descreve como você acha que vai ser o futuro com base no que foi estudado no módulo 1?**

**2 – Quais impactos você acredita que vão ocorrer devido a quarta revolução industrial?**

## Capítulo 2 – Introdução à plataforma Arduino-Ardublock

Iremos trabalhar agora com o *Arduino*, mas você pode estar se perguntando, o que é o *Arduino*? Os desenvolvedores oficiais descrevem tal plataforma da seguinte maneira:

“O Arduino é uma plataforma de prototipagem eletrônica open-source que se baseia em hardware e software flexíveis e fáceis de usar. É destinado a artistas, designers, hobbistas e qualquer pessoa interessada em criar objetos ou ambientes interativos. O Arduino pode sentir o estado do ambiente que o cerca por meio da recepção de sinais de sensores e pode interagir com os seus arredores, controlando luzes, motores e outros atuadores. O microcontrolador na placa é programado com a linguagem de programação Arduino, baseada na linguagem Wiring, e o ambiente de desenvolvimento Arduino, baseado no ambiente Processing. Os projetos desenvolvidos com o Arduino podem ser autônomos ou podem comunicar-se com um computador para a realização da tarefa, com uso de software específico (ex: Flash, Processing, MaxMSP).” (Arduino oficial)

E de acordo com Lopes (2019) o Arduino é:

[...] uma plataforma de testes para construção de produtos finais. A placa Arduino é projetada com um microcontrolador da Atmel AVR, com pinos digitais e analógicos para automação de projetos, seu conceito é de hardware e software livre, ou seja, permite que o usuário faça modificações e construa sua própria placa ou software baseado na plataforma do Arduino, o que além de tornar o seu custo mais barato, também propicia a criação de uma comunidade de usuários bastante ampla que se auxiliam em seus projetos tanto na construção do circuitos quanto na programação.

A placa que iremos utilizar neste minicurso é o Arduino Uno, tem como principais características:

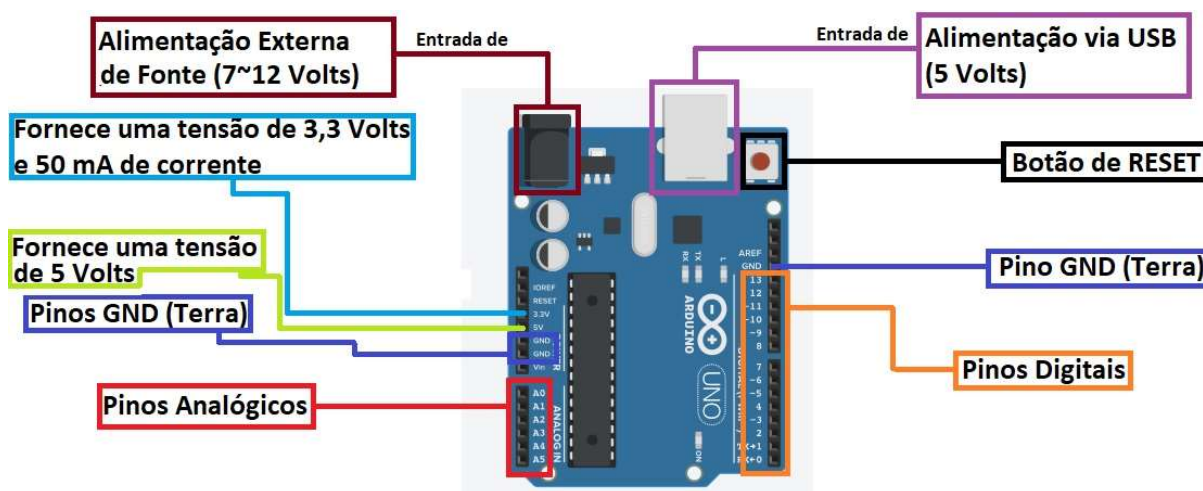
- 6 pinos analógicos;
- 14 pinos digitais;
- 1 Pino de que fornece uma tensão de 5 Volts;
- 1 pino que fornece uma tensão de 3,3 Volts e 50 mA de corrente;
- 3 pinos GND (terra);
- 1 entrada de alimentação via USB de 5 Volts;
- 1 entrada de alimentação externa para baterias 7~12 Volts;



- 1 Botão de RESET (Reinicia a programação a programação do Arduino). (Lopes, 2019, p.87)

Observe a Figura 8 para ver onde ficam as características mencionadas acima:

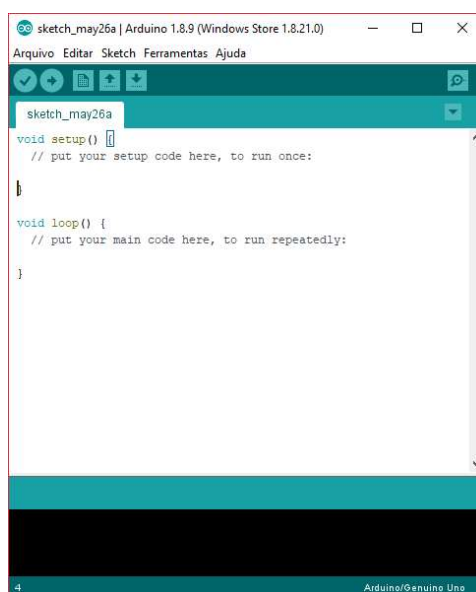
Figura 8 – Placa Arduino Uno



Fonte: Lopes (2019).

O Arduino também tem um ambiente de programação chamado *Arduino IDE* que de acordo com Lopes (2019, p. 87) “é uma aplicação em Java derivada dos projetos Processing e Wiring, sua programação é feita através de linhas de comando”.

Figura 9 – Ambiente de programação em linhas no Arduino (Arduino IDE)



Fonte: Lopes (2019)

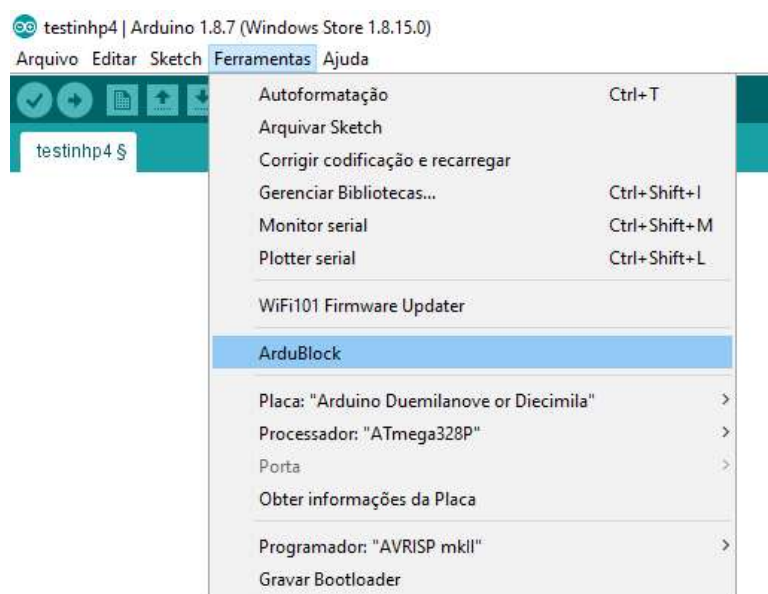
Porém iremos trabalhar com uma interface mais “amigável” no ponto de vista da programação, utilizaremos o Ardublock <sup>3</sup>. Consta também no Apêndice A o repositório do Ardublock Modificado pelo autor.

Mas o que seriam o “Ardublock? Lopes (2019) descreve que:

“O Ardublock é um aplicativo em Java no qual adicionamos como uma ferramenta no Arduino IDE, sua principal utilidade é no ensino de programação, uma vez que, neste aplicativo, o usuário cria uma programação utilizando blocos e este é transcrito para a linguagem em linhas utilizada na no IDE do Arduino

Para abrir o Ardublock, primeira abra o Arduino IDE vá em “Ferramentas” e selecione a opção “Ardublock conforme a Figura 10.” (Lopes, 2019, p.90)

Figura 10 – Aba do Ardublock no Arduino IDE.

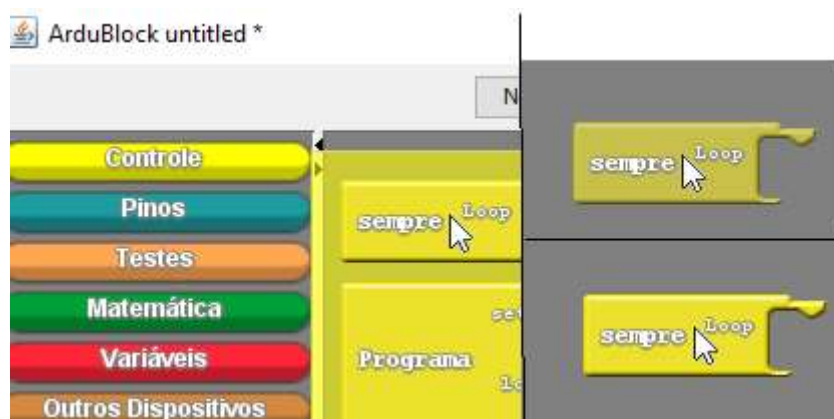


Fonte: LOPES (2019)

A programação no Ardublock, como pode-se observar no próprio nome, é em blocos, ou seja, basta arrastar os blocos (Figura 11) para parte cinza para começarmos a programar. (Lopes, 2019, p.90)

<sup>3</sup> Disponível no Anexo A o guia de instalação do Ardublock.

Figura 11 – Arrastar Blocos.



Fonte: LOPES (2019)

Como podemos ver na Figura 12, o ArduBlock possui alguns comandos básicos.

Figura 12 – Comandos do ArduBlock.



Fonte: Autor (2021).

A seguir são descritos brevemente os comandos ilustrados na Figura 21.

**Novo:** Este comando é para quando necessitamos iniciar uma nova programação no ArduBlock, ou seja, criar um novo programa.

**Salvar:** Serve para salvar a programação que está sendo realizada no ArduBlock.

**Salvar Como:** Se assemelha ao comando anterior com a diferença de que, é possível salvar em outro formato e/ou com outro nome.

**Abrir:** Serve para abrir programações que foram salvas no computador.

**Enviar para o Arduino:** Este é um dos comandos mais importante que tem como função converter a linguagem em blocos para a linguagem em linhas de comando (C++). Após terminar a programação, é necessário enviar o programa para plataforma do Arduino IDE.

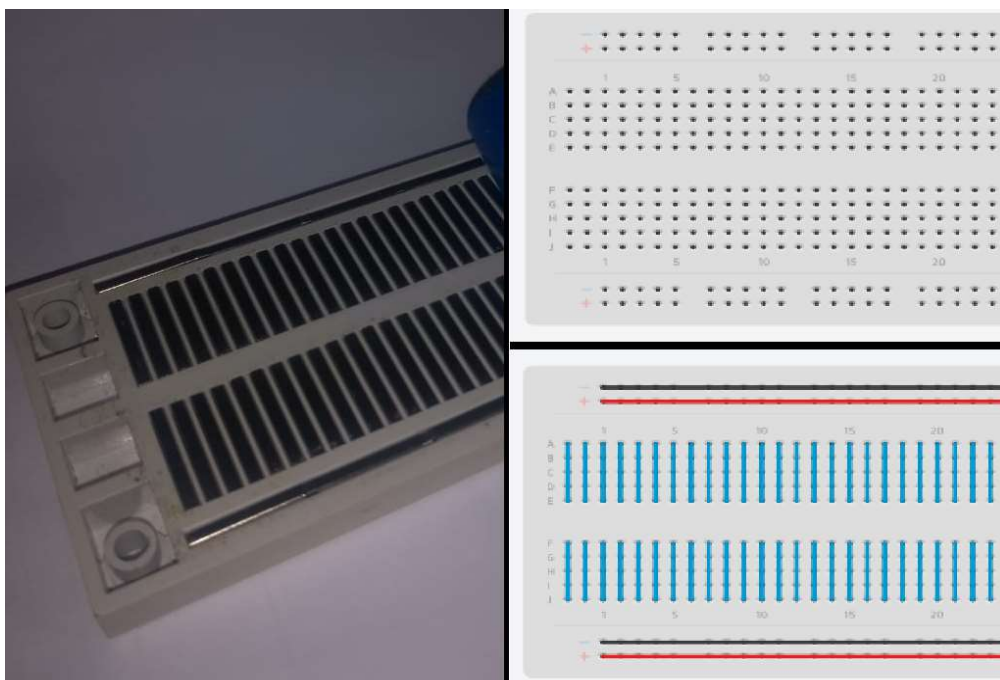
**Monitor Serial:** Este comando serve para monitorar a comunicação Serial entre a placa do Arduino e o computador. (Lopes, 2019, p.92)

Por último, vamos abordar o uso da protoboard:

A *protoboard* é uma placa de ensaios experimentais ou também conhecida como placa de prototipagem, ou seja, como o próprio nome diz, é uma placa que permite o teste de circuitos eletrônicos e a criação de protótipos, sem a necessidade de soldar componentes.

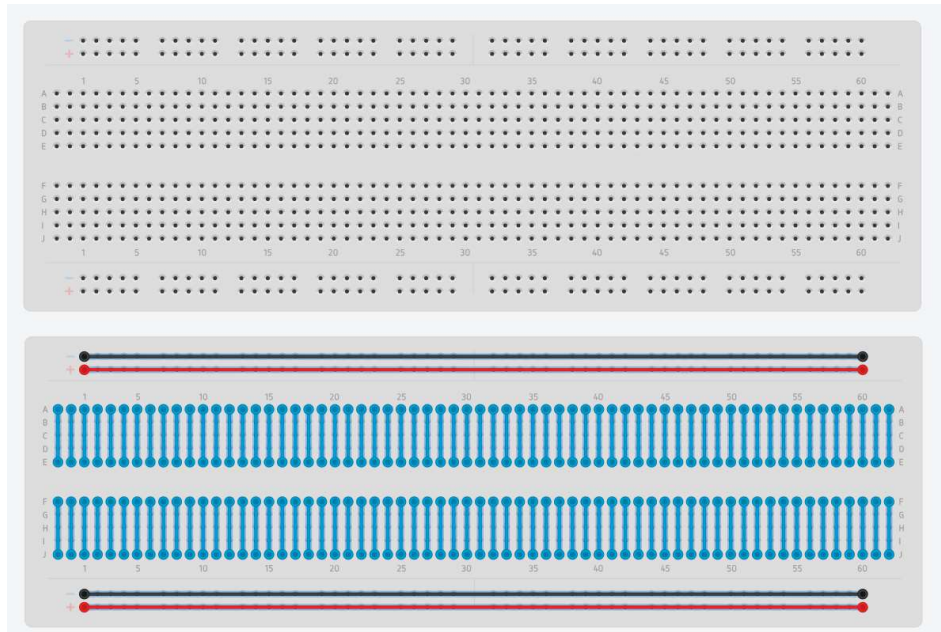
A *Protoboard* possui vários “furos” organizados na seguinte maneira: o centro (onde uma placa metálica condutora que liga estes “furos” na vertical) e os extremos (onde tem uma placa metálica que liga estes “furos” na horizontal”). Podemos observar isso melhor na Figura 13 (Lopes, 2019, p.89)

Figura 13 – *Protoboard* e seu funcionamento



Fonte: Lopes (2019)

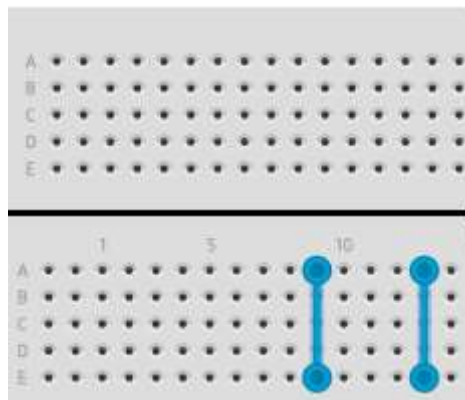
Figura 14 – Protoboard e suas conexões



Fonte: Lopes (2019)

Pense agora na seguinte situação: duas dessas linhas de “furos” não estão conectadas entre si, como pode ser vista na Figura 15 (as linhas em azul): (Lopes, 2019, p.89)

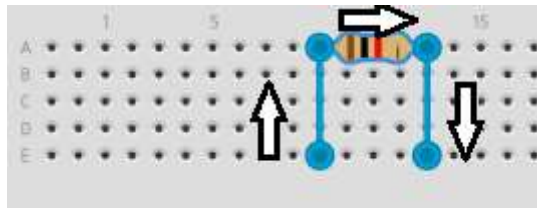
Figura 15 – Protoboard conexões duas linhas centrais



Fonte: Lopes (2019)

Agora, ao conectarmos um resistor entre esses dois terminais, estes passarão a ficar conectados entre si, conforme a Figura 16:

Figura 16 – *Protoboard* (resistor conectando os terminais em azul)



Fonte: Lopes (2019).

## ATIVIDADES DO Capítulo 2

### 1 – Blink (Acender e apagar um LED)

Um dos programas mais simples que existem na plataforma Arduino é fazer um LED (*Light Emitting Diode* - Tradução: Diodo Emissor de Luz) piscar, sendo que a própria placa Arduino possui um embutido ligado a um pino digital (normalmente ao pino 13) com a finalidade de realizar alguns testes. A programação será feita na extensão Ardublock para plataforma Arduino IDE. (Lopes, 2019, p.94)

## Capítulo 3 – Construção de cronômetro utilizando a plataforma Arduino e a utilização do LDR

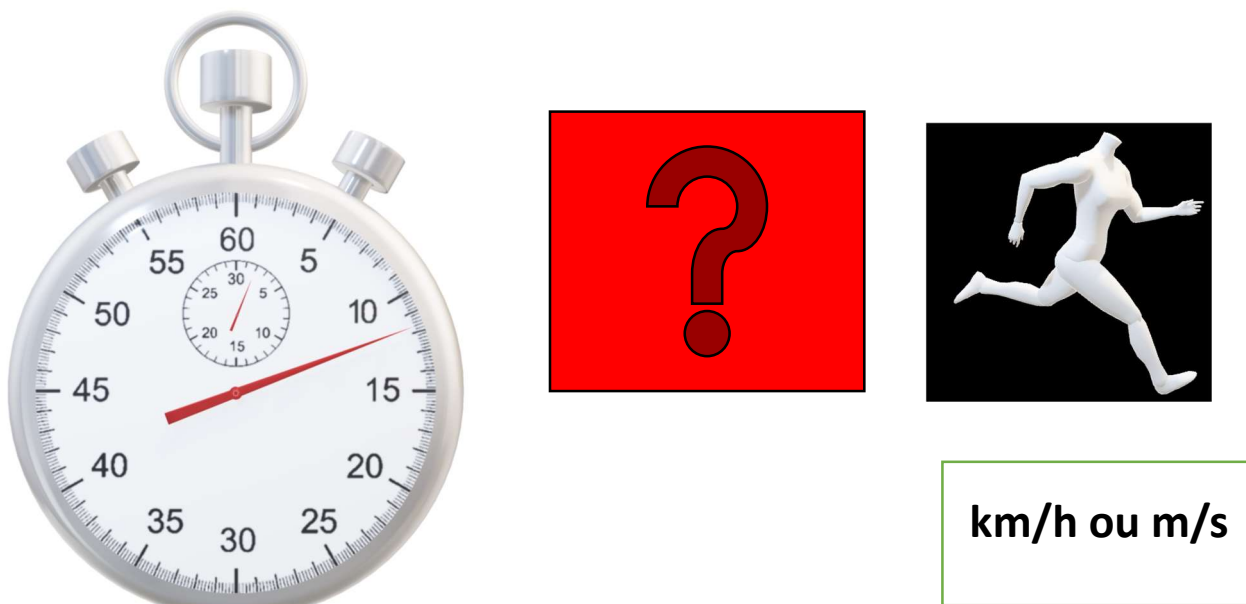
**Observação dos autores:** Dentre os sensores disponíveis, os estudantes escolheram três sensores, levando em consideração seus interesses. O sensor de luminosidade foi escolhido para ser trabalhado ao longo deste capítulo, enquanto os outros dois sensores serão abordados nos capítulos subsequentes.

A primeira atividade referente a este capítulo é a construção de um cronômetro utilizando a plataforma Arduino. O objetivo central é focar na utilização de variáveis inteiras e decimais e na diferenciação entre elas.

### 3.1 – Construção de um cronômetro

Qual a razão de medirmos tempo com um cronômetro?

Figura 18 – Cronômetro e grandezas dependentes do tempo



Fonte: Autor (2021).

Grandezas como velocidade ou aceleração dependem do tempo, ou seja, medimos uma dada distância e o tempo transcorrido.

Vamos diferenciar variáveis inteiras e decimais

- Variáveis inteiras servem para coletar certas informações que usa apenas números Inteiros: 1; 2; 3; 4; 5; 6; [...].
- Variáveis decimais servem para coletar certas informações que usa apenas números decimais: 3,14; 2,71; 9,8; 6,67; [...].

Figura 19 – Exemplo de váriavrel inteira



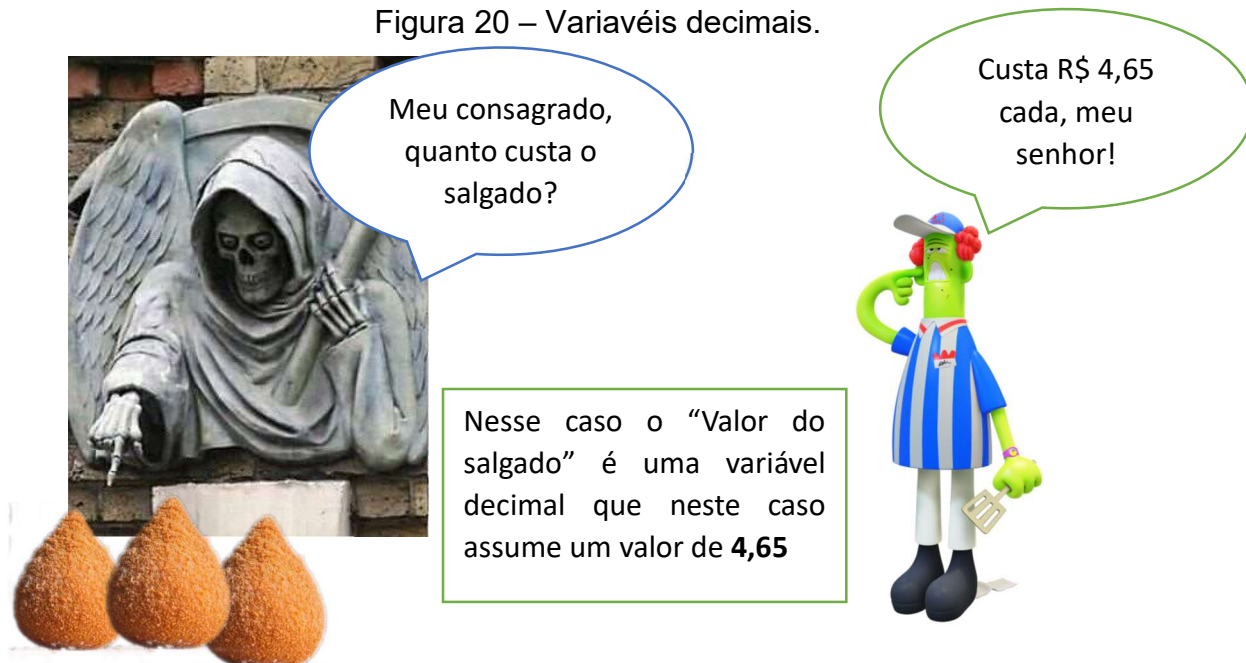
Fonte: Autor (2021).

Note que no caso acima o número de variável a ser armazenado será um número inteiro que nesse caso é o número de salgados.

A Figura 21 mostra as variáveis inteiras e decimais no ardublock:



Figura 20 – Variáveis decimais.

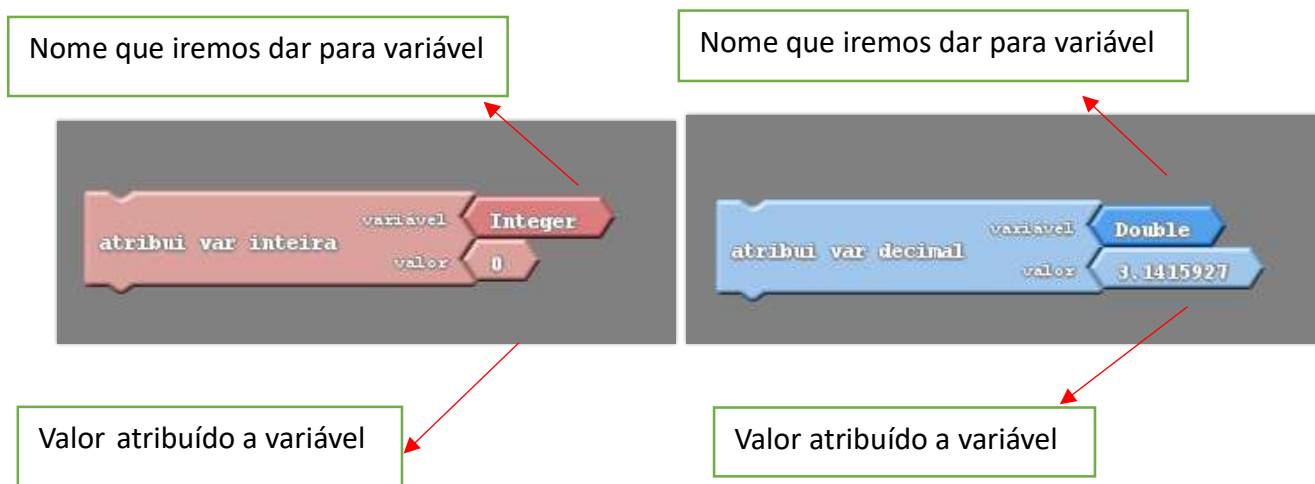


Fonte: Autor (2021).

No caso acima, o valor do salgado é um exemplo de um valor decimal que deve se utilizar variável do tipo decimal para seu armazenamento.

Veremos agora as variáveis inteiras e decimais no ardublock

Figura 21 – Variáveis inteiras e decimais no Ardublock



Fonte: Autor (2021).

Para a atividade de construção do cronômetro também iremos utilizar o bloco de tempo transcorrido:

Figura 22 – Bloco do tempo transcorrido no Ardublock



Fonte: Autor (2021).

Note que o bloco acima mede o tempo em milisegundos, sendo necessário realizar a divisão por 1000 para obtermos o tempo em segundos.

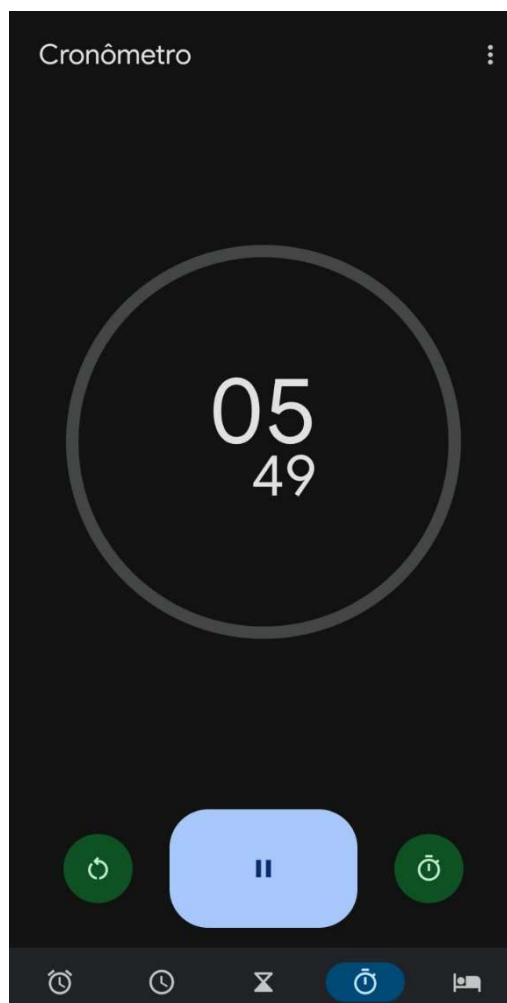
### 3.1.1 Atividade de construção do cronômetro

Construção de um cronômetro. Para esta atividade será necessário o seguinte material:

- Placa Arduino
- Computador com a IDE do arduino.

Antes de realizar qualquer atividade, responda a seguinte pergunta, qual tipo de variável devemos escolher? Para auxiliar na reflexão sobre essa pergunta abra o crônometro de seu aparelho de celular.

Figura 23 – Cronômetro do Celular



Fonte: Autor (2021).

Após este primeiro passo, observe que temos que comunicar ao usuário para que o mesmo possa ter acesso a essa informação coletadas pelo Arduino. Será necessário a utilização de blocos de comunicação, conforme pode ser visto na Figura 24:

Figura 24 – Comunicação bloco



Fonte: Autor (2021).

Inicialmente conecte a placa Arduino no computador e abra o Ardublock.

Construa o cronômetro utilizando tanto variáveis inteiras quanto decimais e analise a diferença entre os resultados. Abra o monitor serial para esta análise.

### Resolução:

A resolução desta atividade se dá através do seguinte código:

Figura 25 – Resolução do cronômetro



Fonte: Autor (2021).

Em seu trabalho, Lopes (2019, p.113), realiza esta mesma atividade e demonstra a diferença de se utilizar variáveis inteiras e decimais na construção do cronômetro, como mostra a Figura 26.

Figura 26 – Diferença dos resultados utilizando variáveis inteiras e decimais

```

COM5 (Arduino/Genuino Uno)
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 0
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1
Tempo transcorrido(s): 1

COM3
Tempo transcorrido(s): 1.21
Tempo transcorrido(s): 1.24
Tempo transcorrido(s): 1.27
Tempo transcorrido(s): 1.30
Tempo transcorrido(s): 1.34
Tempo transcorrido(s): 1.37
Tempo transcorrido(s): 1.40
Tempo transcorrido(s): 1.43
Tempo transcorrido(s): 1.46
Tempo transcorrido(s): 1.49
Tempo transcorrido(s): 1.52
Tempo transcorrido(s): 1.55
Tempo transcorrido(s): 1.59
Tempo transcorrido(s): 1.62
Tempo transcorrido(s): 1.65
Tempo transcorrido(s): 1.68
Tempo transcorrido(s): 1.71
Tempo transcorrido(s): 1.74
Tempo transcorrido(s): 1.77
Tempo transcorrido(s): 1.80
Tempo transcorrido(s): 1.84
Tempo transcorrido(s): 1.87
Tempo transcorrido(s): 1.90
Tempo transcorrido(s): 1.93
Tempo transcorrido(s): 1.96
Tempo transcorrido(s): 1.99

```

Fonte: Lopes (2019).

### 3.2 Introdução a sensores

Primeiramente vamos introduzir o conceito de resistor elétrico. O que vem a sua mente quando falamos em “resistência elétrica”?

Geralmente quando falamos de resistência elétrica o que vem a nossa mente é a “resistência de chuveiro” que no caso é um tipo de resistor elétrico.

Figura 27 – Chuveiro Elétrico



Fonte: Autor (2021).

É necessário fazer uma diferenciação entre resistência e resistor. Resistor é nome do dispositivo elétrico em si, resistência é uma propriedade física que tem como principal função barrar a passagem de corrente.

Resistores Ôhmicos seguem um comportamento linear, de tal modo que a lei de ohm para tais resistores terá um efeito linear na lei de Ohm pelo tempo.

Figura 28 – Lei de Ohm

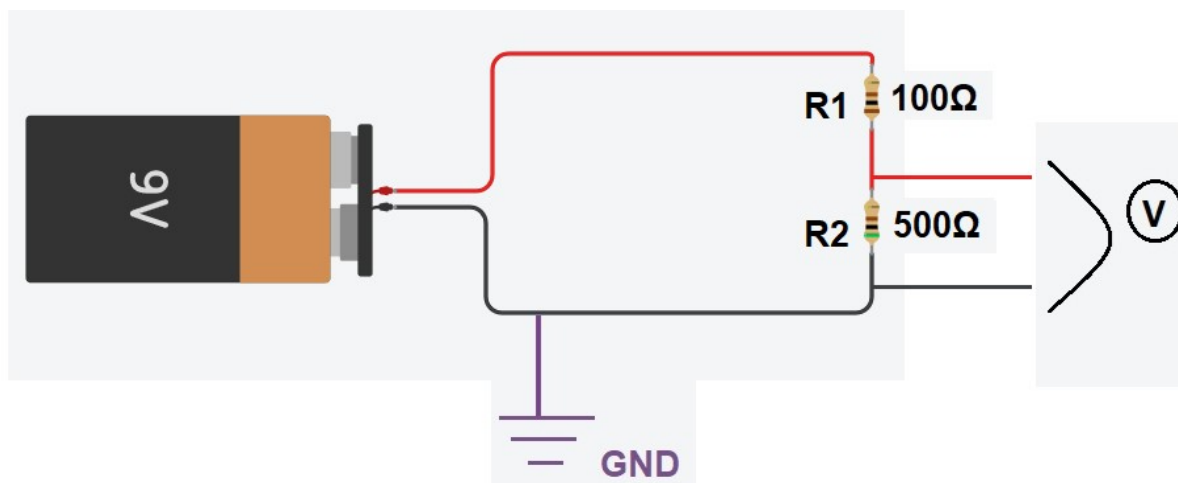
<p><math>i</math> – Corrente (A)</p> <p><math>V</math> – Diferença de potencial (volts)</p> <p><math>R</math> – Resistência (ohm)</p>
---

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

Fonte: Autor (2021).

A necessidade de compreensão do fenômeno da resistência elétrica se dá pelo fato de que ao associarmos duas resistências em série temos o que chamamos de divisão de tensão. O funcionamento de diversos sensores segue o princípio da divisão de tensão.

Figura 29 – Divisão de tensão



Fonte: Lopes (2019, p.105).

### 3.2.1 Atividade Sensor

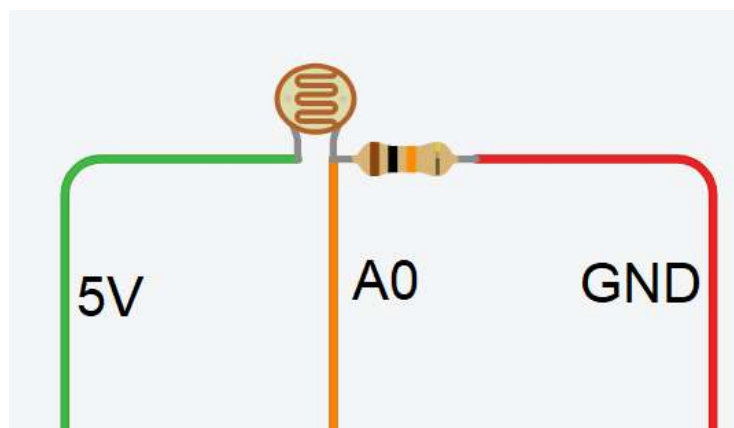
Para esta atividade será necessário

- Placa Arduino
- Computador com IDE do Arduino
- LED
- LDR (Sensor de Luminosidade)
- 1 resistor de 10 kΩ (10.000 Ohms)
- 1 resistor de 220 Ohms
- *Jumpers* (conectores)

O LDR é um resistor de resistência variável, mas o que isso significa? Que quanto maior a intensidade luminosa, menor vai ser a resistência.

Primeiramente vamos explorar as conexões do sensor LDR. Como podemos observar na Figura 30:

Figura 30 – Conexões LDR



Fonte: Autor (2021).

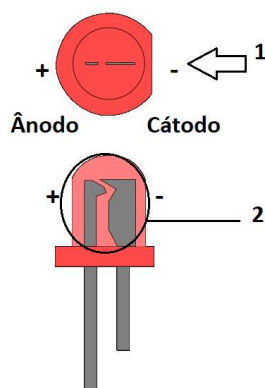
O princípio de funcionamento do sensor acima é o de divisão de tensão, previamente explicado, na qual associamos duas resistências, uma fixa que no nosso caso corresponde a  $10\text{ k}\Omega$  e uma resistência variável com uma determinada grandeza física, que neste caso é o LDR.

Na extremidade da esquerda, conectamos a alimentação de 5 Volts, na extremidade da direita conectamos o fio terra (GND) e no fio do meio medimos a variação da divisão de tensão. O conceito resistências associadas estão interligadas com a medição nos pinos analógicos da placa Arduino, será explorada de forma subsequente ainda neste capítulo.

Utilizaremos o LED como um atuador e o LDR como sensor, ou seja, ao atingir uma determinada intensidade luminosa, o LED vai acender, caso contrário ele vai apagar. As configurações de polo positivo (ânodo) e polo negativo (cátodo) pode ser visto na Figura 31.



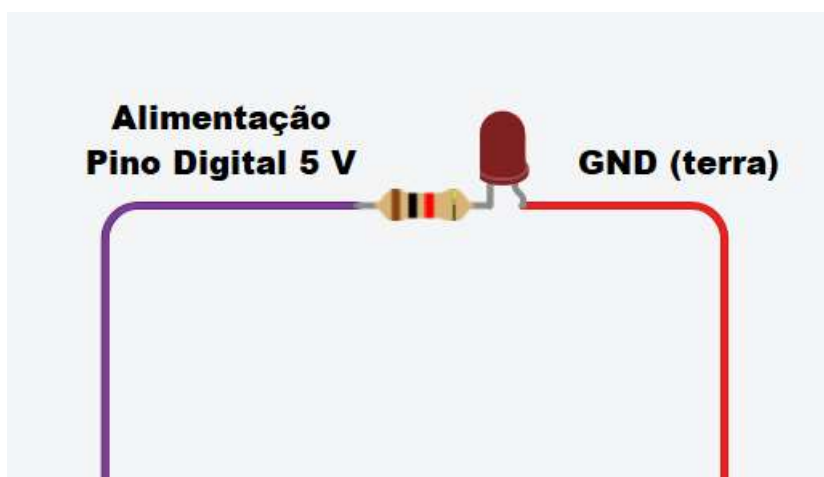
Figura 31 – Ânodo e Cátodo do LED



Fonte: Curto Circuito (2021).

Se tentarmos ligar o LED diretamente na alimentação (5 V) do Arduino direto, ele irá queimar, para isso precisamos fazer uma associação em série do LED com uma resistência de 220 Ohms, conforme pode ser visto na Figura 32.

Figura 32 – Conexão de um LED

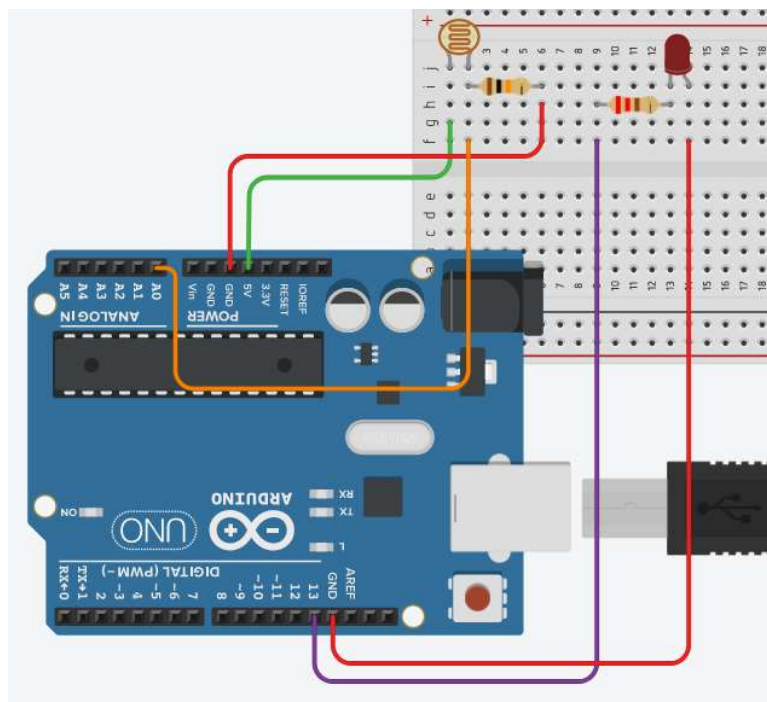


Fonte: Autor (2021).

Podemos observar na figura acima que na extremidade da esquerda será onde iremos fazer a conexão com uma porta digital e a extremidade da direita conectaremos com o fio terra (GND). Utilizaremos a *protoboard* para a construção do circuito, sua utilização foi explorada no capítulo 2.

A construção da automação pode ser vista na Figura 33

Figura 33 – Montagem do Circuito



Fonte: Autor (2021).

Após a montagem do circuito vamos para o processo de abstração, que se refere à explicar os processos lógicos, antes da programação propriamente dita.

As medidas analógicas do Arduino se baseiam na medida de resolução de 10 bits, ou seja,  $2^{10} = 1024$ .

Observe a tabela a seguir:

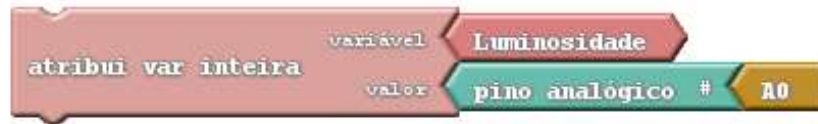
Tabela 1 – Resolução de medida analógica Arduino

Vtagem	Resultado
0 Volts	0
5 Volts	1023

Fonte: Autor (2021).

Iremos nos aprofundar agora na programação. Os valores de luminosidade estão associados a divisão de tensão que tem valores do tipo inteiro (1,2, 3, 4, 5 ...), logo o tipo de variável adotada será do tipo inteira. Temos que substituir o valor da variável pelo valor de leitura do pino analógico A0.

Figura 34 – Operação matemática para a obtenção da temperatura



Fonte: Autor (2021).

A fim de obtermos um retorno do programa sobre a medida de luminosidade utilizaremos o comando imprimir para obtenção desta informação no monitor serial

Figura 35 – Comando de Impressão de resultado



Fonte: Autor (2021).

Partiremos agora para o questionamento referente a estruturas de condição. Utilizaremos uma estrutura do tipo se/senão. Utilizaremos a condição se a luminosidade for superior a 170 o LED será ligado, caso ao contrário, permanecerá desligado. Antes de explicitarmos o código da estrutura de condição vamos observar o funcionamento dos pinos digitais:

Figura 36 – Pinos Digitais



Fonte: Autor (2021).

O número de pinos digitais varia conforme o modelo do Arduino. O modelo que foi utilizado foi o Arduino UNO que possui um total de 14 pinos digitais (do 0 ao 13). Os pinos digitais funcionam como se fossem “interruptores” nos quais podem ter dois estados possíveis: ALTO (Ligado ou um) e BAIXO (Desligado ou zero).

Observe a tabela abaixo.

Tabela 2 – Estados Arduino

Estado	Voltagem	Estado
ALTO	5 Volts	Ligado (ou 1)
BAIXO	0 Volts	Desligado (ou 0)

Fonte: Autor (2021).

Escolheremos um estado ALTO, ou seja, ligado para quando a luminosidade tiver um valor de medida inferior a 170 e um estado desligado caso valor de medida de luminosidade seja superior a 170. Podemos observar o conceito acima na Figura 37:

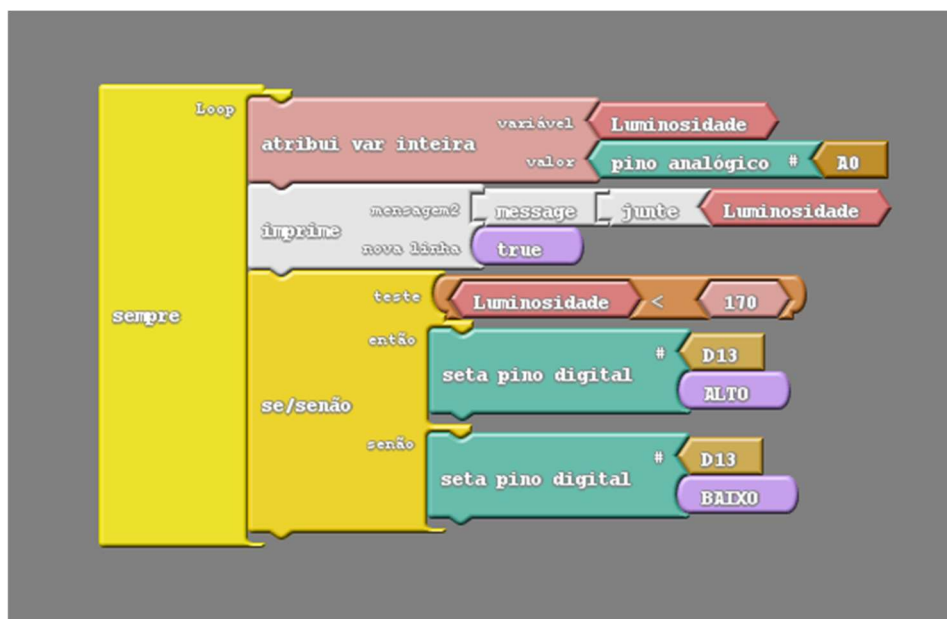
Figura 37 – Estrutura de condição Se/Senão



Fonte: Autor (2021).

A resolução final do problema é união das estruturas com todos os comandos aqui trabalhados. Sendo possível observar a sua solução na Figura 38 :

Figura 38 – Solução da automação envolvendo o sensor de Luminosidade



Fonte: Autor (2021).

O princípio que foi explorado aqui, é o mesmo do poste elétrico, ou seja, caso a variável de luminosidade diminua, o LED acende, caso o valor de luminosidade aumente, o LED apaga.

## Capítulo 4 – Correção de um código e Construção de um Projeto (Sistema de irrigação automático de uma horta escolar) Parte I

No decorrer deste capítulo teremos duas atividades, a correção de um código de programação e sua reestruturação.

Com base nos interesses dos estudantes, a construção inicial de uma automação na forma de um projeto, sendo este, a construção de um sistema de irrigação automatizado para horta escolar,

### 4.1 Atividade de correção do código

Para esta atividade será necessário:

- Placa Arduino
- Computador com IDE do Arduino
- LED
- LM35 (Sensor de temperatura)
- *Protoboard*
- *Jumpers* (conectores)

Utilizaremos no decorrer desta atividade o sensor LM35. Este sensor é um sensor de temperatura, sendo este, um circuito integrado. Foi escolhido o LM35 como um dos três sensores a ser trabalhados no decorrer deste curso. Inicialmente foi mostrado aos alunos o monitor serial de uma programação previamente feita do LM35, cujos valores variavam de 46°C a 53°C em um ambiente fechado sendo perguntado; “Faz sentido aquela medida”?

Figura 39 – Comparação da variação da temperatura

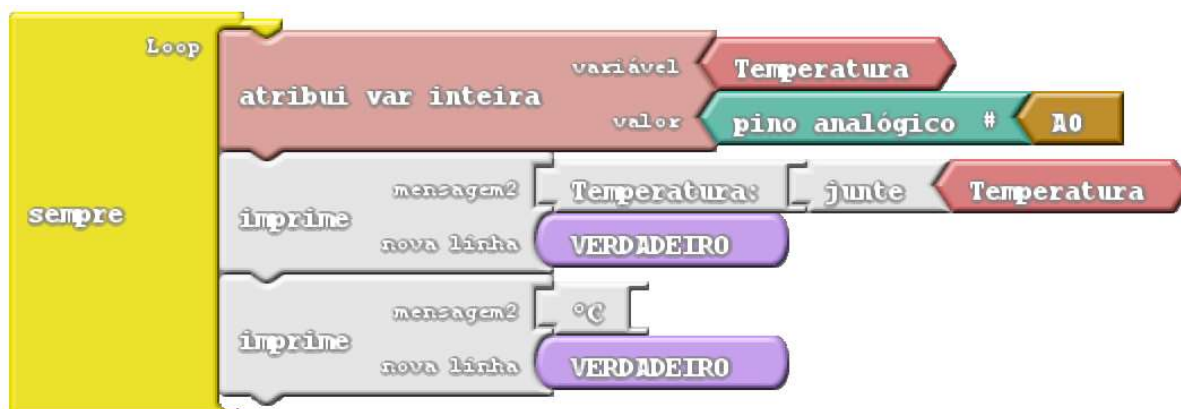
COM3	COM3
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 52 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 47 °C
temperatura: 52 °C	Temperatura: 46 °C
temperatura: 52 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 53 °C	Temperatura: 46 °C
temperatura: 52 °C	Temperatura: 46 °C
temperatura: 52 °C	Temperatura: 46 °C
temperatura: 52 °C	Temperatura: 46 °C
temperatura: 53	Temperatura: 46 °C

Auto-rolagem  Show times
  Auto-rolagem  Show time

Fonte: Autor (2021).

O objetivo aqui, apesar de ser um dia quente, é demonstrar a estranheza da medida, pois é um valor muito elevado. O código utilizado nesta atividade pode ser visto na Figura 40:

Figura 40 – Código incorreto para medida de temperatura

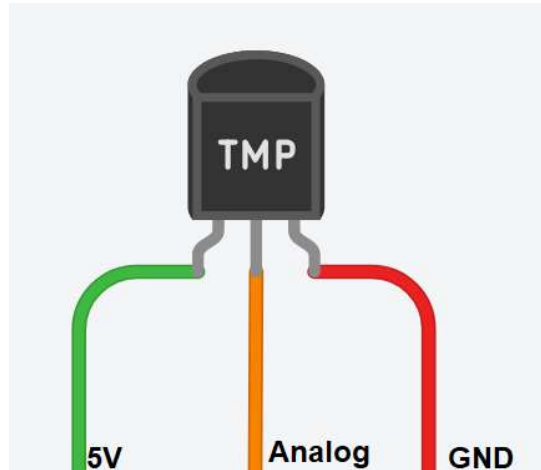


Fonte: Autor (2021).

O primeiro questionamento para a solução do código é: “Temperatura pode ser uma variável do tipo decimal?”, ou seja, temperatura pode ser um número fracionado? Para exemplificar isso, foi utilizado um termômetro para medida de temperatura corporal, obtendo um resultado de 36,5°C. Os estudantes chegam à conclusão que sim, temperatura pode sim ser um valor decimal. Para podermos continuar a correção do código, precisamos primeiro compreender como o sensor LM35 funciona.

O LM35 é um circuito integrado que tem como objetivo principal a aferição de temperatura. As conexões de seus terminais estão ilustradas na Figura 41.

Figura 41 –Conexões do LM35



Fonte: LOPES 2019.

Na extremidade da esquerda na cor verde temos o terminal responsável pela entrada de 5 Volts. O pino da extremidade da direita em vermelho é o terminal GND (fio terra). O pino do meio em laranja é o pino analógico, onde obteremos informações para podermos aferir a temperatura.

Precisamos também saber sobre o funcionamento do LM35. De acordo com o fabricante o LM35 apresenta uma variação 10 mV (mili volts) a cada 1°C de variação de temperatura. Deste modo, precisamos realizar a seguinte operação matemática para obtenção da temperatura:

$$5 \text{ Volts} - 1023$$

$$0,01 \times T - A0$$

Obtemos então como resposta:

$$T = \frac{5 \times A0}{10,23}$$



Para obtermos os valores de uma medida analógica no caso a variável A0 acima, temos que utilizar o seguinte comando no Ardublock:

Figura 42 – Medida Analógica



Fonte: Autor (2021).

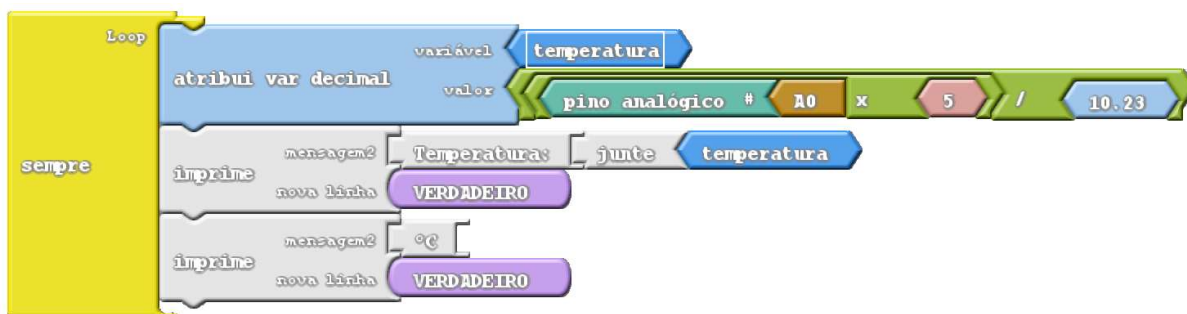
Figura 43 – Armazenamento e cálculo de medida digital



Fonte: Autor (2021)

A solução final do código é a seguinte:

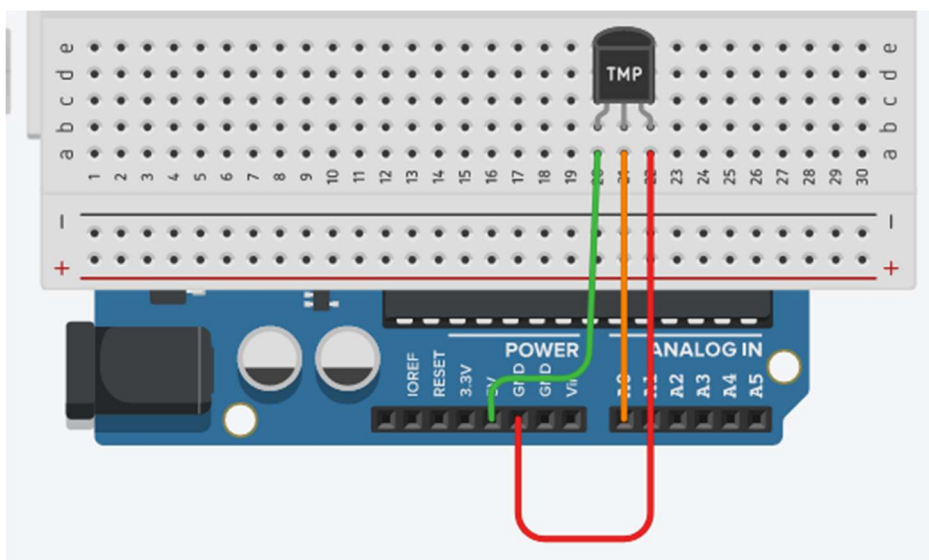
Figura 43 – Programa de captura de temperatura



Fonte: Autor (2021).

Sendo a construção do circuito feita da seguinte maneira:

Figura 44 – Circuito da atividade



Fonte: Autor (2021).

## 4.2 Construção de um Projeto de Irrigação autônoma

Material que utilizaremos:

- Arduino Uno;
- Computador com IDE do Arduino;
- Higrômetro (sonda/módulo);
- *Jumpers* (conectores).

De acordo com a atividade do último capítulo, os alunos tinham que escolher um projeto e se interessaram pela construção de um sistema de irrigação automatizado. Isso ocorreu porque já haviam realizado a montagem de uma horta no meio escolar como projeto para a feira de ciências da Unipampa - Campus Bagé. No entanto, desejavam a construção de um sistema de irrigação automatizado para que a horta pudesse ser mantida mesmo quando não estivessem na escola, como durante as férias escolares. Tendo em vista o desejo dos alunos pelo projeto, foram escolhidos dois sensores para trabalhar: o LDR, que mede a intensidade luminosa, e o Higrômetro, que mede a umidade do solo.

É necessário analisar os motivos pelos quais queremos coletar informações sobre a umidade do solo. A razão é que só faz sentido ligarmos um possível sistema de irrigação caso a terra esteja seca; caso esteja úmida, não há motivo para irrigarmos uma horta. Três perguntas podem ser feitas:

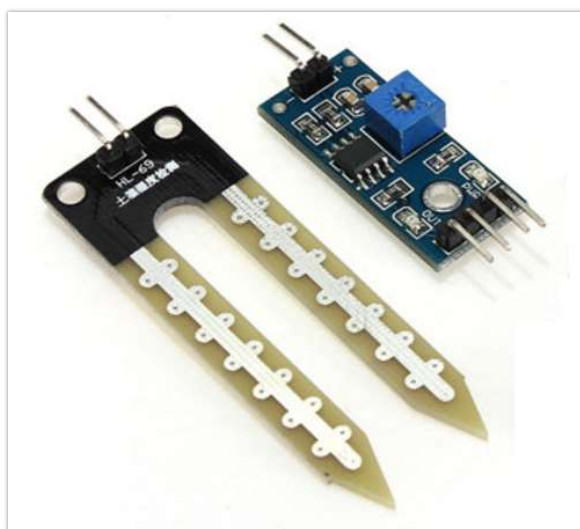
Quais são os dados? **Resposta:** Umidade do solo

Quais ações? **Resposta:** Ligar e desligar o sistema de irrigação.

Quais os processos? **Resposta:** Se atingir determinado valor o sistema liga, caso ao contrário, fica desligado. Sendo como sugestão a construção de um fluxograma.

Nesta atividade será explorado a utilização de único sensor. O sensor escolhido foi o Higrômetro (Umidade da terra).

Figura 45 – Sensor de umidade da terra

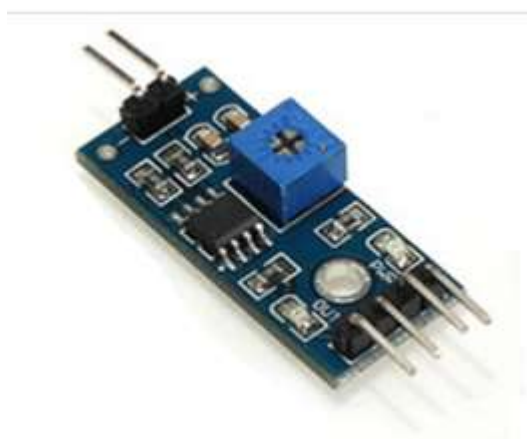


Fonte: Autor (2021).

O sensor de umidade da terra se divide em duas partes: O módulo e a sonda.

O módulo possui as seguintes características:

Figura 46 – Características do módulo do higrômetro



Descrição:

- Contém um chip comparador LM393.
- Contém um potenciômetro para ajustar a sensibilidade.
- Possui 4 pinos que ligam no Arduino:
- GND (terra)
- Alimentação (5 volts ou 3.3 volts)
- Pino D0
- 1 Pino de Leitura Analógica

Fonte: Autor (2021).

As características da sonda são:

Figura 47 – Características da sonda do higrômetro



A sonda tem dois terminais

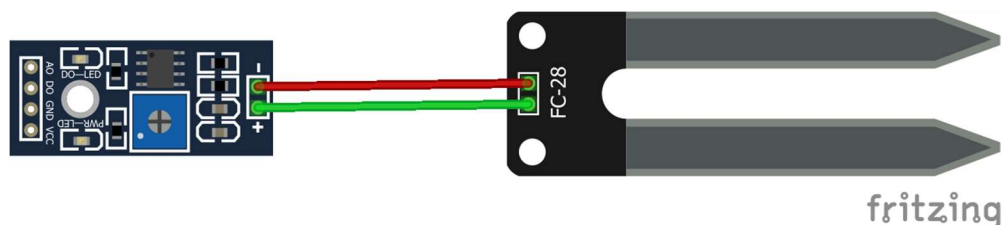
- Um pino de Alimentação de 5 volts
- Um pino GND

Fonte: Autor (2021).

O higrômetro é um sensor de umidade do solo e funciona com dois eletrodos para conduzir uma corrente elétrica pelo solo, fazendo a leitura de umidade relativa é feita por comparação com uma resistência. A água aumenta a condutividade elétrica, enquanto o solo seco conduz com mais dificuldade.

Vamos iniciar com as conexões, primeira do módulo com a sonda:

Figura 48 – Conectando a sonda ao módulo

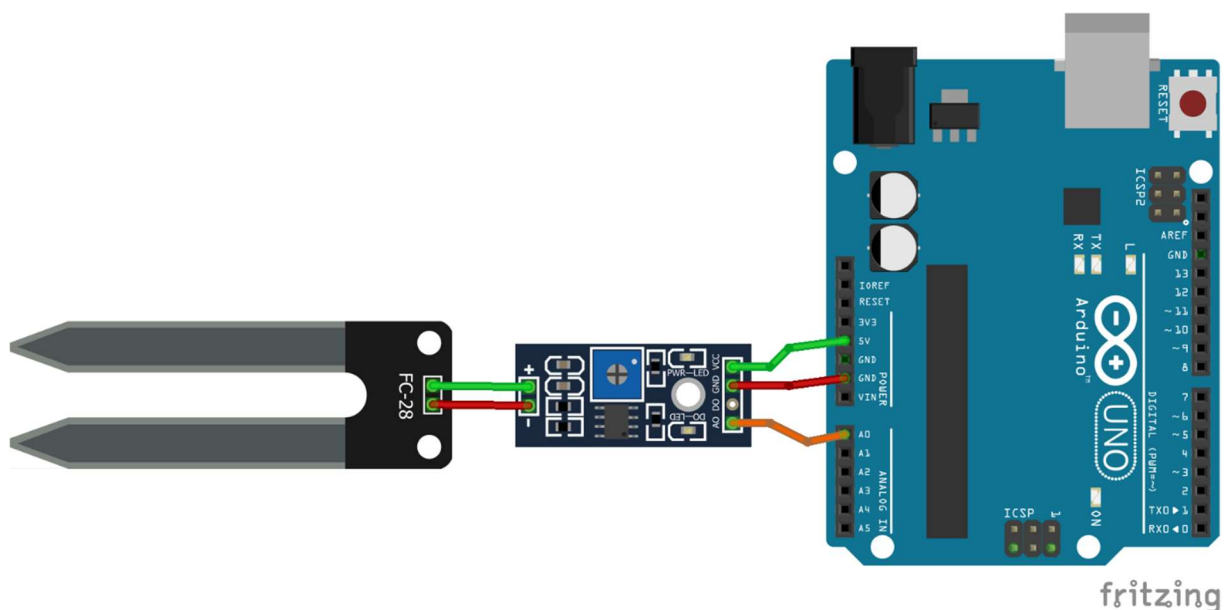


Fonte: Autor (2021).

Como podemos ver na figura acima, o fio verde corresponde a alimentação e o fio vermelho terra do módulo.

Conectaremos agora o módulo ao Arduino:

Figura 49 – Conexões do higrômetro ao Arduino



Fonte: Autor (2021).

O pino analógico (em laranja) do módulo será conectado na porta A0 do Arduino, o pino GND do módulo (em vermelho) ao GND da placa e por último o VCC da placa (em verde) será conectado na porta 5 Volts do Arduino.

Após a montagem do circuito, partiremos para a programação no Ardublock. O funcionamento do higrômetro é exatamente igual aos sensores já explorados aqui, ou seja, segue o princípio da divisão de tensão.

Figura 50 – Programação do Higrômetro

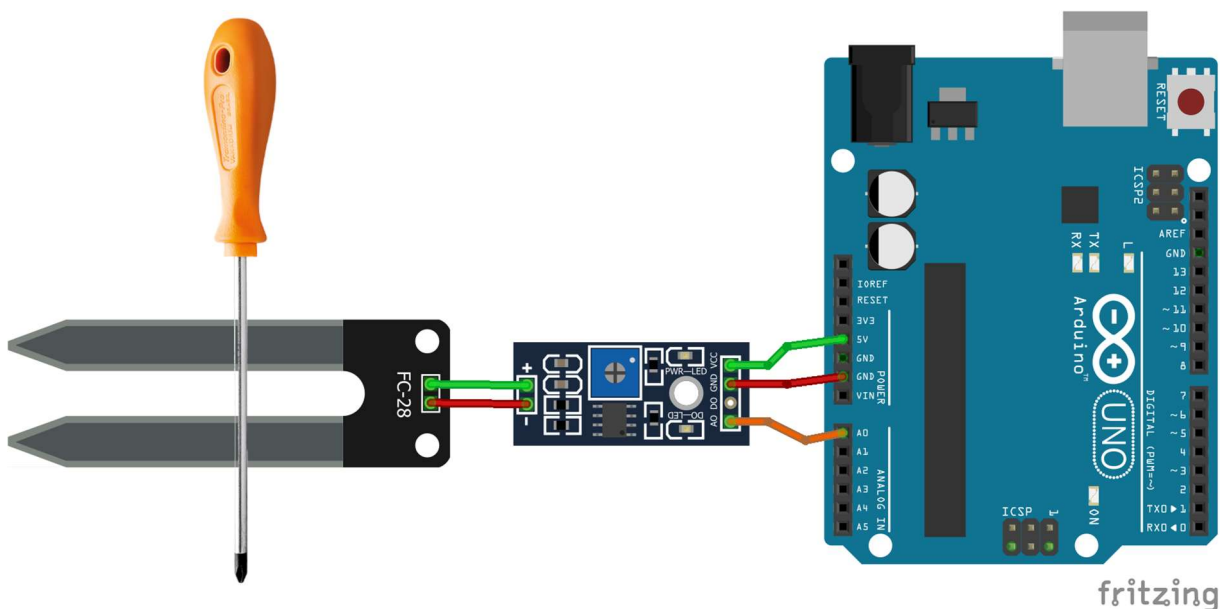


Fonte: Autor (2021).

Coletamos a informação da porta A0 e armazenamos numa variável que chamaremos de “Solo” e imprimiremos ao usuário, o valor referente a umidade relativa do solo. Ao molharmos o solo, o valor de medida diminui, caso o solo volte a ficar seco, o valor de medida aumenta. Com a finalidade de demonstrar o funcionamento

do sensor e da condutividade elétrica, se coloca entre os terminais da sonda uma chave de Philips como pode ser visto na imagem abaixo:

Figura 51 – Funcionamento do sensor (condutividade)



Fonte: Autor (2021).

Ao realizar o procedimento acima, o valor de leitura corresponderá a 0. É salientado aos alunos que, caso o valor tenha um valor tão elevado igual a 0, pode significar que o sensor está posicionado sobre um material condutor. Para evitar erros na medida, foi sugerido aos estudantes que isolassem os terminais de conexão da sonda com “cola quente”, pois o mínimo contato com um material condutor faria com que o valor fosse a medida mínimo de 0.

## Capítulo 5 – Construção de um sistema de irrigação automatizado (Parte II)

O objetivo central deste capítulo é auxiliar os estudantes na utilização de dois sensores ao mesmo tempo.

### 5.1 Utilização de dois sensores

Material para esta atividade:

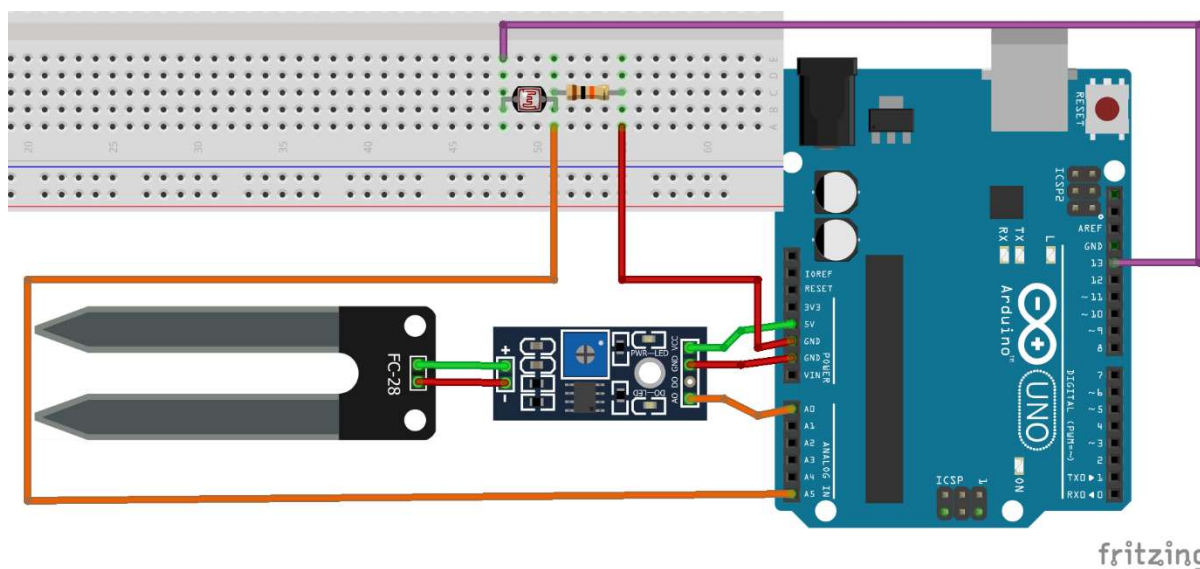
- Placa Arduino Uno
- Computador com Arduino IDE
- LDR (Sensor de Luminosidade)
- 1 resistor de 10 k $\Omega$  (10.000 Ohms)
- 1 Sensor de umidade do solo (higrômetro)
- *Protoboard*
- *Jumpers* (conectores)

O sentido da utilização dos sensores de luminosidade e umidade do solo, se deve pois os estudantes informaram que, de acordo com suas pesquisas, para um sistema automatizado de irrigação, a luminosidade do sol não pode estar muito elevada, uma vez que, isso levaria ao comprometimento da horta, assim como, caso o solo já estivesse úmido, não faria sentido ligar o sistema de irrigação.

Tendo em vista que ambos os sensores já foram utilizados anteriormente, a montagem do circuito pode ser feita pelos estudantes. Lembrando apenas de que a conexão do LDR seja feita por uma porta digital, no caso, a porta digital 13.

As conexões dos dois sensores podem ser vistas na imagem na Figura 52:

Figura 52– Conexão dos dois sensores



Fonte: Autor (2021).

Note que as conexões exploradas aqui são exatamente iguais as que foram exploradas nos módulos anteriores, entretanto, o que difere, é que a alimentação do sensor LDR será feita por uma porta digital. A leitura de sua medida de luminosidade se dará na porta analógica A5, poderia ser utilizada qualquer porta desde que a mesma, fosse uma porta analógica, mudando apenas o comando para escolha da porta.

Para podermos utilizar uma porta digital como alimentação devemos selecionar a porta escolhida e mudar seu estado para ALTO.

Figura 53– Como utilizar um pino digital para alimentação.

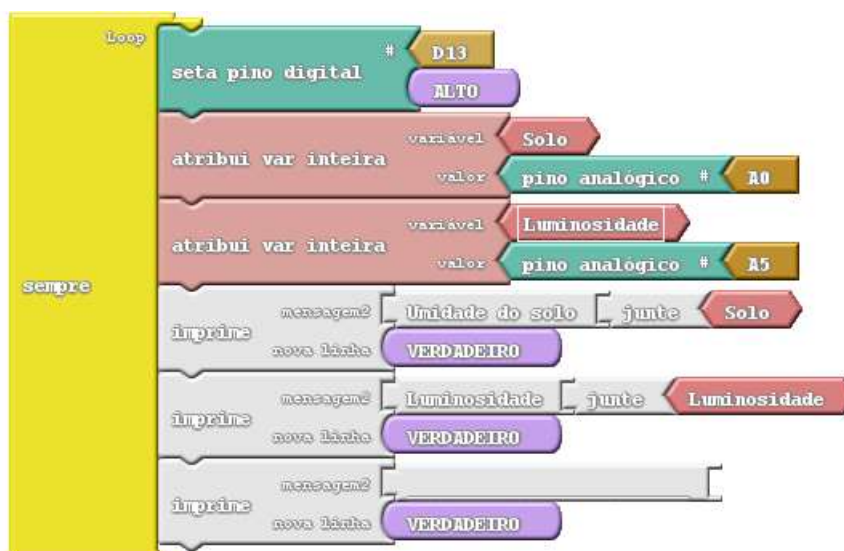


Fonte: Autor (2021).

Após escolhermos a porta digital, vamos para as variáveis. Utilizaremos duas variáveis para armazenar a informação de umidade e luminosidade como mostra a Figura 54:



Figura 54 – Programação



Fonte: Autor (2021).

A impressão das variáveis é um importante, uma vez que, os estudantes precisam determinar os parâmetros iniciais antes da automação final. É necessário coletar informações como: o quão seco ou úmido os valores podem variar ou ainda, quanto a luminosidade pode aumentar ou diminuir no decorrer do dia, procurando o momento mais propício para ligar o sistema de irrigação. Estes parâmetros devem ser coletados e os alunos devem ser orientados a salvar a programação para a última aula (Capítulo 6).

## Capítulo 6 – Sistema de irrigação automatizado Parte Final

Este é o último capítulo da Parte I, sendo o objetivo central capítulo, a demonstração final do sistema de irrigação para a horta escolar. Além dos sensores já explorados, será necessário uma relê como atuador e uma válvula solenoide sendo recomendada uma válvula de 12 Volts.

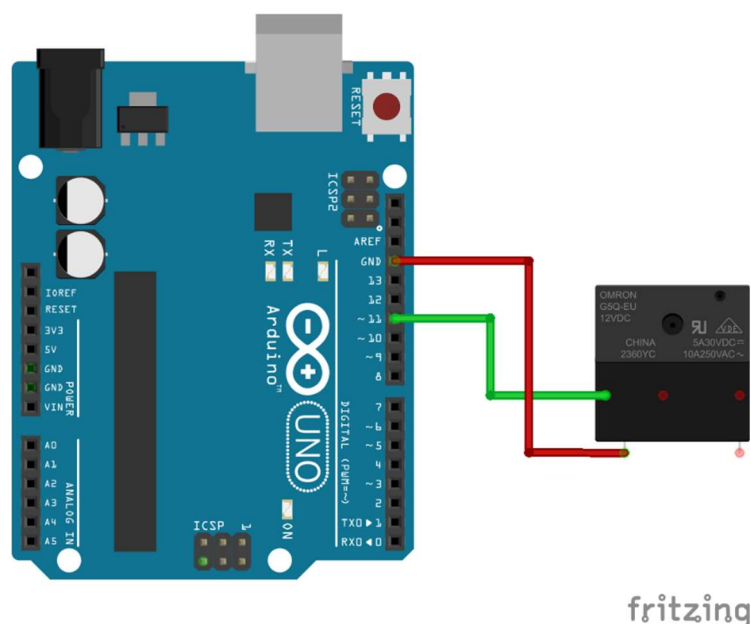
### 6 – Sistema de irrigação

Para esta parte final de nossa automação precisaremos

- Placa Arduino Uno
- Computador com Arduino IDE
- LDR (Sensor de Luminosidade)
- 1 resistor de 10 k $\Omega$  (10.000 Ohms)
- 1 Sensor de umidade do solo (higrômetro)
- 1 relê
- 1 válvula solenoide
- 1 reservatório de água
- Mangueiras
- *Protoboard*
- *Jumpers* (conectores)

Iremos, inicialmente, explorar as conexões da relê que funciona como um tipo de chave, vamos conectar a relê ao Arduino:

Figura 55 – Conexões da relê com o Arduino

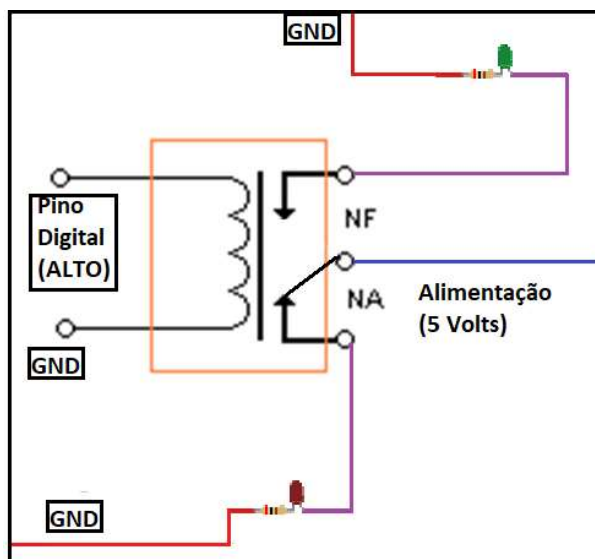


Fonte: Autor (2021).

Vamos conectar o VCC da relê numa porta digital, que no caso, escolhemos a porta digital 12 e o fio terra do relê no GND da placa Arduino. Vamos discutir o funcionamento do relê. De acordo com Lopes (2019):

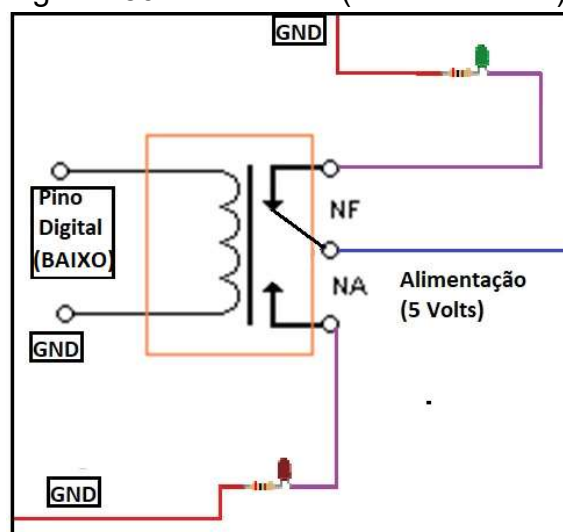
O relê é um dispositivo que funciona como uma chave, sendo um atuador e tendo uma bobina que pode ligar uma chave ou outra. A relê abaixo tem um total de 5 terminais (indicado pelas bolinhas) temos dois terminais para ligarmos ou desligarmos a bobina, um no pino digital e outro no pino GND. Existe um terminal que fica na chave e os outros dois terminais restantes ficam ligados nos estados NA e NF (que serão explicados a seguir). Ao ligar o estado alto a chave NA (Normalmente aberto) de fecha. Caso o pino digital esteja no estado baixo, a chave da relê permanece na posição NF (Normalmente Fechado). Imagine que você tem dois LED's, um verde e um vermelho, e quer deixar um deles ligado em quanto o outro fica desligado, temos os dois pinos para ligar ou desligar a bobina. Ao definirmos um estado ALTO [Figura 56.a] deixamos a bobina ligada e a posição que era NA fica fechada, fazendo com que, LED vermelho acenda. Se definirmos o estado um estado BAIXO [Figura 56.b], a bobina se desliga e a chave vai para o seu estado normal (NF), fazendo com que, o LED verde acenda. (Lopes, 2019, p. 111).

Figura 56.a – Relê (Estado Alto)



Fonte: LOPES (2019).

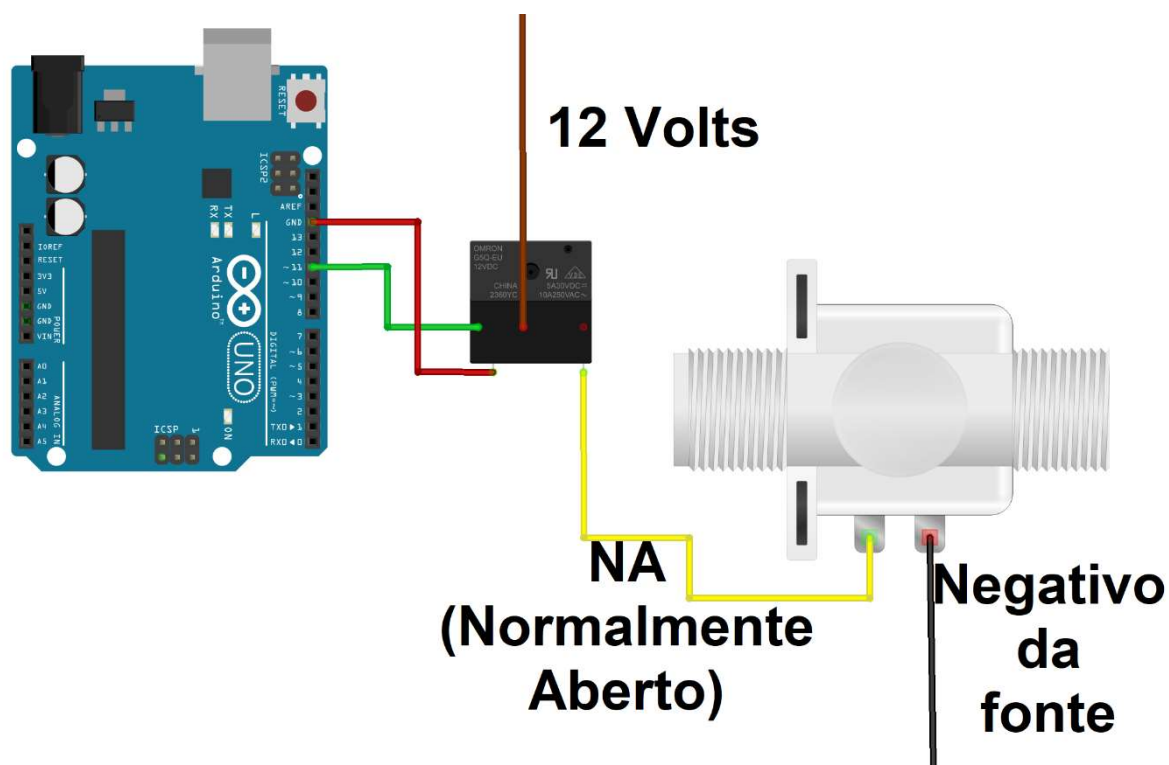
Figura 56.b – Relê (Estado Baixo)



Fonte: LOPES (2019)

Agora que explicamos como funciona um relê, iremos explorar sua conexão com a válvula solenoide:

Figura 57 – Conexões da relê com a válvula solenoide

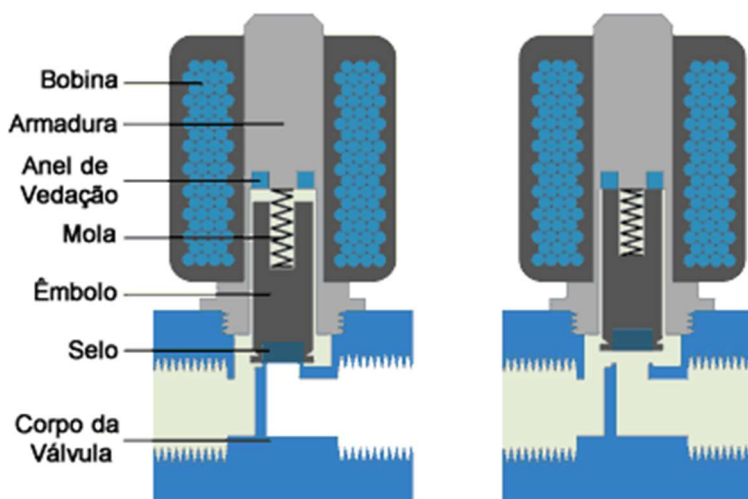


Fonte: Autor (2021).

Conectaremos o positivo da fonte em nossa relê (em marrom) e o negativo da fonte no terminal ao polo negativo da válvula solenoide (em preto). Ligamos o polo referente ao positivo da válvula no pino normalmente aberto da relê.

Após compreendermos o funcionamento do relê, vamos tentar compreender como funciona a válvula solenoide. Observe a figura abaixo:

Figura 58 – Válvula Solenoide

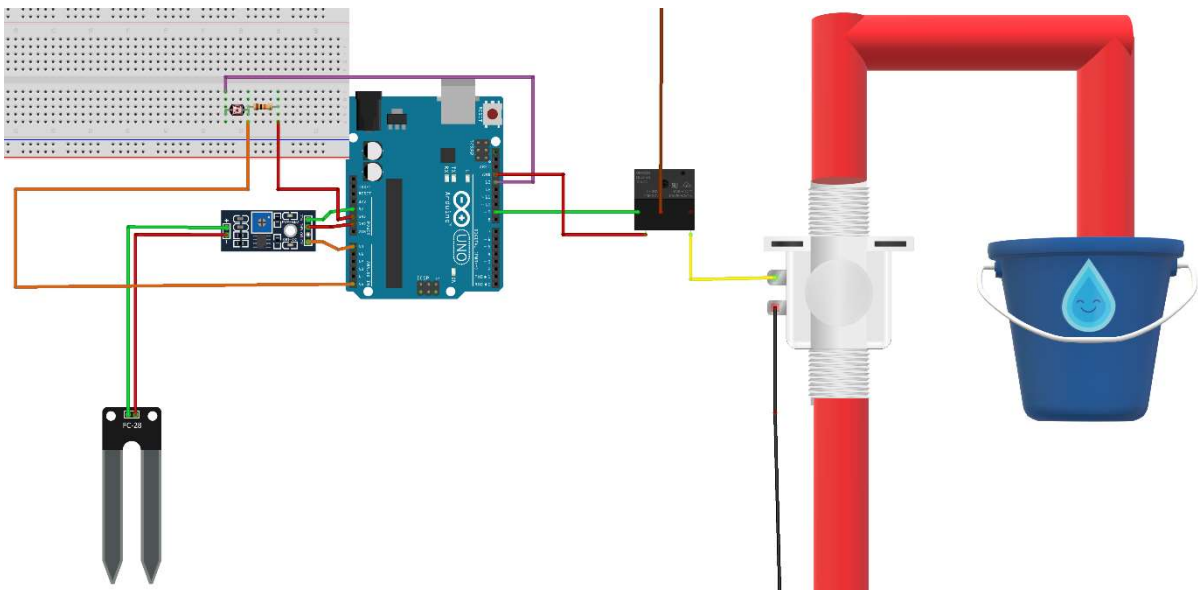


Fonte: Retirado de <https://casadasvalvulasmg.com.br/wp-content/uploads/2021/12/valvula-solenoide-1.png> Acesso em 2021.

A válvula solenoide é controlada por uma corrente elétrica, que ao atravessar uma bobina elétrica, que ao ser acionada, faz com que um êmbolo móvel se mova para cima, permitindo o fluxo de água de um ponto A para um ponto B.

A montagem final do projeto integrando os sensores e atuadores fica da seguinte maneira:

Figura 59 – Sistema de irrigação



Fonte: Autor (2021).

A construção do sistema pode ser vista acima, o sensor de luminosidade e higrômetro ficam continuamente coletando informações de luminosidade e umidade do solo e caso atinja os parâmetros estabelecidos, acionará a relê que ligará a válvula solenoide começando a irrigar a horta liberando água do reservatório. Após o solo estar úmido, o relê irá para o estado normalmente fechado, desligando o sistema de irrigação.

Agora partiremos para a programação necessária para o funcionamento do sistema de irrigação. Para isso, utiliza-se a programação explorado no ultimo capítulo. Observando a Figura 60 relembra a programação:

Figura 60 – Programação



Fonte: Autor (2023)

Ao coletarmos informações referente a luminosidade e umidade do solo, obtemos que caso a luminosidade caia de 170 e a umidade do solo caia para valores abaixo de 550, o sistema deverá ser ativado. Ressaltamos que esses valores podem variar de acordo com o sensor e condições de luminosidade, por isso é importante que você faça as medidas de acordo com seus materiais disponíveis. Precisaremos utilizar um operador do tipo “E” no Ardublock conforme pode ser visto abaixo:

Figura 61 – Operador lógico



Fonte: Autor (2021).

Note que que na estrutura acima, ambas as condições precisam ser satisfeitas para que seja realizada uma determinada ação, que neste caso é o acionamento do relê, caso contrário o relê não será acionada.

Para isso, teremos que utilizar uma estrutura de condição do tipo se/senão, conforme pode ser visto abaixo:

Figura 62 – Estrutura de condição

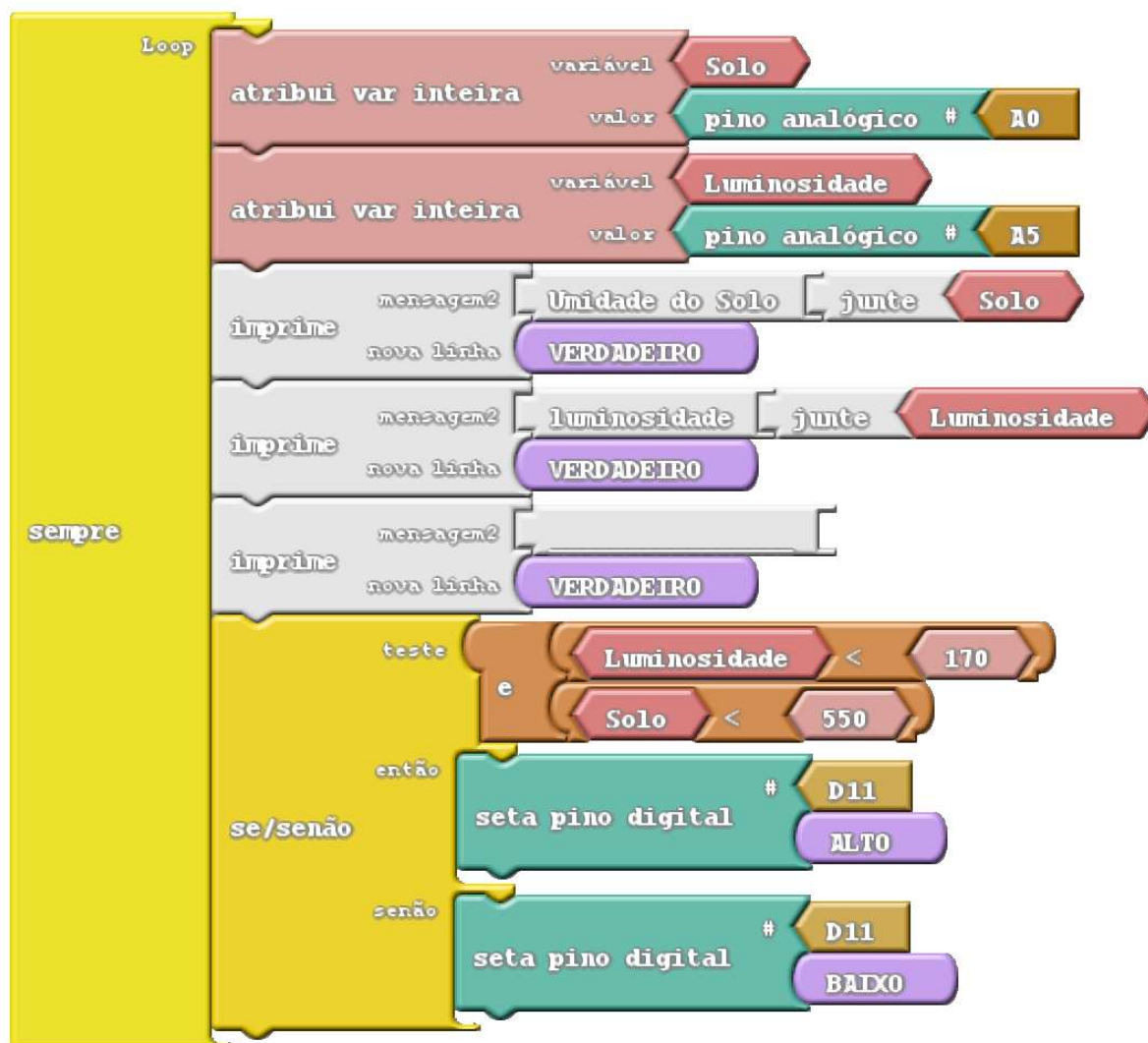


Fonte: Autor (2021).

Caso as condições sejam satisfeitas, o relê presente na porta digital 11 será acionada, estando no estado normalmente fechado, caso contrário, o relê será “desligado” indo para o estado normalmente aberto.

A programação no final ficará da seguinte maneira:

Figura 63 – Programação do sistema de irrigação



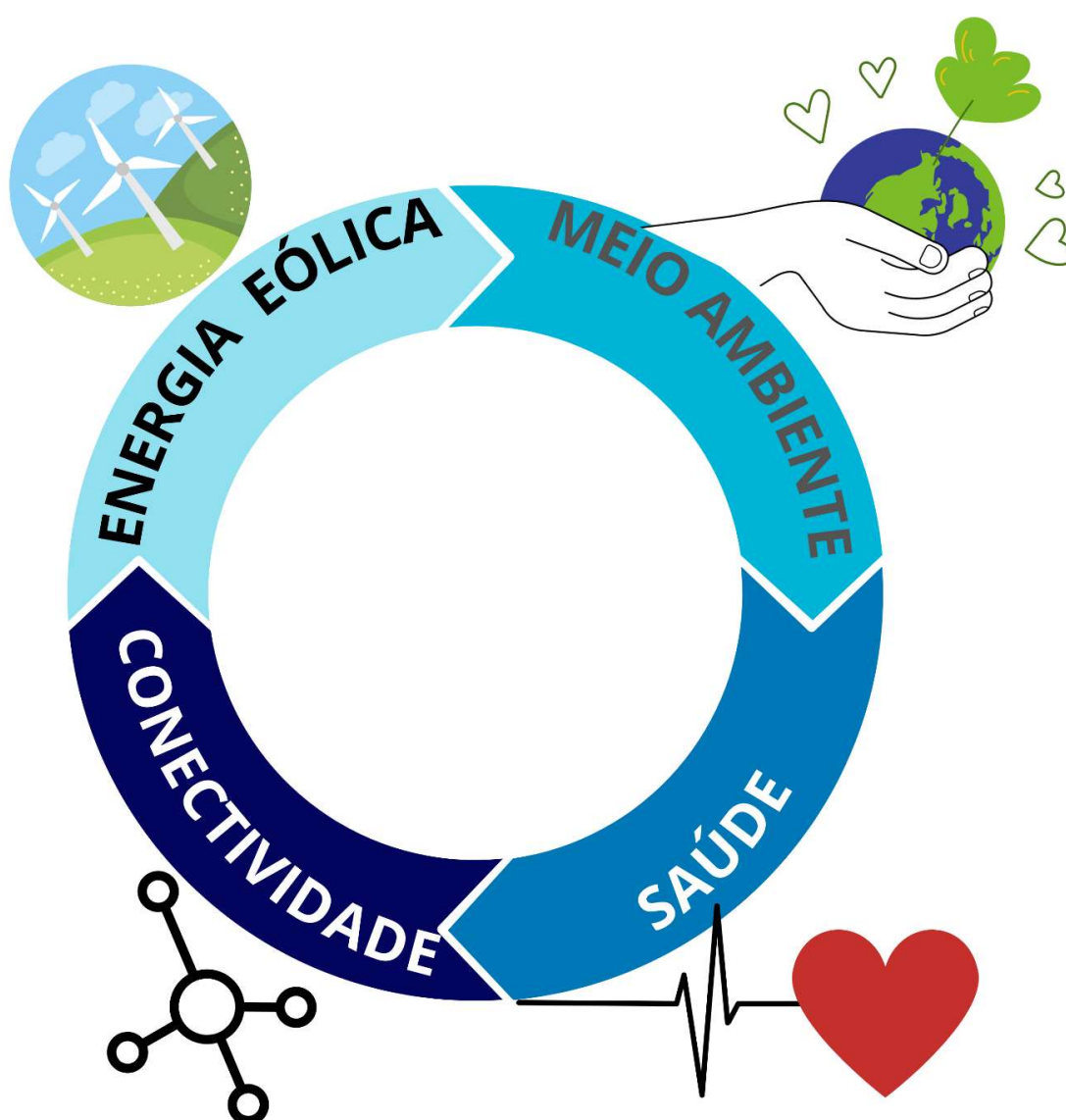
Fonte: Autor (2021).



## Parte II – Tecnologia aplicada em outros contextos

No decorrer desta parte do ebook, dividiremos em quatro capítulos focados no uso de tecnologia na: energia eólica, meio ambiente, saúde e conectividade. O objetivo desta parte é apresentar a aplicação da tecnologia em diferentes áreas do conhecimento e como ela pode servir para tornar a vida melhor.

Figura 64 – Capa referente a Parte II



Fonte: Autor utilizando Canva pro (2023).

## Capítulo 7– Energia Eólica

Primeiramente, vamos contextualizar a razão da utilização da fonte de energia eólica. A escola na qual foi aplicada este projeto está localizada muito próxima de uma usina que utiliza como base o carvão mineral como combustível, o que desperta o interesse dos estudantes sobre o tema.

É necessário destacar inicialmente que a energia a base de carvão em si não é um problema, o objetivo central de uma sociedade moderna tem que ser a diversificação de sua matriz energética, ou seja, ter proporcionalmente diferentes tipos de energia distribuídas de forma igualitária. Os impactos de uma usina de carvão podem ser minimizados se, por exemplo, existirem incentivos para florestamento para que o impacto das emissões de dióxido de carbono (CO<sup>2</sup>) pela usina sejam neutralizadas.

Para podermos encerrar a discussão sobre o uso de uma usina de carvão e podermos seguir para a utilização de energia eólica, é necessário primeiro, explicar o funcionamento da energia de uso residencial.

Um dos maiores desafios da humanidade é armazenar energia de forma eficiente, o ser humano não descobriu ainda como fazer isso, sendo um dos maiores desafios da ciência atualmente, tanto que a energia que estamos consumindo agora em nossas lâmpadas, eletrônicos e etc, está sendo produzido neste exato momento. A corrente que utilizamos é do tipo alternada com uma frequência de 60 Hertz, ou seja, em um segundo a luz variou sua corrente 60 vezes a sua corrente, o que quer dizer que uma lâmpada acesa oscila o tempo todo, mas nossa visão não permite que tenhamos essa percepção.<sup>4</sup>

A discussão sobre o uso de energias apenas do tipo renovável não faz sentido, a luz proveniente da luz solar estar suscetível a irradiação nos painéis solares e de noite a energia não é produzida, assim como hidrelétricas estão suscetíveis a ter sua capacidade de funcionamento reduzida em épocas de estiagem, a energia eólica depende da força do vento para sua produção, por isso se torna impossível extinguir por completo o uso de energias de origem fósseis e nucleares sendo uma tarefa desafiadora no mundo moderno que demanda cada vez mais o uso de eletricidade,

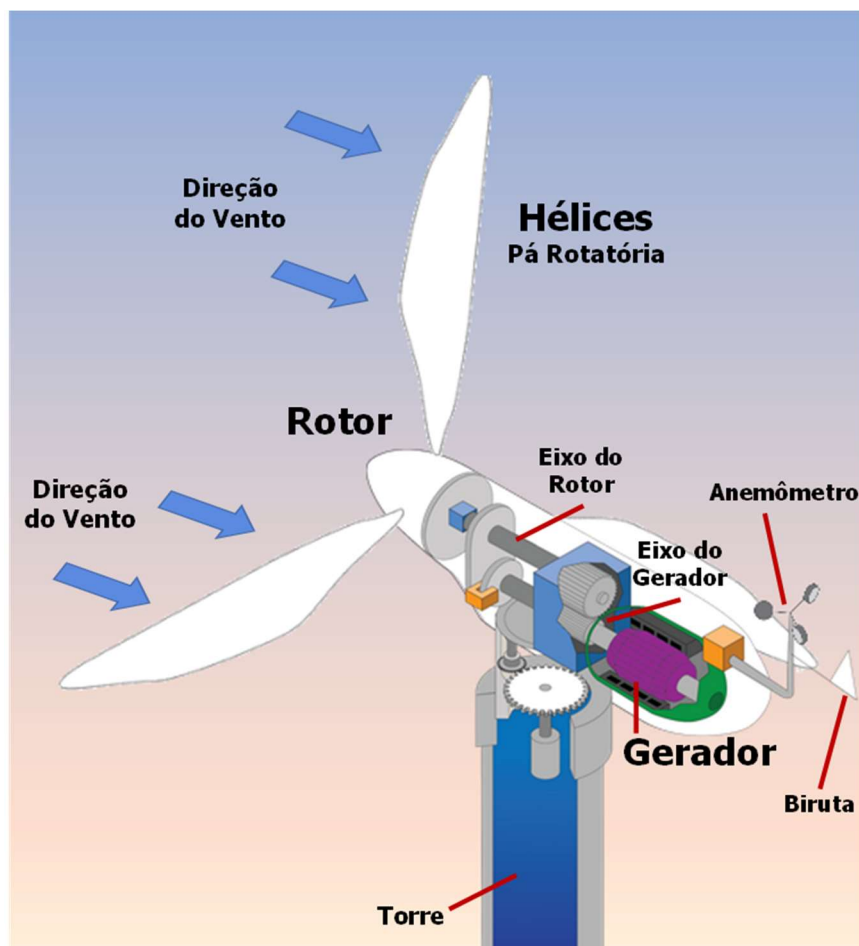
---

<sup>4</sup> Estudo referente a percepção do olho humano:  
[https://www.lcs.poli.usp.br/~gstolfi/mack/Ap2\\_PercepVisual\\_M8.pdf](https://www.lcs.poli.usp.br/~gstolfi/mack/Ap2_PercepVisual_M8.pdf)

entretanto, seus impactos devem ser minimizados como políticas de florestamento, por exemplo.

Sendo feita esta discussão inicial, vamos discutir o funcionamento da energia eólica, começando com o seguinte questionamento, como funciona?

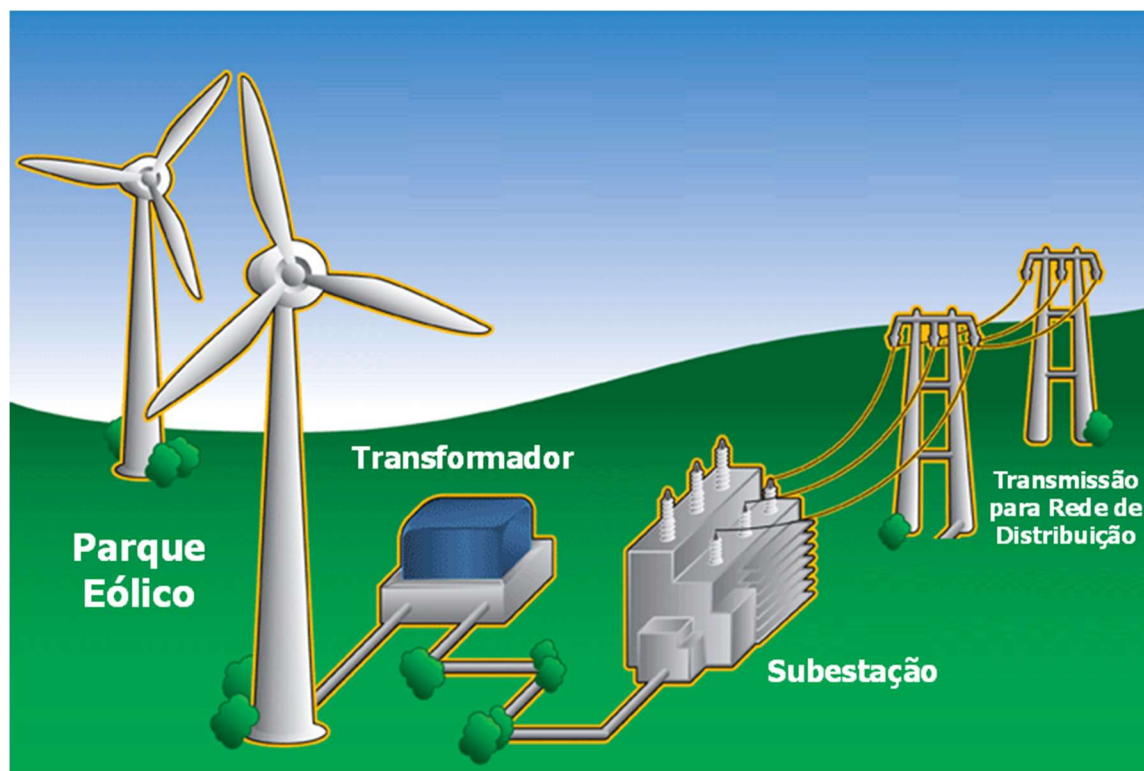
Figura 65 – Funcionamento de um aerogerador



Fonte: Retirado de <https://cbie.com.br/storage/2020/01/voce-sabia-1.png>

O funcionamento de um aerogerador se baseia na transformação de energia do movimento do vento em energia elétrica. A Figura 65 mostra o funcionamento de um aerogerador, no qual o vento faz com as hélices se movimentem, fazendo com que o eixo do rotor comece a rotacionar, ao iniciar o movimento de rotação, é passado por um sistema de redução, aumentando a rotação do eixo do gerador para 1000 rpm (rotações por minutos) produzindo assim eletricidade. O objetivo do anemômetro é medir a velocidade do vento e da biruta a direção do vento, estes são dois sensores acoplados no aerogerador. Podemos ver na figura a seguir como a energia eólica chega em nossas residências.

Figura 66 – Sistema de transmissão



Fonte: Retirado de <https://cbie.com.br/storage/2020/01/voce-sabia-3.png>

Após produzir energia no aerogerador, a energia passa por um transformador que altera para corrente que vai ser utilizada nas linhas de transmissão, porém antes, esta passa por uma subestação de energia.

### 7.1 – Atividade prática

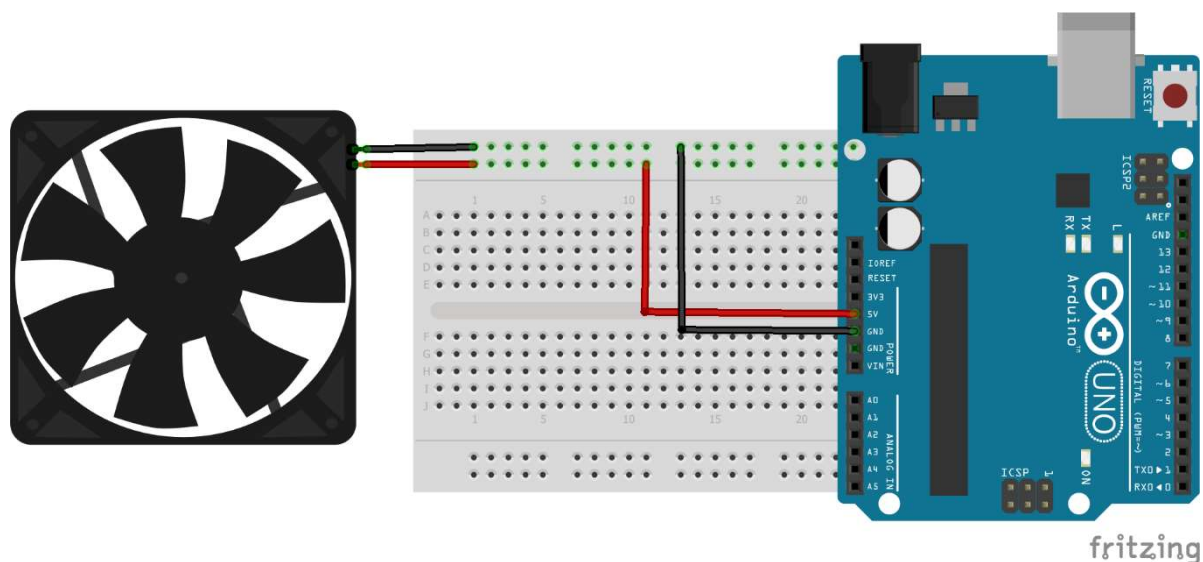
#### Material Utilizado

- Arduino uno
- Secador de cabelo
- Fan de computador
- 1 Led
- *Protoboard*
- Conectores

O Arduino será utilizado apenas como uma fonte de alimentação. Conectamos os terminais que correspondem aos 5 volts e ao GND da fan na *protoboard* e

posteriormente conectamos os 5 volts e o GND da placa na protoboard. Ligamos o Arduino na porta USB. As conexões podem ser vistas na figura a seguir:

Figura 67 – Conexões do Fan



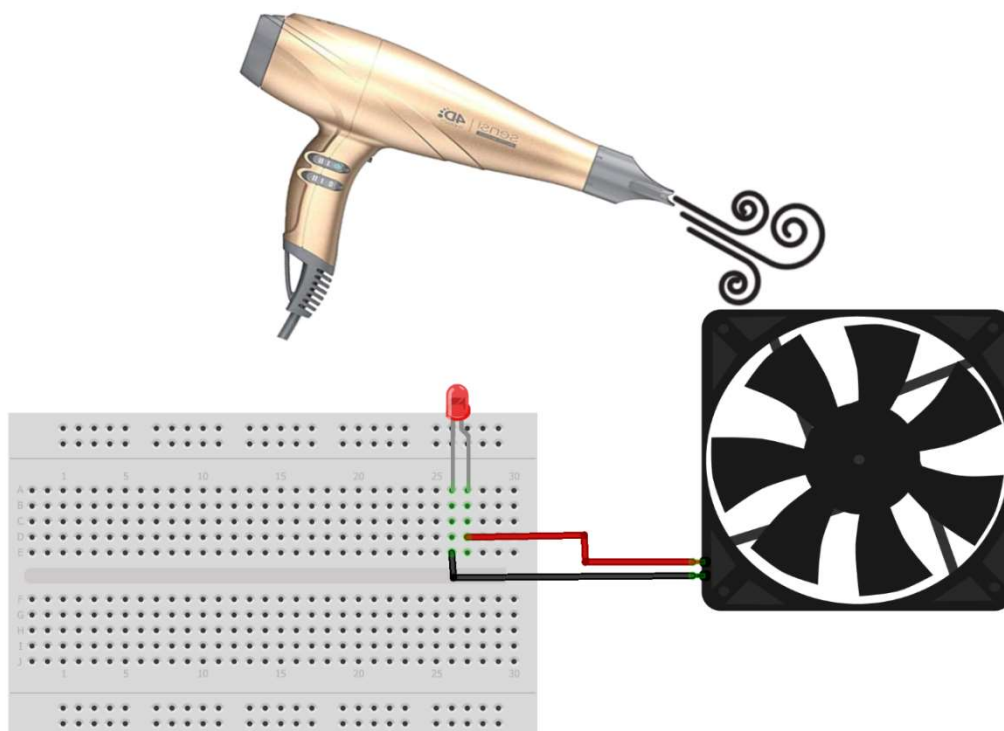
Fonte: Autor (2022).

Ao ligarmos a alimentação sobre o Fan, ele começa a rotacionar, pois o mesmo é um motor de corrente contínua. O funcionamento é exatamente como ocorre dentro de um computador.

Agora ao invés de alimentarmos o Fan com uma diferença de potencial, iremos utilizar um secador de cabelo para mover as hélices do Fan, produzindo assim, eletricidade suficiente para acendermos um LED.

Conectaremos o Fan na *protoboard* e conectaremos o ânodo e cátodo do LED nos terminais referentes a conexão do fio terra e de alimentação. Posteriormente ligamos um secador de cabelo para movimentar as pás como pode ser visto na Figura 68:

Figura 68 – Produção de energia elétrica utilizando um fan de computador



Fonte: Autor (2022).

O objetivo desta aula é a demonstração da produção de energia eólica, utilizando materiais de baixo custo. O LED acende ao ser exemplo.

## Capítulo 8 – Meio Ambiente

A humanidade emite cada vez mais os gases do efeito estufa, o foco deste capítulo é a detecção destes gases com o uso de um sensor de gás do tipo MQ-2, juntamente com o Arduino, contextualizando com os impactos destas emissões.

A concentração de gás em um determinado ambiente geralmente é medida em ppm (partes por milhão) ou ppb (parte por bilhão). O sensor do tipo MQ-2 detecta gases causadores do efeito estufa como dióxido de carbono e monóxido de carbono. Também é possível detectar álcool, hidrogênio, butano (fluido de isqueiro), gás natural, gás liquefeito de petróleo (GLP) e propano. Devido aos perigos ao se colocar fogo em um determinado material a escolha aqui será de utilizar um material mais seguro possível, que neste caso será o butano.

### 8.1 Atividade Prática

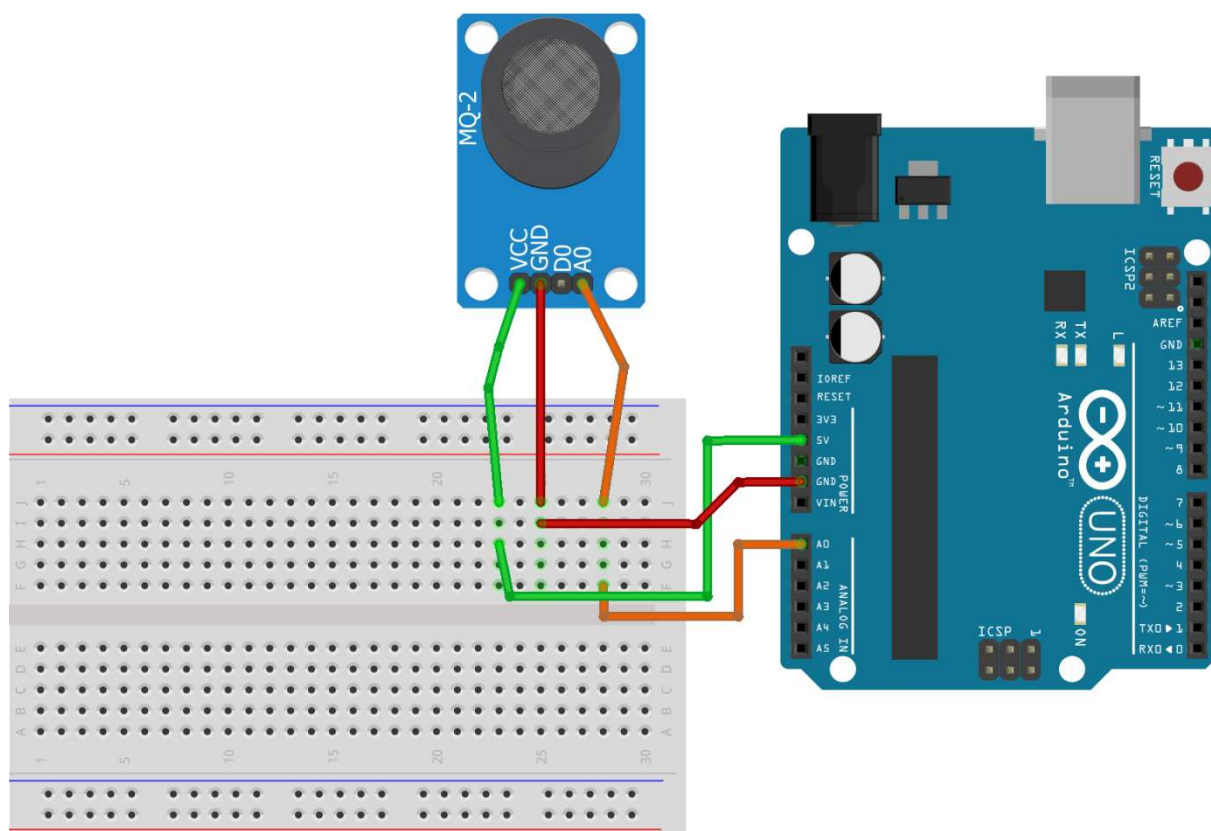
#### Material Utilizado

- Arduino UNO
- Computador com IDE do Arduino
- Sensor MQ-2
- Buzzer
- Isqueiro
- *Protoboard*
- *Jumpers* (conectores)

Vamos inicialmente contextualizar a nossa prática, o objetivo central desta atividade é a criação de um alarme de gás, principalmente dos associados ao efeito estufa. Cidades como *Pequim* na China apresentam forte indícios de dióxido de carbono na atmosfera, sendo as pessoas mais sensíveis como: idosos, crianças, pessoas com o sistema imune comprometido e pessoas com problemas crônicos de ordem respiratória, as mais suscetíveis a enfrentar os efeitos desta poluição.

Vamos inicialmente explorar o uso de nosso sensor, sendo suas conexões:

Figura 69 – Conexões do MQ-2



fritzing

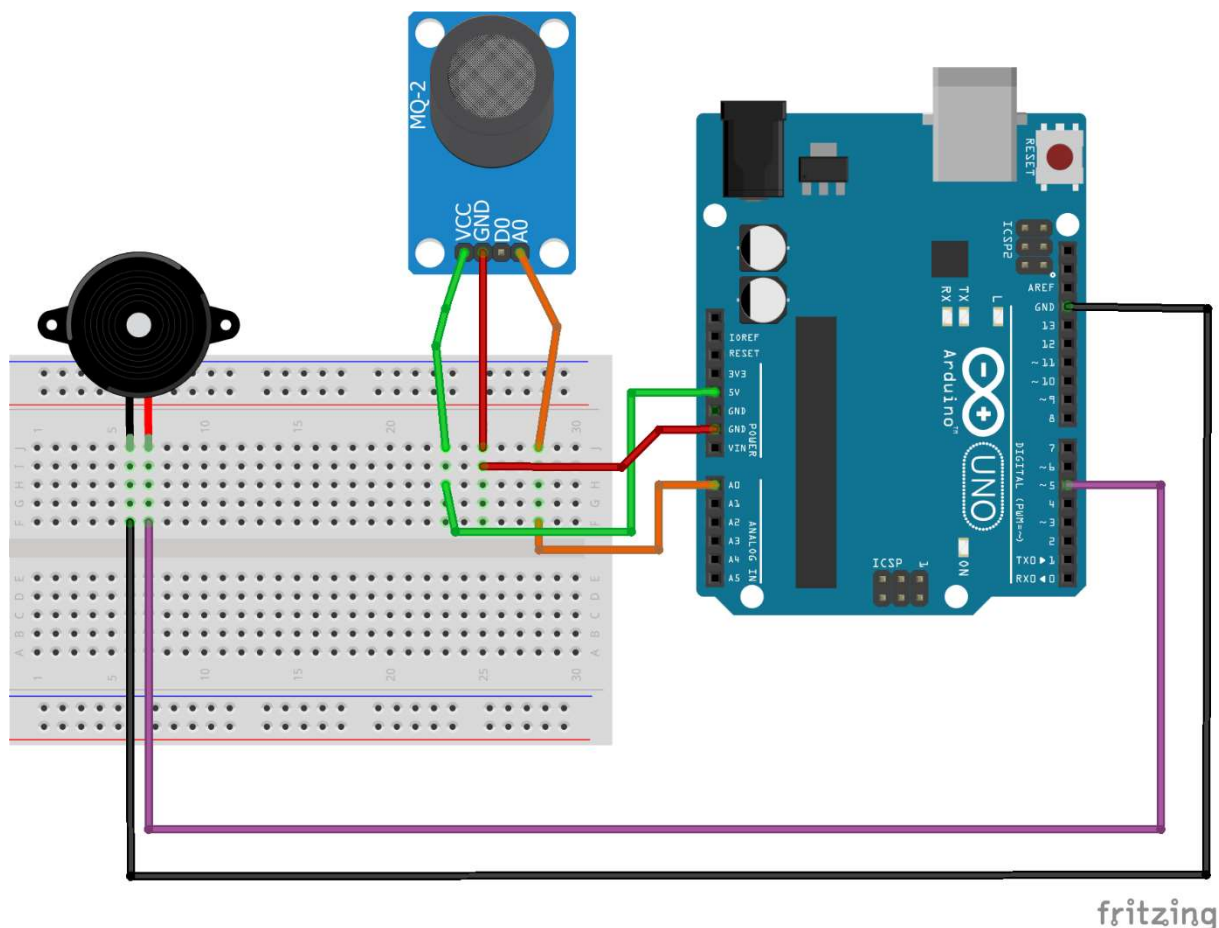
Fonte: Autor (2023).

As conexões são feitas usando a placa *protoboard*, o VCC (em verde) será ligado aos 5 volts da placa, o GND do módulo (em vermelho) ao GND do Arduino e o pino A0 do sensor (em laranja) ao A0 da placa Arduino.

Após a realização das conexões de nosso módulo de gás, faremos a conexão do *buzzer* a nossa placa.



Figura 70 – Conexão completa do circuito



Fonte: Autor (2023).

Conectaremos o positivo do buzzer (em roxo) à porta digital 5 (D5) e o negativo do buzzer (em preto) ao GND da placa.

Partiremos agora para a programação de nossa automação, sendo o primeiro passo a ser tomado, a coleta da medida analógica. Foi adicionado um bloco referente ao sensor de gás no Ardublock modificando o *software*, onde essa programação obtém a informação em porcentagem de gás no sensor. O bloco em questão pode ser visto a seguir:

Figura 71 – Bloco de coleta de informação de gás



Fonte: Autor (2022).

A operação matemática realizada dentro do bloco para obtenção do valor em percentual se dá da seguinte maneira:

$$100\% - 1023$$

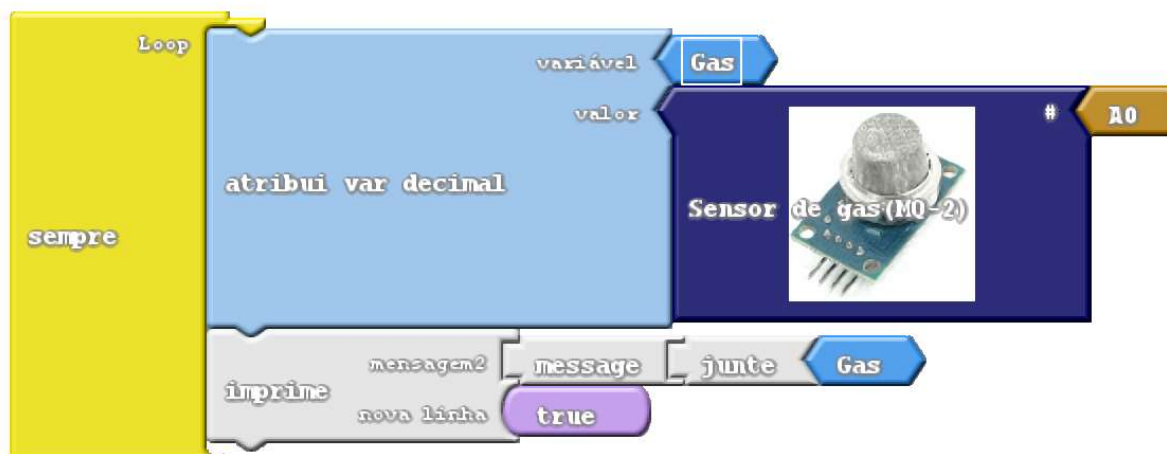
$$P - A0$$

A variável P corresponde o valor em percentual da quantidade de gás e o A0 a medida analógica coletada pela placa. Reorganizando a operação acima, ficamos com:

$$P = \frac{100 \times A0}{1023}$$

Armazenaremos o valor do percentual de gás em uma variável do tipo decimal e imprimiremos o valor desta variável:

Figura 72 – Armazenamento da variável do gás



Fonte: Autor (2022).

Utilizaremos como parâmetro concentração do gás. Quando essa concentração aumenta para acima de 40%, um alarme é disparado. Para isso, teremos que utilizar o bloco “som” e “sem som”, escolhendo o pino no qual o *buzzer* está conectado, no caso o pino digital 5

Figura 73 – Alarme gás(condição)

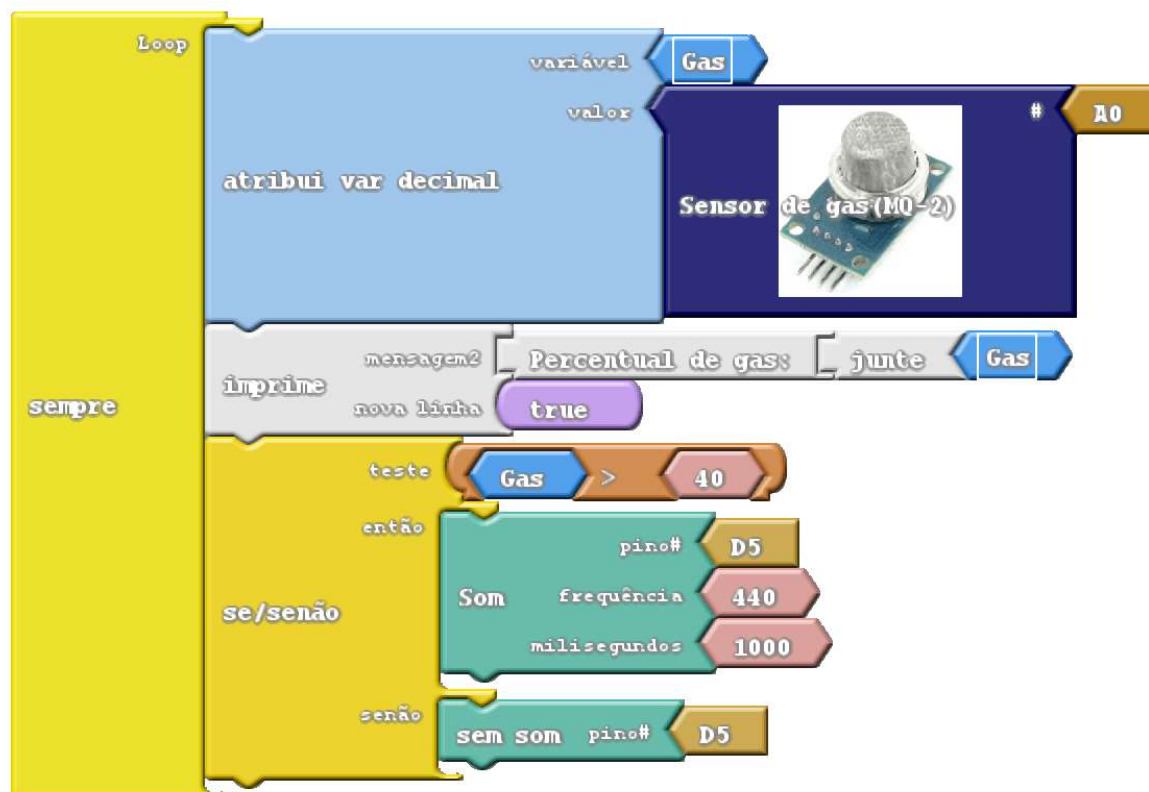


Fonte: Autor (2022).

Podemos escolher a frequência e os milissegundos no bloco “Som” e para encerrar qualquer som produzido pelo buzzer, utilizaremos o bloco sem som.

Unindo a estrutura da última figura, com o restante do código obtemos:

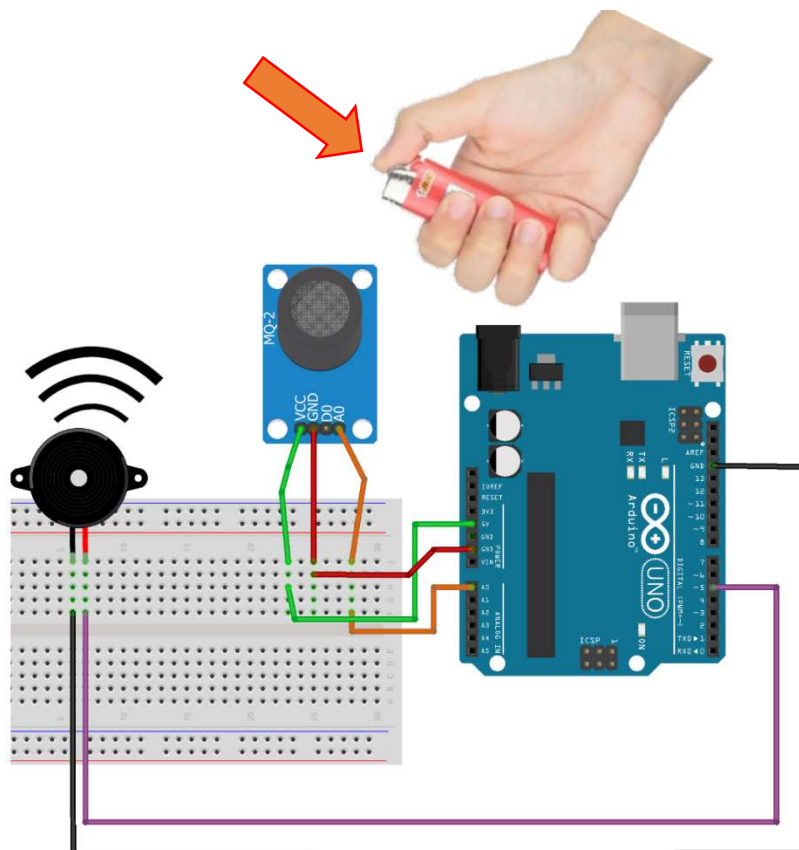
Figura 74 – Código completo



Fonte: Autor (2022)

Para finalizar devemos seguir as seguintes instruções:

Figura 75 – Automação Final



Fonte: Autor (2023).

Devemos apenas pressionar o êmbolo vermelho do isqueiro, para liberar o gás propano, porém sem acende-lo, perto do receptor do sensor. Quando o percentual de gás atingir 40%, o buzzer será disparado.

Figura 76 – Resultado das medidas

Timestamp	Percentage	Timestamp	Percentage
20:42:24.150	20.14	20:42:56.320	76.74
20:42:24.197	20.23	20:42:56.366	76.74
20:42:24.197	20.23	20:42:56.366	76.74
20:42:24.197	20.23	20:42:56.413	76.74
20:42:24.243	20.14	20:42:56.413	76.74
20:42:24.290	20.23	20:42:56.460	76.83
20:42:24.336	20.14	20:42:56.508	76.83
20:42:24.336	20.23	20:42:56.508	76.93
20:42:24.382	20.14	20:42:56.555	77.13
20:42:24.382	20.14	20:42:56.555	77.22
20:42:24.428	20.23	20:42:56.601	77.42
20:42:24.428	20.14	20:42:56.648	77.61
20:42:24.477	20.23	20:42:56.648	77.81
20:42:24.523	20.1	20:42:56.695	78.01

Fonte: Autor (2022).

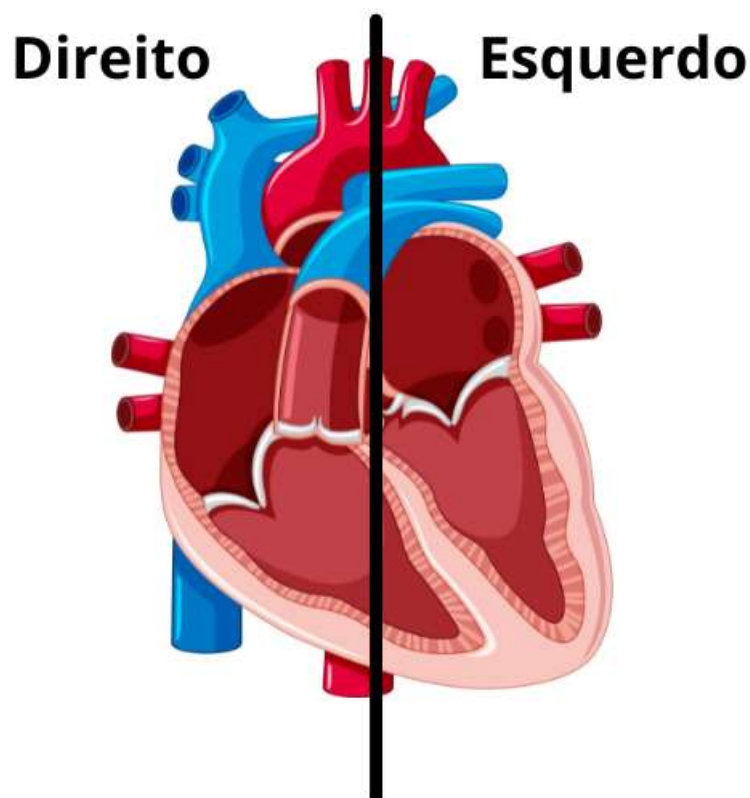
Podemos observar na figura acima que ao apertarmos o êmbolo do isqueiro, os valores de leitura do sensor se alteram aumentando o percentual de concentração de gás.

Pode-se discutir com os alunos os impactos ambientais da emissão de gases causadores do efeito estufa, sendo essa atividade, uma contextualização sobre os possíveis efeitos ao meio ambiente e saúde coletiva.

## Capítulo 9 – Saúde

O objetivo deste capítulo é apresentar uma aplicação da tecnologia na saúde, neste caso, utilizaremos um sensor de ECG (Eletrocardiograma) para ser programado pelos alunos. Precisaremos entender primeiro, como funciona o coração. O coração é dividido em duas partes, o lado direito e o lado esquerdo:

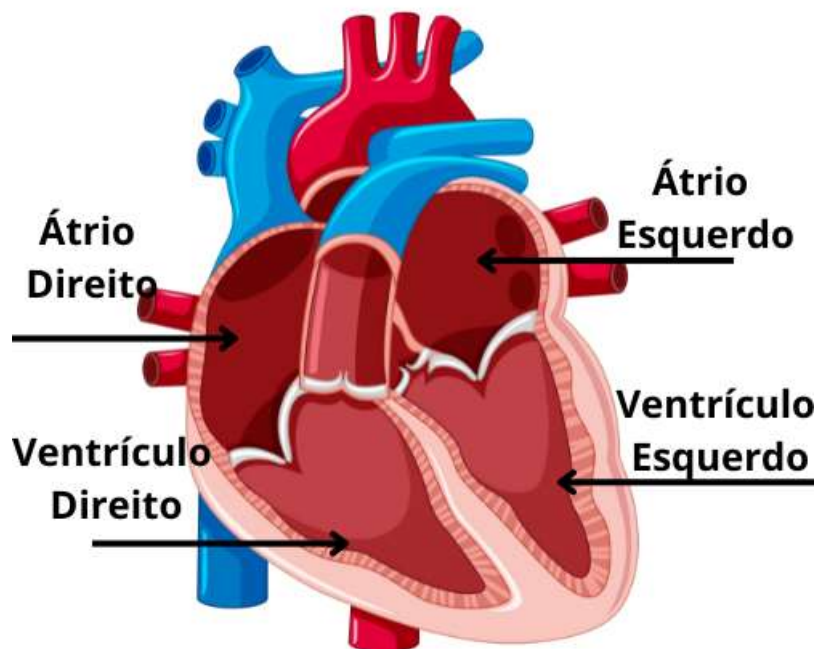
Figura 77 – Lados do coração



Fonte: Autor utilizando Canva pro (2023).

O coração tem um sistema de duas bombas, chamados de ventrículos. O sangue chega através das artérias e veias na cavidade do átrio, sendo liberado por uma válvula que chega até o ventrículo no qual realiza o bombeamento do sangue.

Figura 78 – Átrios e ventrículos

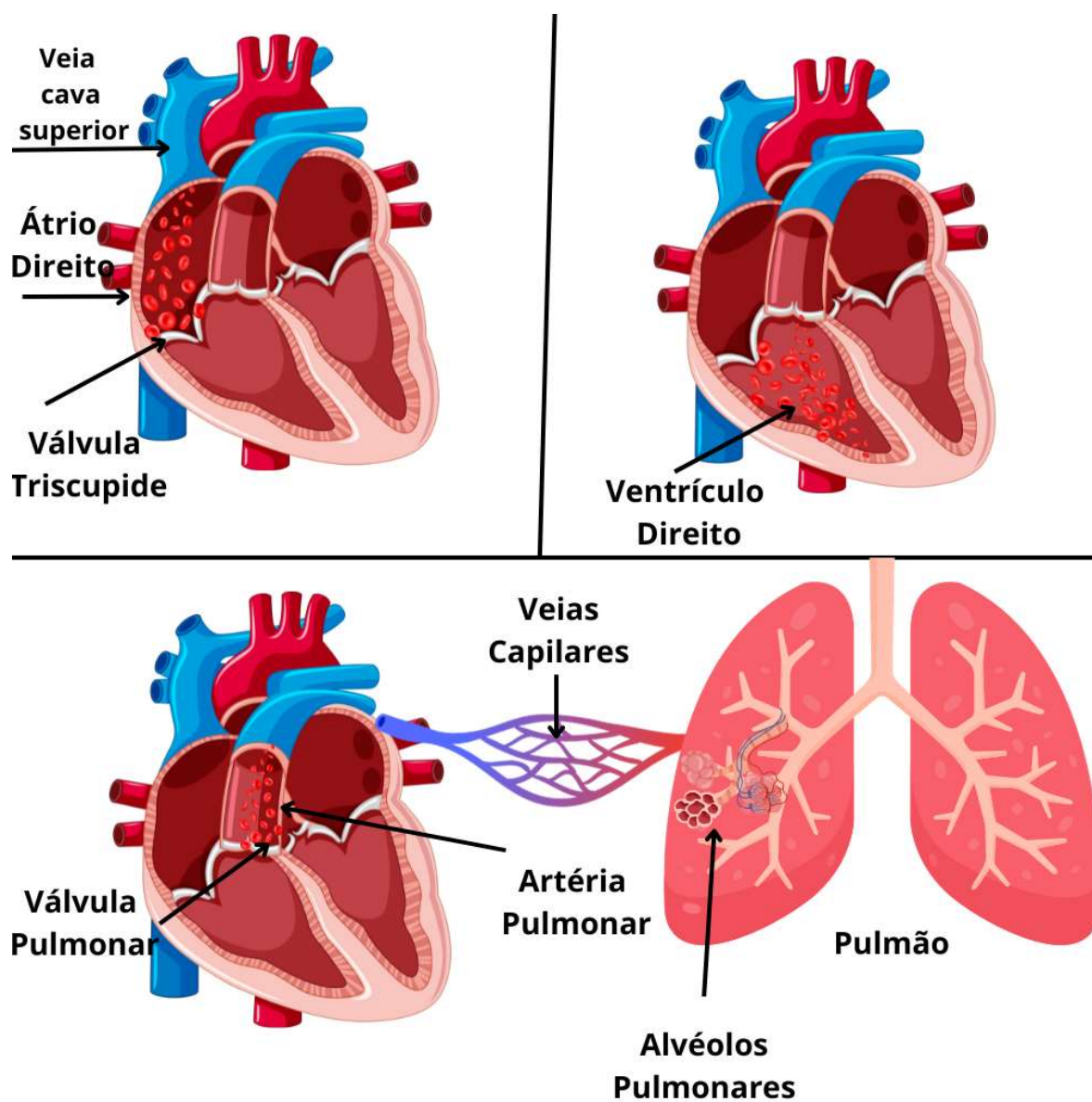


Fonte: Autor utilizando Canva pro (2023).

O sangue chega inicialmente pela veia cava superior ao átrio, passando pela válvula tricúspide até o ventrículo direito, que por sua vez, faz o bombeamento até artéria pulmonar passando primeiro pela válvula pulmonar. O sangue ao ser passado pela artéria pulmonar vai até o pulmão aonde vai ser saturado com oxigênio através de vasos capilares (vasos sanguíneos extremamente finos). O oxigênio é transportado até os vasos capilares através dos alvéolos pulmonares que funcionam como pequenos bolsões de ar, atravessando uma membrana extremamente fina. A explicação aqui dada é o que se chama de microcirculação e pode ser visualizada na Figura 79, sendo o sangue representado pelas hemácias:



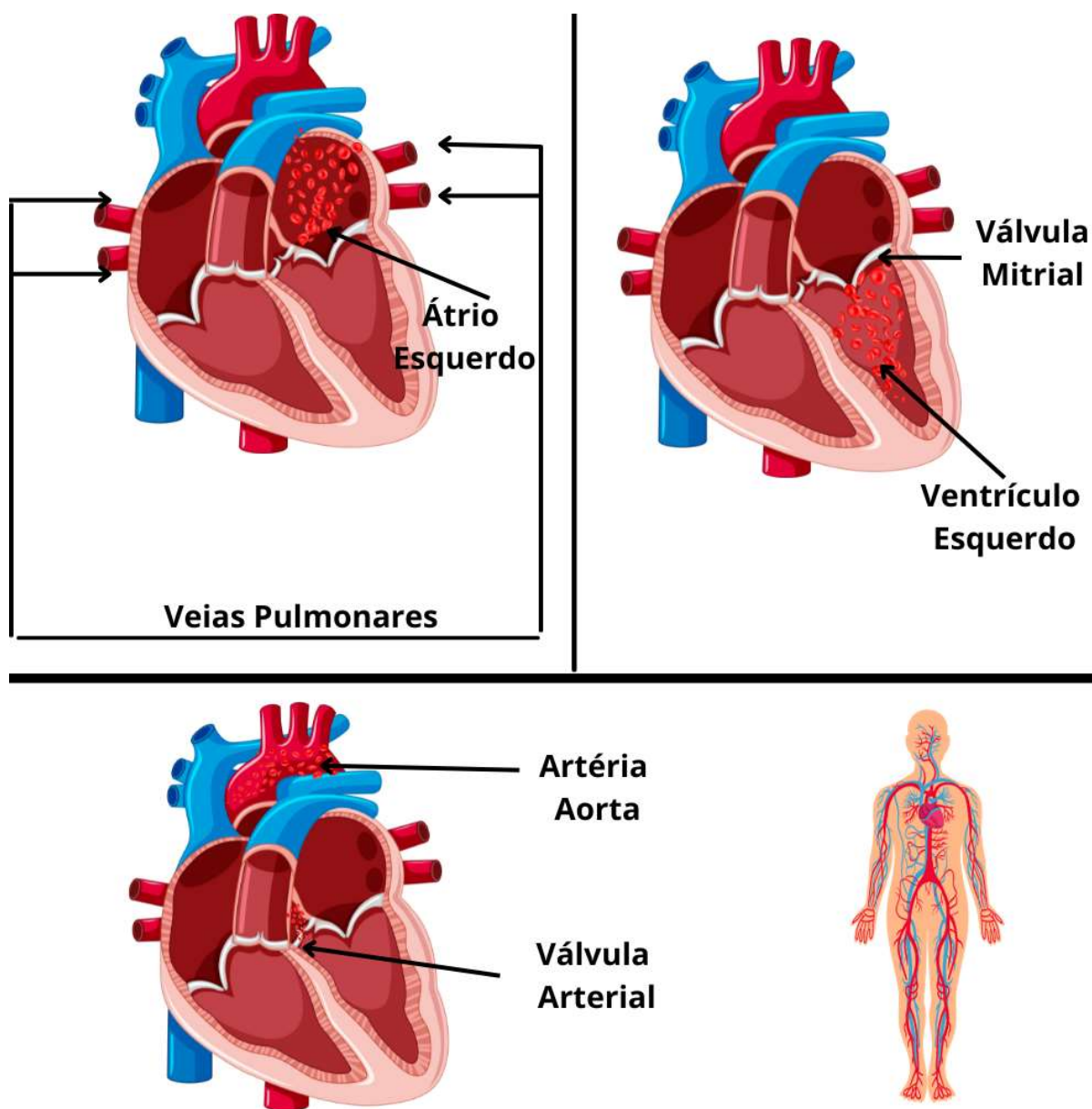
Figura 79 – Processo de bombeamento sanguíneo – Parte I



Fonte: Autor utilizando Canva pro (2023).

O sangue retorna dos vasos capilares do pulmão para o coração, pelas veias pulmonares, passando pela cavidade do átrio esquerdo. Posteriormente o sangue chega ao ventrículo esquerdo passando pela válvula mitral, o sangue então vai ser bombeado pelo ventrículo até a artéria aorta, porém tem que passar pela válvula arterial primeira. Ao ser bombeado pela artéria aorta será bombeado para todos os tecidos do corpo com exceção do pulmão. A aorta é o a artéria/vaso sanguíneo de maior calibre do corpo humano, ou seja, que passa a maior quantidade de sangue. A figura a seguir exemplifica os conceitos aqui abordados:

Figura 80 – Processo de bombeamento sanguíneo – Parte II

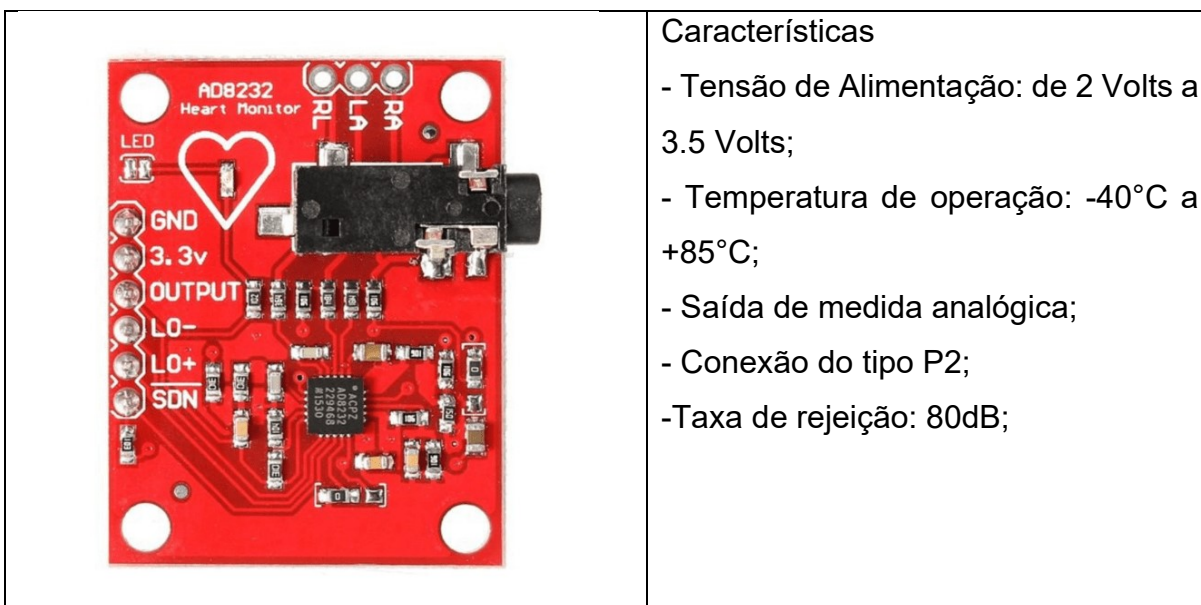


Fonte: Autor utilizando Canva pro (2023).

Qual a razão de estarmos abordando o funcionamento do coração aqui? O motivo é que, o conhecimento sobre tecnologia ele é mais amplo do que parece, sendo utilizado em diferentes seguimentos da ciência que neste caso, está associado a área de saúde.

Vamos finalmente abordar o funcionamento de nosso sensor. O sensor escolhido foi o AD8232 que se divide em duas partes, os módulos e o fio com os eletrodos.

Figura 81 – Módulo AD8232



Fonte: Retirado de <https://www.smartkits.com.br/sensor-de-batimentos-cardiacos-ecg-para-arduino-ad8232>

Figura 82 – Conector e elétrodos



Fonte: Retirado de <https://www.smartkits.com.br/sensor-de-batimentos-cardiacos-ecg-para-arduino-ad8232>

De acordo com o *datasheet* do fabricante do sensor, o mesmo tem como função de extrair, amplificar e filtrar pequenos sinais biopotenciais, ou seja, o mesmo consegue extrair informações referente a movimentação muscular. Apesar de ter

como finalidade o ECG, o mesmo pode ser utilizado como sensor de sinais de estímulos musculares, sendo popularmente utilizado também para construção de braços robóticos. O seu funcionamento é de um sensor comum com o uso de medidas analógicas, medindo, a variação do movimento muscular que neste caso, queremos analisar os movimentos cardíacos que ocorrem de maneira rítmica e bem ordenada.

Observe a figura a seguir para compreendermos o sinal que receberemos do sensor:

Figura 83 – Sinal do ECG



Fonte: Retirado de <https://www.angomed.com/wp-content/uploads/2015/05/Ondas-ECG.jpg>

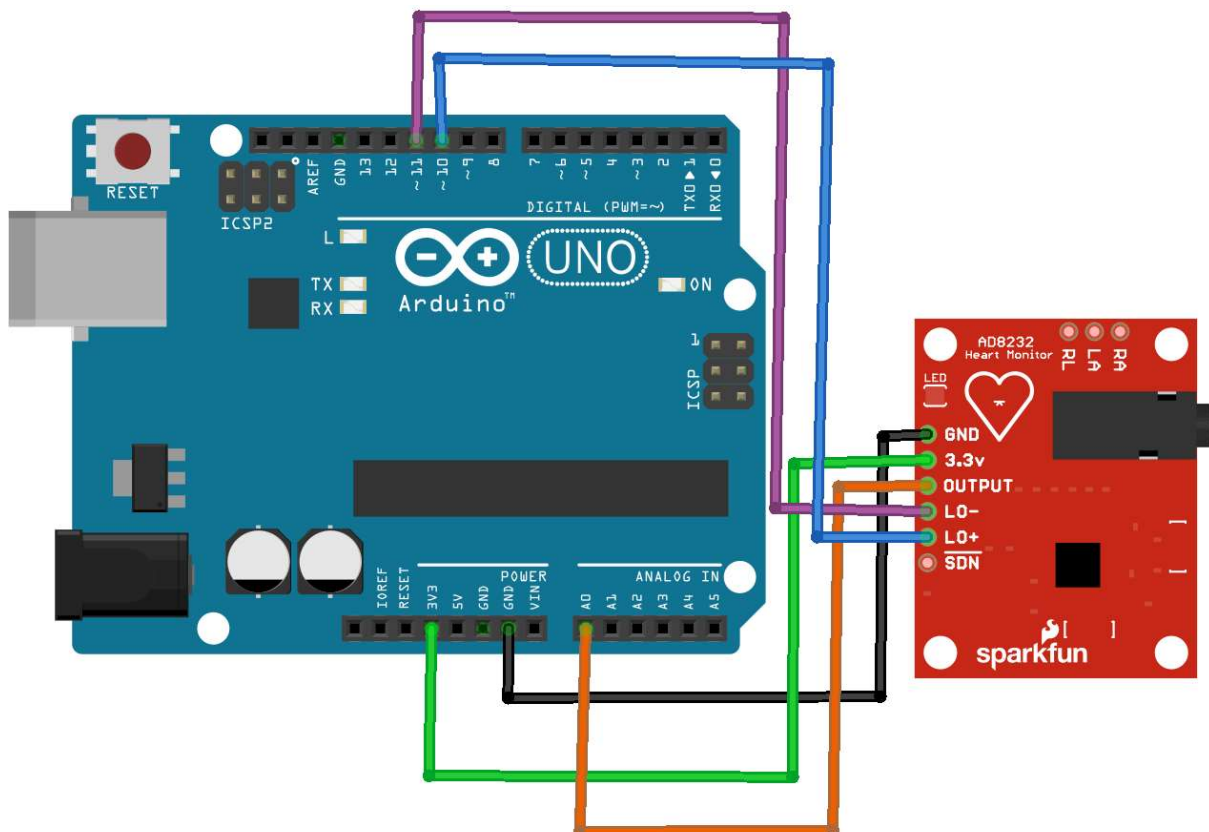
No ponto P da imagem acima, corresponde a contração atrial, que é quando o sangue passa das válvulas do coração para os ventrículos, ou seja, representa o enchimento de sangue do ventrículo.

Agora temos o sinal chamado de complexo QRS, onde Q corresponde a uma pequena deflexão pra baixo. O ponto R corresponde a uma subida ou deflexão para cima e então uma decida ou deflexão para baixo (ponto S). O ponto T e U correspondem uma onda crescente, demonstrando a repolarização ventricular, sendo o movimento de liberação de íons.

O funcionamento do coração se dá por processos periódicos de contração e relaxamento muscular.

Vamos as conexões de nosso sensor:

Figura 84 – Conexões módulo AD8232



fritzing

Fonte: Autor (2023).

Ligaremos o pino GND do sensor ao pino GND da placa (em preto), em sequência ligaremos a alimentação de 3,3 volts do sensor no pino de 3,3 volts da placa. O output do sensor ligaremos numa porta analógica que neste caso, escolhemos a porta A0. O conector LO- ligaremos na porta digital 11 e a porta LO+ na porta digital 10.

Tabela 3 – Pinos conectores do Sensor AD8232

Sensor	Pino
GND	GND
3,3 Volts	3,3 Volts
Output	A0
LO-	D11
LO+	D10

Fonte: Autor (2023)

As conexões do cabo de comunicação e dos eletrodos se dá da seguinte maneira:

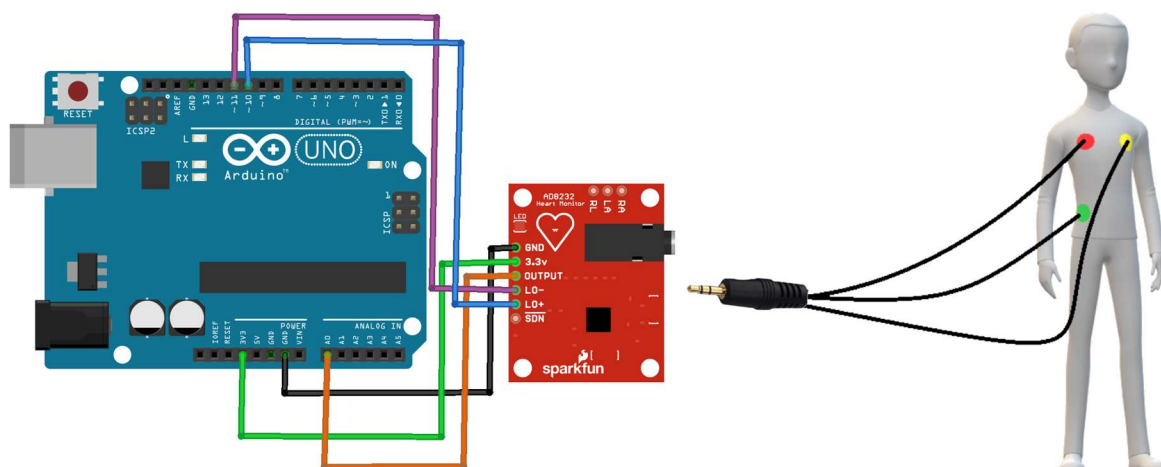
Figura 85 – Conectores dos eletrodos



Fonte: Autor (2023)

O eletrodo amarelo colocaremos no lado esquerdo do peito, o eletrodo vermelho no lado direito do peito e o eletrodo verde no abdômen do lado direito. O ideal é que o participante esteja imóvel de preferência sentado ou deitado.

Figura 86 – ECG conexão final



Fonte: Autor (2023).

Após terminada todas as conexões, iremos então para a programação e aspectos do software. Foi incluído o seguinte bloco no Ardublock com intuito de auxiliar os estudantes.

Figura 87 – Bloco módulo AD8232



Fonte: Autor (2022).

No bloco acima temos 3 pinos. O de leitura que corresponde a uma medida analógica ligado ao pino A0. O pino LO+ e LO- que são dois pinos digitais de leitura de sinal (caso receba um sinal terá um valor igual a 1 caso contrário, terá um valor igual a zero).

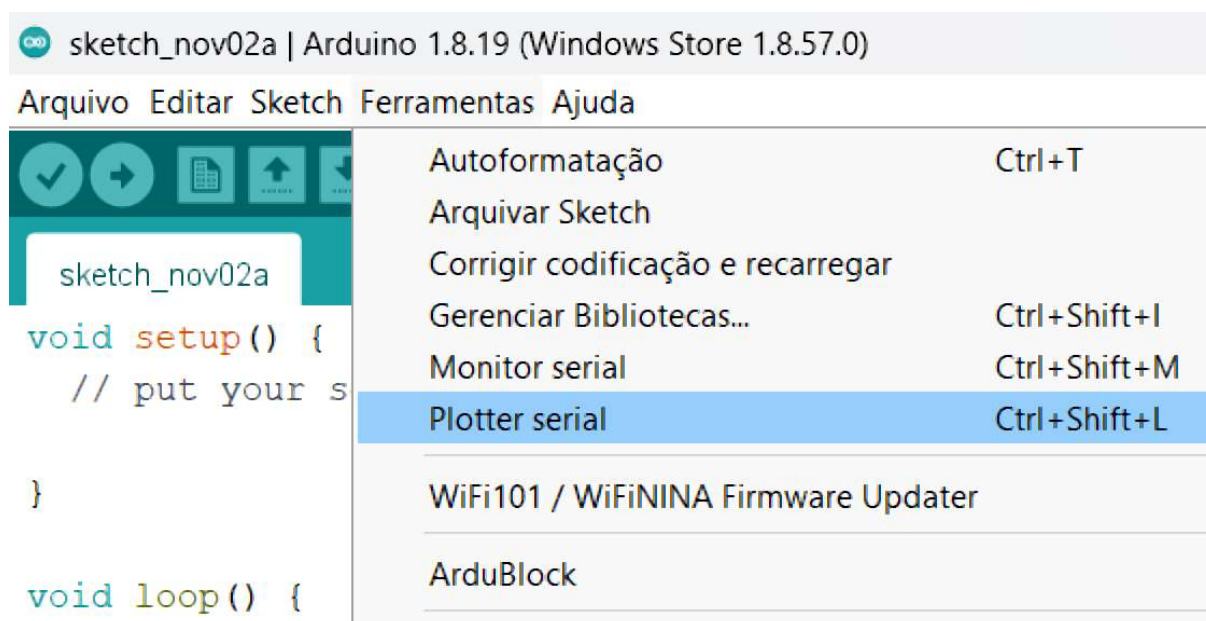
Figura 88 – Programação final ECF



Fonte: Autor (2022).

Para verificarmos os resultados temos que irmos até ferramentas e Plotter serial:

Figura 89 – Abrindo o Plotter Serial

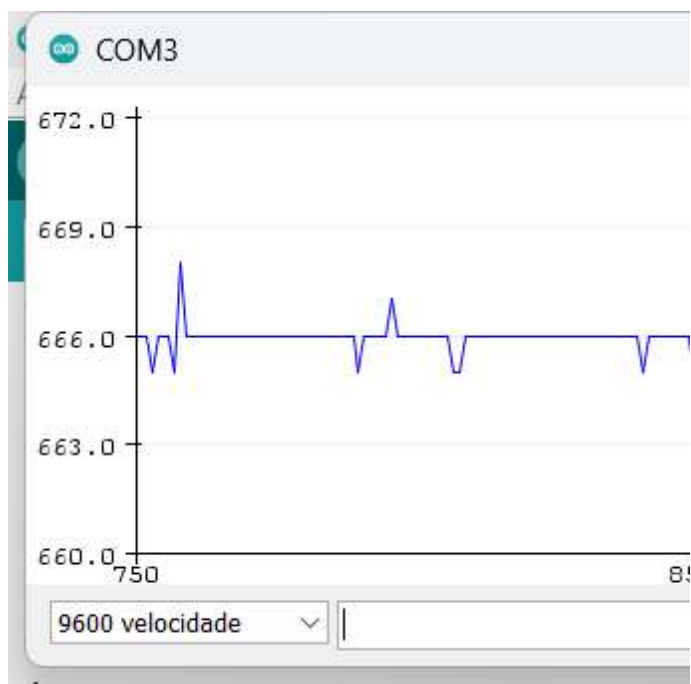


Fonte: Autor (2023).

O resultado que obtemos no nosso sensor AD8232, será semelhante a este:



Figura 90 – Resultado Plotter Serial



Fonte: Autor (2023).

Na figura acima podemos ver a variação da medida analógica devido a medida ao processo descrito aqui sobre o processo que ocorre em um batimento cardíaco.

**OBSERVAÇÃO:** O uso do sensor, assim como, a coleta de medidas é apenas para fins didáticos e **NÃO** devem ser usados como resultado de exame médico real. Caso necessite, ou apresente dores no peito, formigamento, taquicardia, tonturas, desmaios entre outros sintomas, consulte um médico especialista.

Esta atividade pode ser introduzida tanto com alunos do ensino médio, quanto com os alunos do ensino fundamental, sendo o objetivo central a utilização da tecnologia juntamente com a ciência, de modo, a demonstração de que as mesmas tem como o objetivo promover uma qualidade de vida e prevenção de doenças.

## Capítulo 10 – Conectividade

Chegamos finalmente ao nosso último capítulo, aqui abordaremos uso da conectividade, mais especificamente, automações controladas pela internet. Para isso, utilizaremos uma nova placa de automação chamada ESP 32. Continuaremos a utilizar a IDE do Arduino. Em mundo cada vez conectado, o uso da IoT (internet das coisas) se torna cada vez uma realidade, a utilização de produtos e serviços conectado é uma realidade crescente, de tal modo que, este tópico não poderia ser deixado de lado. O guia de instalação do ESP32 está disponível no Apêndice B.

Vamos nos aprofundar na utilização do ESP32 e seu funcionamento.

Figura 91 – ESP32



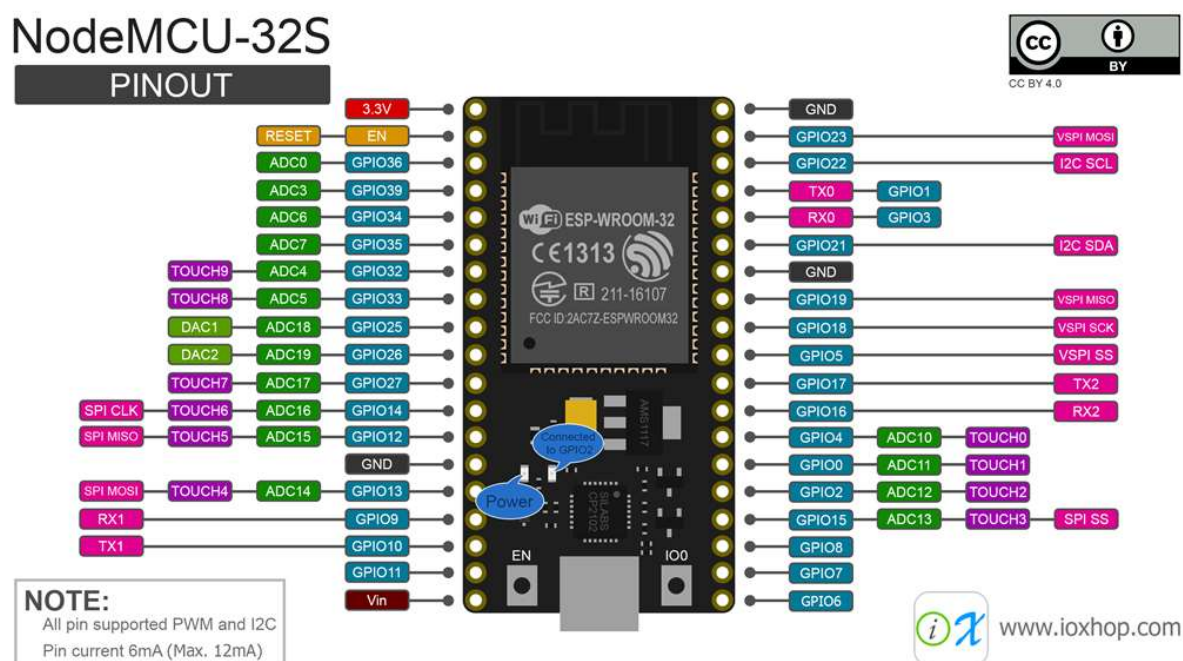
Fonte: Retirado de [https://www.makerhero.com/wp-content/uploads/2017/11/6WL55\\_1.jpg](https://www.makerhero.com/wp-content/uploads/2017/11/6WL55_1.jpg)

O ESP32 é um microcontrolador lançado no mercado no ano de 2016 pela empresa chinesa TSMC. Ele conta com um processador de 40 nm (nanômetros) tendo sua frequência de trabalho 240 Mhz (mega hertz) e 384 kb de memória RAM e 4 Mb de memória flash. Conta também com módulos de bluetooth versão 4.2 e wifi 802.11 b/g/n (que opera numa frequência de 2,4 Ghz) integrados à placa.

Mas o que essa placa traz de diferencial em relação a placa Arduino UNO para optarmos sua troca? Ambas são *opensource*, apresentam um desempenho equivalente para maioria das automações, a linguagem é equivalente. Bom, para podermos utilizar *wifi* e *bluetooth* na placa Arduino precisamos utilizar módulos específicos e, devido a facilidade e praticidade de termos esses componentes

integrados no ESP32, foi feita a escolha pela troca de microcontrolador. Vamos ver a seguir as principais características dos pinos do ESP32:

Figura 92 – Pinos do ESP32



Fonte: Retirado de <https://www.fernandok.com/>

Uma das primeiras diferenças entre o Arduino e o ESP32 é que os pinos do Arduino são ou digitais (entrada e saída) ou analógicos, no ESP32 as entradas são do tipo GPIO (*General-Purpose Input-Output*), sendo em tradução literal como: entradas-saídas de uso geral, ou seja, um canal pode ser usado para por exemplo ligar um LED (como um pino digital) ou coletar uma leitura analógica. Outra diferença está justamente na leitura digital enquanto o Arduino tem uma resolução de leitura analógica de 10 bits, e o ESP32 tem uma resolução de 12 bits, que na prática significa:

Tabela 4 – Comparação leitura analógica Arduino vs ESP32

Arduino (10 bits) $2^{10}$	ESP32 (12 bits) $2^{12}$
1024	4096

Fonte: Autor (2023).

Outra diferença está na diferença de potencial (ddp) que cada plataforma trabalha, enquanto os pinos do Arduino trabalham numa diferença de potencial base

de 5 Volts, os GPIOs do ESP32 trabalham com diferenças de potencial de 3,3 Volts. Esse fator deve ser levado em conta, pois existem sensores específicos e relês que trabalham com ddp de 5 volts e a escolha do ESP32 pode dificultar a sua utilização, entretanto, o problema é contornável com fontes externas específicas que transformam os 3,3 volts do ESP32 em uma diferença de potencial de 5 volts.

Tabela 5 – Comparação de diferença de potencial do Arduino vs ESP32

Arduino	ESP32
5 Volts	3,3 Volts

Fonte: Autor (2023).

O ESP32 possui um total de 34 GPIO, sendo o número de canais destes pinos:

- Canais de 18 pinos de saída (medidas analógicas)
- 10 canais do sensor de toque capacitivo
- 3 canais UART
- 2 canais SPI
- 2 canais I2S
- 2 canais DAC
- 2 canais I2C
- PWM – disponível em todos os pinos GPIO
- Interrupções – disponível em todos os pinos GPIO
- Entrada/saída digital disponível em todos os pinos GPIO.

O ESP32 ainda conta com um total de 3 pinos GND, 1 pino de *reset*, 1 pino de 3,3 volts e 1 pino de alimentação da placa (podendo ser feita pela entrada USB).

Feita esta introdução inicial vamos para as modificações no Ardublock dedicadas a programação do ESP32, que não existiam antes deste trabalho e foram adicionadas para facilitar a utilização dos usuários que desejam trabalhar com a placa do ESP32, sendo introduzido duas novas abas com uma família de códigos no Ardublock. As abas adicionadas no Ardublock são ESP32(base) e Servidor Web, como pode ser visto na figura a seguir:

Figura 93 – Abas novas



Fonte: Autor (2023).

Vamos abordar os códigos primeiro da aba ESP32(base), sendo o primeiro bloco o para escolher um IP estático do ESP.

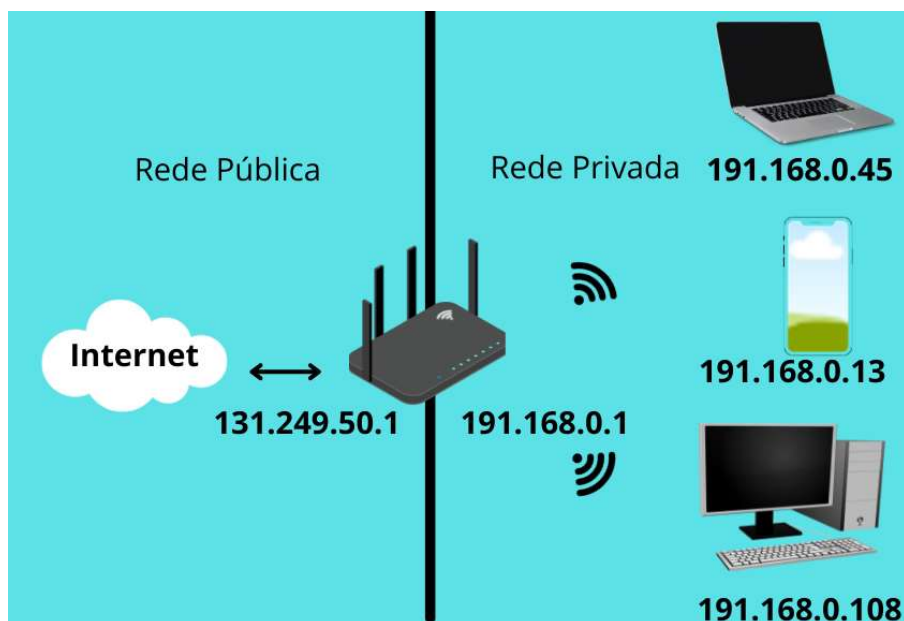
Figura 94 – Bloco IP Estático



Fonte: Autor (2023).

Para compreendermos a funcionalidade do bloco acima, precisamos primeiro ter uma introdução à conexões com a internet e funcionamento de IP.

Figura 95 – IP interno e externo

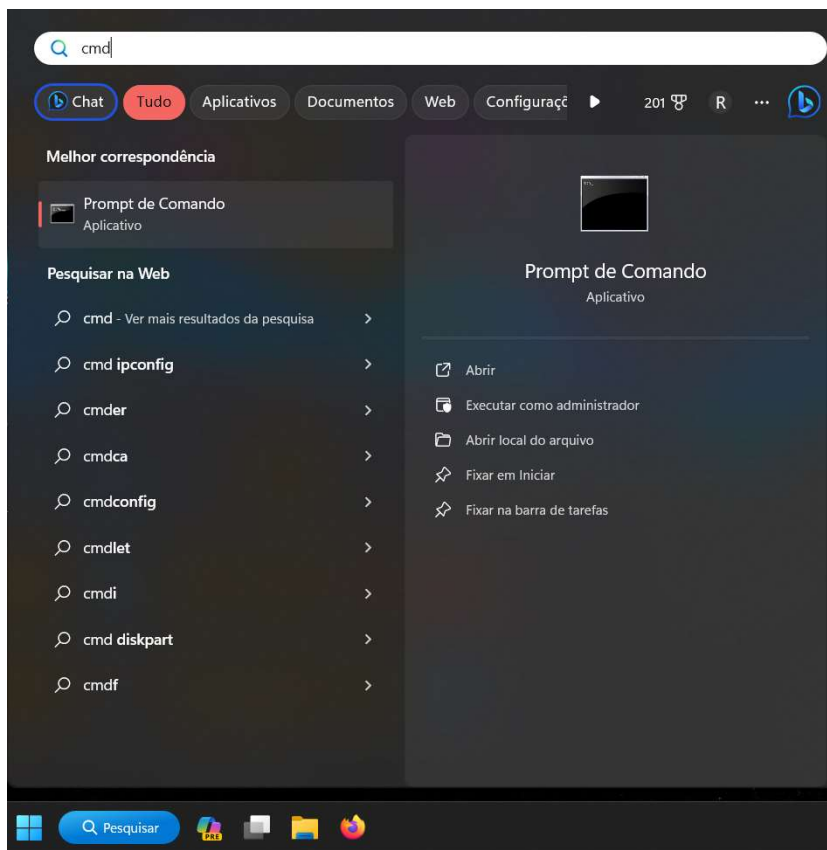


Fonte: Produzido pelo autor utilizando Canva Pro (2022).

Temos dois tipos de IP, o interno e o externo. O externo é a identificação na rede mundial de computadores (internet), ou seja, é a identificação de todos os dispositivos conectados no roteador de internet. Os IPs internos são atribuídos pelo roteador, ou seja, cada dispositivo vai ter um valor distinto, sendo que o início do IP é igual ao do *Gateway*, exemplo 191.168.0 – XXX e os últimos dígitos podem ser qualquer valor entre 2 e 255 como pode ser visto na figura acima. O *gateway* é simplesmente a identificação do endereço do roteador e a máscara de subrede é utilizada pelo roteador para encaminhar pacotes de dados pelo caminho correto. É preciso tomar cuidado ao utilizar o bloco acima caso tenha um dispositivo já com aquele endereço de IP (IP interno). A configuração de IP interno estático pode ser feita tanto mexendo nas configurações do roteador quanto pelo bloco que utilizamos. Caso você não tenha as informações do Gateway e da máscara de subrede, você pode se conectar utilizando seu computador na mesma rede que irá conectar o ESP32 e fazer os seguintes passos:

Passo 1 – Apertar a tecla Windows e digitar “cmd”

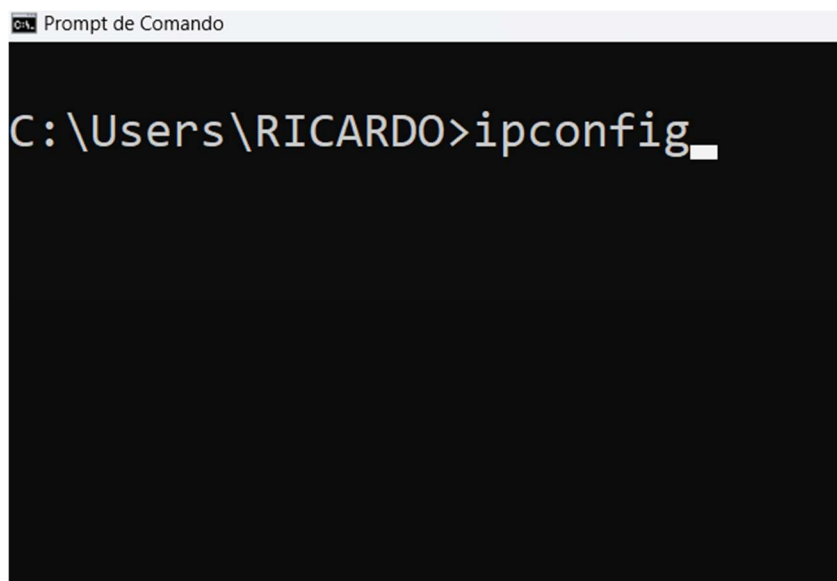
Figura 96 – Passo 1 para obter as informações de Gateway e subrede



Fonte: Autor (2023).

Passo 2 – Digitar o seguinte comando no cmd “ipconfig”

Figura 97 – Passo 2 para obter as informações de Gateway e subrede



Fonte: Autor (2023).

Passo 3 – Identificar a placa de rede conectada e pegar as informações.

Figura 98 – Passo III informações da rede

```
Prompt de Comando
Adaptador de Rede sem Fio Conexão Local* 3: ←
Estado da mídia. . . . . : mídia desconectada
Sufixo DNS específico de conexão. . . . . :

Adaptador de Rede sem Fio Wi-Fi:
Sufixo DNS específico de conexão. . . . . : ██████████
Endereço IPv6 . . . . . : ████████████████████
Endereço IPv6 Temporário. . . . . : ████████████████████
Endereço IPv6 de link local . . . . . : ████████████████████
Endereço IPv4. . . . . : ████████████████████
Máscara de Sub-rede . . . . . : 255.255.255.0
Gateway Padrão. . . . . : ████████████████████
                            192.168.0.1
```

Fonte: Autor (2023).

**Observação:** As informações mais sensíveis foram ocultadas.

Um último ponto que precisa ser mencionado é que os números no bloco de IP estático tem que ser separados por vírgula. Exemplo: 192,168,0,44

**Observação:** É altamente recomendável utilizar a fixação de IP para o dispositivo através das configurações do roteador ao invés da utilização deste bloco, a razão é para evitar o que chamamos de conflito de IP que é quando dois aparelhos tem o mesmo número de IP interno. O bloco é apenas auxiliar na configuração, mas não é sempre recomendável a sua utilização. É interessante sua utilização para casos de conexão internas ou configurações de portas.

O bloco a seguir é o que vai permitir a conexão do ESP32 com a rede de internet e se conectar com o roteador de internet.



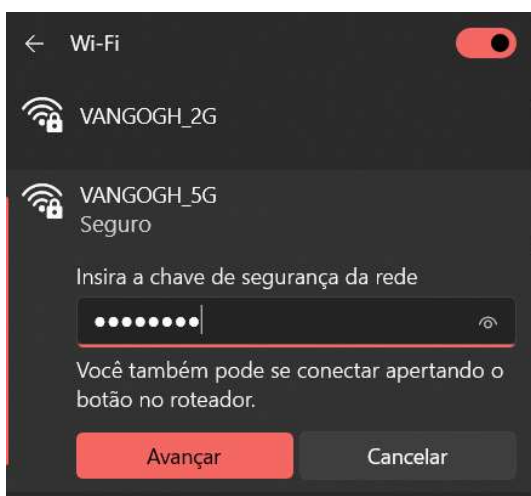
Figura 99 – Módulo de Wifi



Fonte: Autor (2023).

Pra utilizarmos o módulo wifi basta digitar o nome da rede e a senha da rede wifi, caso tenha dúvidas é possível olhar pelo computador e verificar as redes disponíveis. É necessário que o nome da rede e a senha estejam corretos para que se possa conectar na internet.

Figura 100 – Verificar o nome da rede

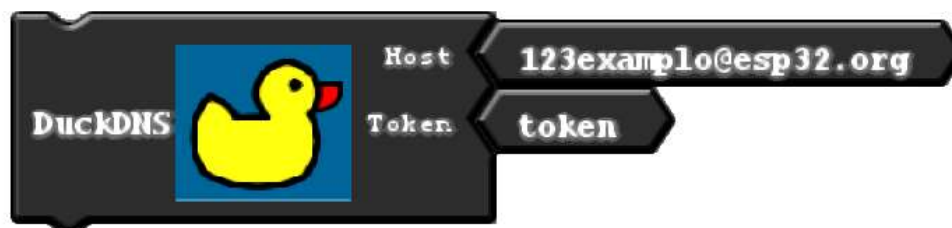


Fonte: Autor (2023).

O ESP32 aceita apenas conexões de redes de 2,4 Ghz (2G), então é necessário tomar cuidado com este detalhe na hora de realizar a conexão com a internet.

O bloco a seguir é do duckdns, ele funciona como um host que aponta para a o ESP32

Figura 101 – Bloco do DuckDNS

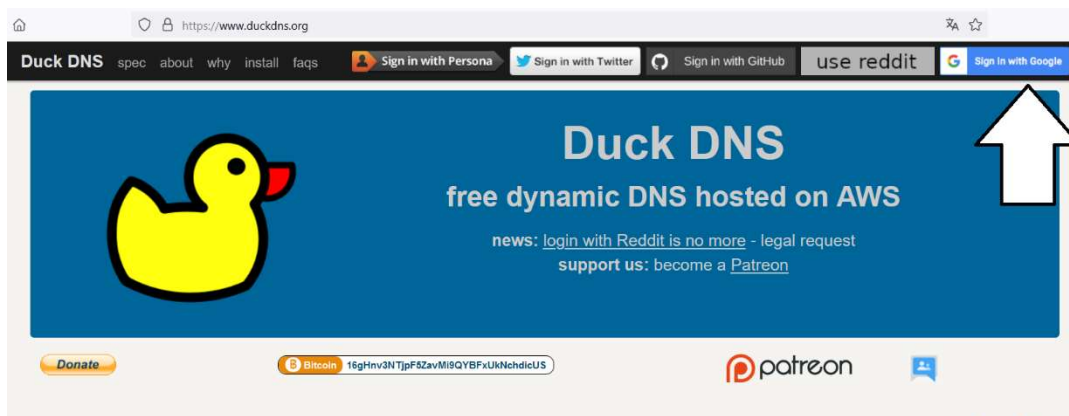


Fonte: Autor (2023).

O duckdns foi escolhido pois se trata de um serviço gratuito e *opensource*. Seu funcionamento vai ser explicado a seguir. O primeiro passo é criar uma conta no endereço <https://www.duckdns.org/> :

Passo 1 – Criar uma conta no Duckdns

Figura 102 – Criação de uma conta no DuckDNS

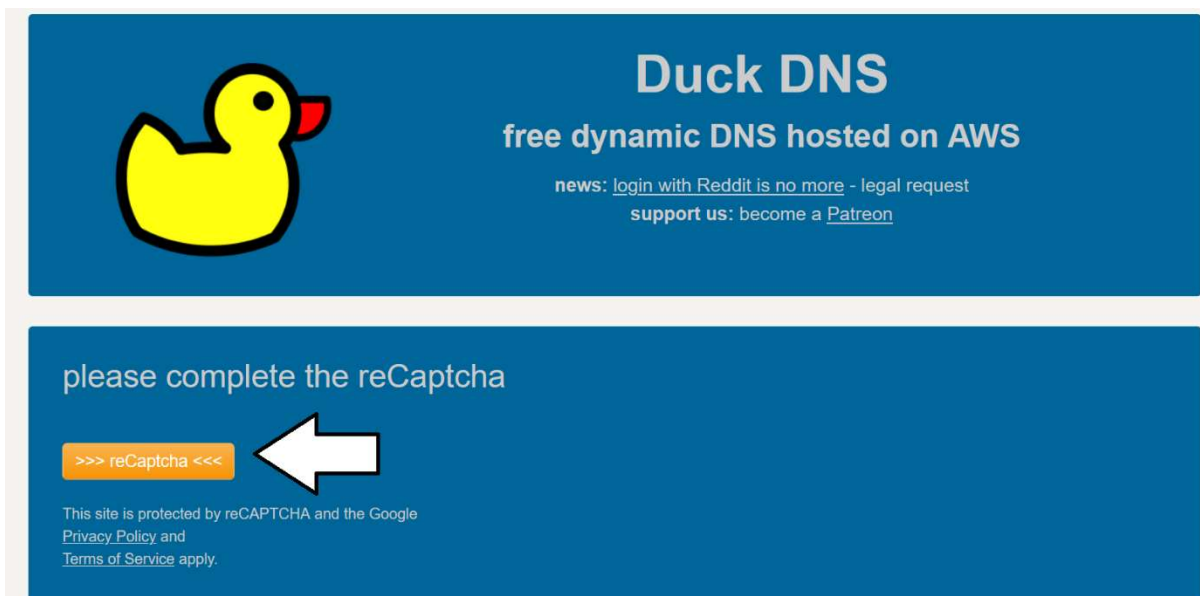


Fonte: Autor (2023).

Recomenda-se pela praticidade a utilização de uma conta google (gmail)

Passo 2 – Apertar o reCaptcha após logar (será necessária uma única vez)

Figura 103 – Apertar no reCaptcha



Fonte: Autor (2023).

A primeira informação necessária para o nosso bloco é o token gerado:

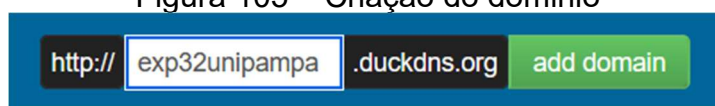
Figura 104 – Obtendo o token



Fonte: Autor (2023)

Após a criação da conta, temos que escolher um domínio, ou seja, um endereço de um site que será utilizado para acessar a nossa automação:

Figura 105 – Criação do domínio



Fonte: Autor (2023).

Após a criação do domínio (host) temos, temos a lista de domínios. Colocamos então em nosso bloco o endereço que nesse nosso exemplo é `esp32unipampa.duckdns.org`

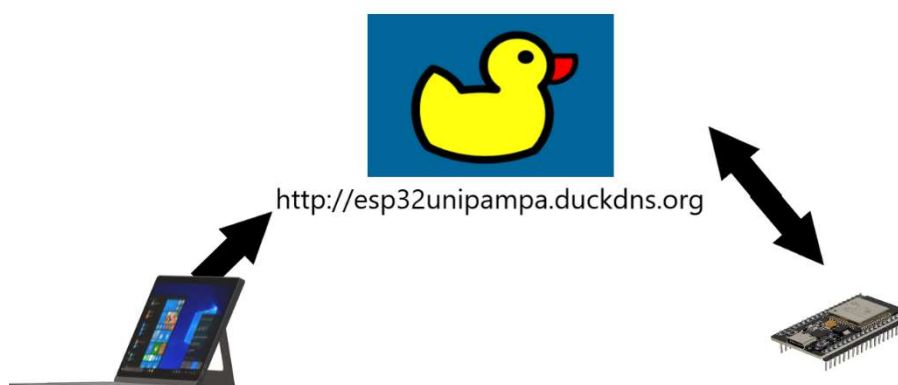
Figura 106 – Domínios

domain	current ip	ipv6	changed
esp32unipampa	179. <input type="text"/> <input type="button" value="update ip"/>	ipv6 address <input type="text"/> <input type="button" value="update ipv6"/>	0 seconds ago <input type="button" value="delete domain"/>

Fonte: Autor (2023)

O duckdns permite a criação de até 5 domínios diferentes por conta. Mas qual a razão de utilizarmos o duckdns? Existem dois motivos, o primeiro é para termos uma estrutura mais elegante e nos conectarmos ao ESP32 por um endereço escrito ao invés de um número de IP. Porém existe uma segunda razão para utilizarmos o duckdns, que é, o IP externo já explorado aqui pode ser estático ou dinâmico, ou seja, ou ele é um endereço de IP fixo que não se altera com o tempo, ou ele utiliza o que chamamos de protocolo DHCP no qual vai atribuir um IP na internet para uma determinada rede, o problema é que o protocolo DHCP acaba alterando o IP externo com o tempo, por isso chamamos de IP dinâmico. A função do duckdns aqui é sempre apontar para nossa aplicação no ESP32, assim se o endereço for alterado ainda sim ele estará conectado ao ESP32. O duckdns funciona então da seguinte maneira:

Figura 107 – Funcionamento do duckdns



Fonte: Autor (2023).

Um computador ou um usuário tenta acessar o ESP32 para controlar suas aplicações, para isso o usuário digitara o endereço do site vinculado pelo duckdns

com o ESP32, caso o IP externo da rede na qual o ESP32 está conectado se altere então o valor no host do duckdns vai se alterar, porém a aplicação ainda permanece online, mesmo que o IP externo da rede se modifique. O código do duckdns está programado para rodar no ESP32 a cada 10 segundos, ou seja, ele verifica se o IP externo se alterou ou não. Isso permite que a aplicação fique sempre online e caso mude o IP levará apenas 10 segundos para verificar a alteração.

**Observação:** O bloco do duckdns NÃO funciona ao mesmo que o do IP estático, recomenda-se configurar o IP estático pelo roteador, pois como já mencionado, existe o risco de existir o conflito de IP interno (quando dois dispositivos adquirem o mesmo IP).

O comando de imprimir abaixo funciona exatamente como o bloco de comunicação para o Arduino, com uma única diferença a velocidade está programada para funcionar na velocidade do ESP32 que no caso é 115200.

Figura 108 – Comando imprimir



Fonte: Autor (2023).

O bloco de medida analógico abaixo está programado para funcionar de acordo os GPIOs do ESP32;

Figura 109 – Medida analógica ESP32



Fonte: Autor (2023).

É necessário tomar cuidado apenas com um detalhe, existem 18 pinos de GPIO's que podem coletar informações analógicas, sendo estes descritos abaixo juntamente com o seu canal de comunicação equivalente:

Tabela 6 – GPIO de medidas analógicas

<b>Pino GPIO</b>	<b>Canal</b>
GPIO 0	ADC2_CH1
GPIO 2	ADC2_CH2
GPIO 4	ADC2_CH0
GPIO 12	ADC2_CH5
GPIO 13	ADC2_CH4
GPIO 14	ADC2_CH6
GPIO 15	ADC2_CH3
GPIO 25	ADC2_CH8
GPIO 26	ADC2_CH9
GPIO 27	ADC2_CH7
GPIO 32	ADC1_CH4
GPIO 33	ADC1_CH5
GPIO 34	ADC1_CH6
GPIO 35	ADC1_CH7
GPIO 36	ADC1_CH0
GPIO 37	ADC1_CH1
GPIO 38	ADC1_CH2
GPIO 39	ADC1_CH3

Fonte: Autor (2023).

O pino abaixo é apenas para escolhermos o pino GPIO, funcionando exatamente como os pino digitais e analógicos para o Arduino,

Figura 110 – Escolher GPIO



Fonte: Autor (2023).

O bloco abaixo é para definir um GPIO como um estado ALTO ou BAIXO. Lembrando que todos os pinos GPIO funcionam como saídas digitais então não é necessário especificar qual dos pinos pois esta função é válida para todos os pinos

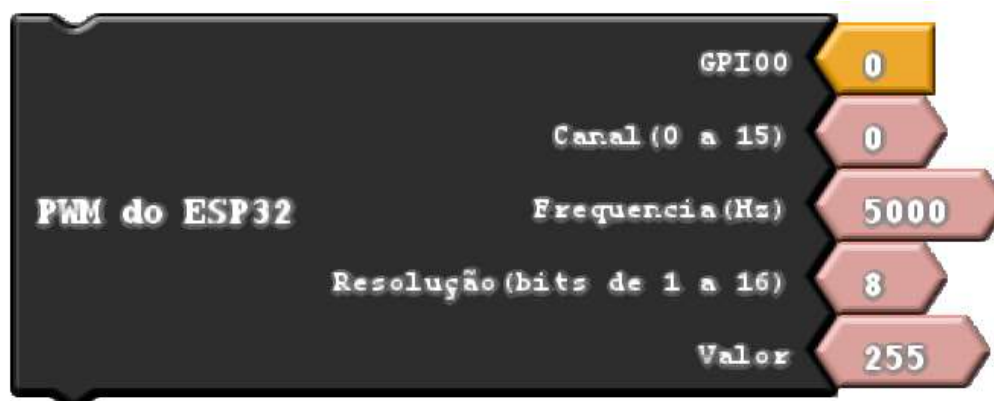
Figura 111 – Seta Pino digital



Fonte: Autor (2023)

Assim como todos os pinos GPIO podem ser saídas digitais, todos podem ser também pinos PWM, entretanto, precisamos tomar cuidado com algumas coisas. Apesar de todos os pinos poderem funcionar como PWM, o ESP32 tem apenas 16 canais disponíveis para essa função, ou seja, é possível ligar apenas 16 pinos PWM. Para isso é necessário escolher o canal que se deseja trabalhar que vai do número 0 até o número 15.

Figura 112 – Bloco PWM



Fonte: Autor (2023).

É preciso salientar também que, ao contrário do Arduino que a resolução é de 8 bits (256 estados possíveis), o ESP32 permite resoluções de 1 a 16 bits vamos ver a operação matemática a seguir para determinarmos a seguir permite nos compreendermos como funciona a resolução.

$$\text{Resolução} = 2^b$$

Observe na equação acima que a resolução do PWM será igual a 2 elevado na potência " b " que no nosso caso, é o número de *bits*.

Exemplos:

Resolução de 1 bit

$$\text{Resolução} = 2^1$$

$$\text{Resolução} = 2$$

No caso acima temos uma resolução igual a 2.

Resolução de 8 bits

$$\text{Resolução} = 2^8$$

$$\text{Resolução} = 256$$

Para determinarmos a frequência temos que levar em consideração que a frequência de *clock* do ESP32 é de 80 MHz (80.000.000 Hz), para descobrirmos a frequência que temos que utilizar, temos que calcular utilizando a seguinte operação matemática

$$\text{Frequência} = \frac{\text{Clock}}{\text{Resolução}}$$

A frequência que desejamos determinar aqui, é a frequência máxima permitida. Vamos usar alguns exemplos para tentar determinar a frequência máxima permitida de acordo com a resolução:

Exemplo para uma resolução de 8 bits

$$\text{Frequência} = \frac{80.000.000 \text{ Hz}}{256}$$

$$\text{Frequência} = 312.500 \text{ Hz}$$



Exemplo para uma resolução de 16 bits

$$Frequência = \frac{80.000.000 \text{ Hz}}{65.536}$$

$$Frequência = 1.220,70 \text{ Hz}$$

Para determinarmos os valores de nossos blocos temos que pegar os valores da resolução e diminuir 1, pois o estado zero também conta como um valor

$$Valor = Resolução - 1$$

Vamos observar os seguintes exemplos:

Exemplo para 8 bits

$$Valor = 256 - 1$$

$$Valor = 255$$

Os valores variam de 0 ao máximo de 255.

Exemplo para 10 bits:

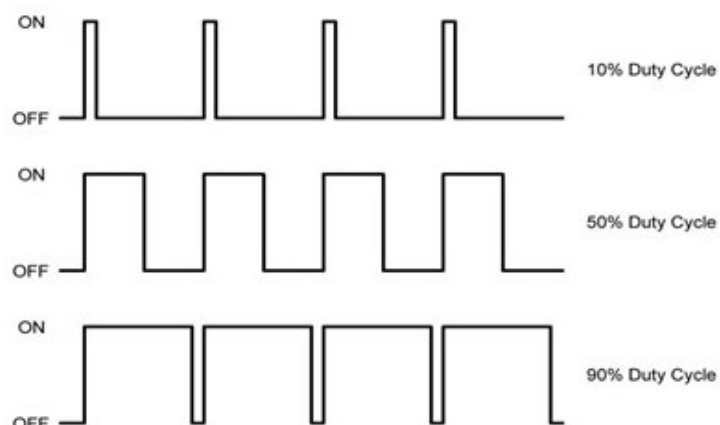
$$Valor = 1024 - 1$$

$$Valor = 1023$$

Os valores variam de 0 ao máximo 1023

Vale a pena destacarmos o funcionamento do PWM. O PWM (*Pulse Width Modulation*) que em tradução literal significa modulação por largura de pulso e tem como função o controle da tensão média de um pino, ou seja, controlar a sua largura e sua frequência de oscilação.

Figura 113 – Variações em pinos PWM



Fonte: Retirado de <https://www.mundodaeletrica.com.br/y/11273/forma-de-onda.jpg> .

Com a finalidade de facilitar para o usuário, a tabela a seguir determina os números de bits, resolução, variação dos valores e a frequência máxima permitida.

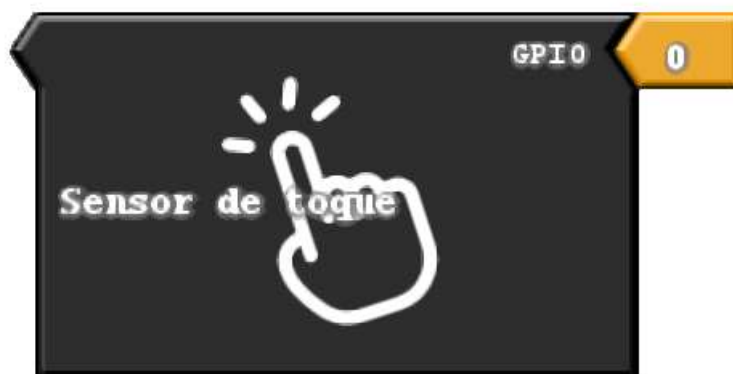
Tabela 7 – Estados PWM

Bits	Resolução	Variação dos valores	Frequência Max (Hz)
1	2	de 0 a 1	40.000.000
2	4	de 0 a 3	20.000.000
3	8	de 0 a 7	10.000.000
4	16	de 0 a 15	5.000.000
5	32	de 0 a 31	2.500.000
6	64	de 0 a 63	1.250.000
7	128	de 0 a 127	625.000
8	256	de 0 a 255	312.500
9	512	de 0 a 511	156.250
10	1024	de 0 a 1023	78.125
11	2048	de 0 a 2047	39.062,5
12	4096	de 0 a 4095	19.531,25
13	8192	de 0 a 8191	9.765,63
14	16384	de 0 a 16383	4.882,81
15	32768	de 0 a 32767	2.441,41
16	65536	de 0 a 65535	1.220,7

Fonte: Autor (2023)

O ESP32 conta com 10 sensores de toque capacitivos associados a 10 GPIOs. O bloco a seguir tem como finalidade a coleta de informações e funciona na prática como uma medida analógica, mas utilizando as funções relacionadas ao sensor de toque.

Figura 114 – Sensor capacitivo



Fonte Autor (2023).

Os GPIOs associados a cada sensor de toque pode ser visto na tabela a seguir:

Tabela 8 – GPIOs associados ao sensor de toque

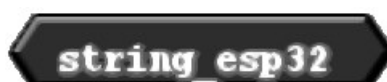
GPIO	Sensor de Toque
GPIO 4	T0
GPIO 0	T1
GPIO 2	T2
GPIO 15	T3
GPIO 13	T4
GPIO 12	T5
GPIO 14	T6
GPIO 27	T7
GPIO 33	T8
GPIO 32	T9

Fonte: Autor (2023)

É necessário escolher o GPIO correto de acordo com o sensor de toque escolhido.

Para encerrarmos este tópico vamos ao nosso bloco de variável, que ao contrário dos abordados até aqui, não está em nenhuma das duas abas novas, mas na aba variáveis:

Figura 115 – Variável ESP32



Fonte: Autor (2023)

A função desta variável é ela ser de uso geral sendo a solução encontrada para adaptar o ESP32 para o Ardublock. Este bloco pega tal qual o texto escrito dentro do bloco e transcreve este texto para dentro do código.

Abordaremos agora os blocos associados ao Servidor Web. A função destes blocos é a criação da interface e dos elementos associados, ou seja, criaremos a estrutura de um site com medidas analógicas, pinos digitais, pinos PWM, geradores de gráficos entre outros. Quando tratamos de estruturas de sites as possibilidades são muito próximas do infinito, tanto no critério de elementos quanto no processo estético, porém o objetivo aqui, é construir uma base sólida e funcional para o controle do ESP32 a distância.

O primeiro bloco que utilizaremos é o Servidor Web que gera a interface base do site, ou seja, criamos uma estrutura geral e colocaremos os blocos associados conectados dentro do servidor web. Na figura a seguir conseguimos visualizar a estrutura do nosso bloco:

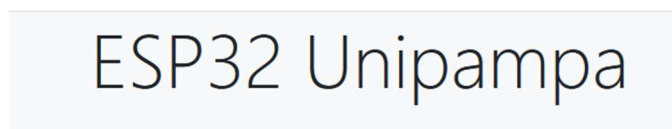
Figura 116 – Bloco de servidor web



Fonte: Autor (2023).

Observe que temos na estrutura do bloco o título do site, sendo este, um elemento gráfico onde aparece no topo do site com determinado título.

Figura 117 – Título do site



Fonte: Autor (2023)

No conector conteúdo é onde todos os elementos do site devem constar, por exemplo os botões que controlam GPIOs, medidas analógicas, gráficos e etc, sendo necessário arrastar para dentro do bloco servidor web essas funções.

O conector de porta é onde deve constar a porta em que o site vai ser encaminhado. É necessário fazer esta configuração nas configurações do roteador para liberação da porta. As configurações podem variar de fabricante para fabricante, entretanto vou demonstrar aqui como é feita a configuração no roteador que tenho disponibilidade.

**Observação:** É possível utilizar o sistema de roteamento do celular, mas NÃO é possível fazer o encaminhamento de portas do dispositivo, o que infelizmente, torna a conexão com ESP32 apenas uma conexão local e não efetivamente uma conexão na internet, mas dependendo da proposta em sala de aula, ainda é possível a sua realização. Foi necessário estudos sobre como funciona o pacote de dados de redes móveis e a descoberta é de que não é possível realizar este tipo de encaminhamento devido a uma limitação do próprio sistema Android e do funcionamento das redes móveis.

Feita a observação acima, vamos efetivamente para as nossas configurações, o primeiro passo a se realizar é dentro do próprio Ardublock, o primeiro passo é escolher um IP estático interno (rede privada), isso vai facilitar durante a nossa configuração de porta. Para tornar estático o IP precisamos primeiro as informações de máscara de sub-rede e Gateway:

Figura 118 – Máscara de sub-rede e gateway

```

Adaptador de Rede sem Fio Wi-Fi:
Sufixo DNS específico de conexão. . . . . : 
Endereço IPv6 . . . . . : 
Endereço IPv6 Temporário. . . . . : 
Endereço IPv6 de link local . . . . . : 
Endereço IPv4. . . . . : 
Máscara de Sub-rede . . . . . : 255.255.255.0
Gateway Padrão. . . . . : 192.168.0.1

```

Fonte: Autor (2023)

Vamos escolher como IP estático interno o IP 192.168.0.194. O código para esta configuração pode ser visto abaixo:

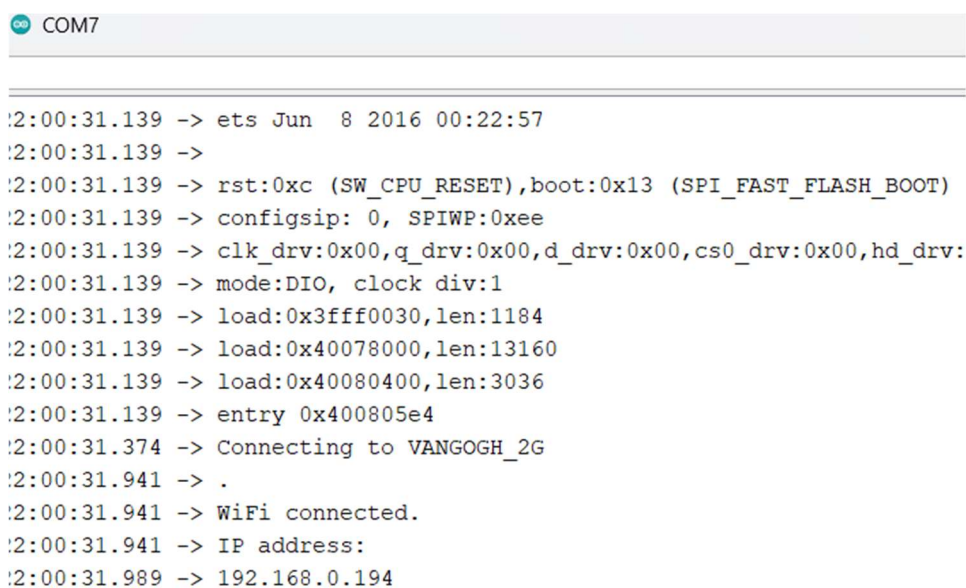
Figura 119 – Configurações de IP estático para configurações de porta



Fonte: Autor (2023).

Se abrirmos o monitor serial, conseguimos verificar que o ESP32 conseguiu se conectar com o roteador de internet:

Figura 120 – Monitor serial



```
COM7
:2:00:31.139 -> ets Jun  8 2016 00:22:57
:2:00:31.139 ->
:2:00:31.139 -> rst:0xc (SW_CPU_RESET),boot:0x13 (SPI_FAST_FLASH_BOOT)
:2:00:31.139 -> configsip: 0, SPIWP:0xee
:2:00:31.139 -> clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:
:2:00:31.139 -> mode:DIO, clock div:1
:2:00:31.139 -> load:0x3fff0030,len:1184
:2:00:31.139 -> load:0x40078000,len:13160
:2:00:31.139 -> load:0x40080400,len:3036
:2:00:31.139 -> entry 0x400805e4
:2:00:31.374 -> Connecting to VANGOGH_2G
:2:00:31.941 -> .
:2:00:31.941 -> WiFi connected.
:2:00:31.941 -> IP address:
:2:00:31.989 -> 192.168.0.194
```

Fonte: Autor (2023)

Para acessarmos as configurações do roteador, precisamos digitar o endereço do gateway em qualquer navegador.

Figura 121 – Configurações roteador (endereço de gateway)

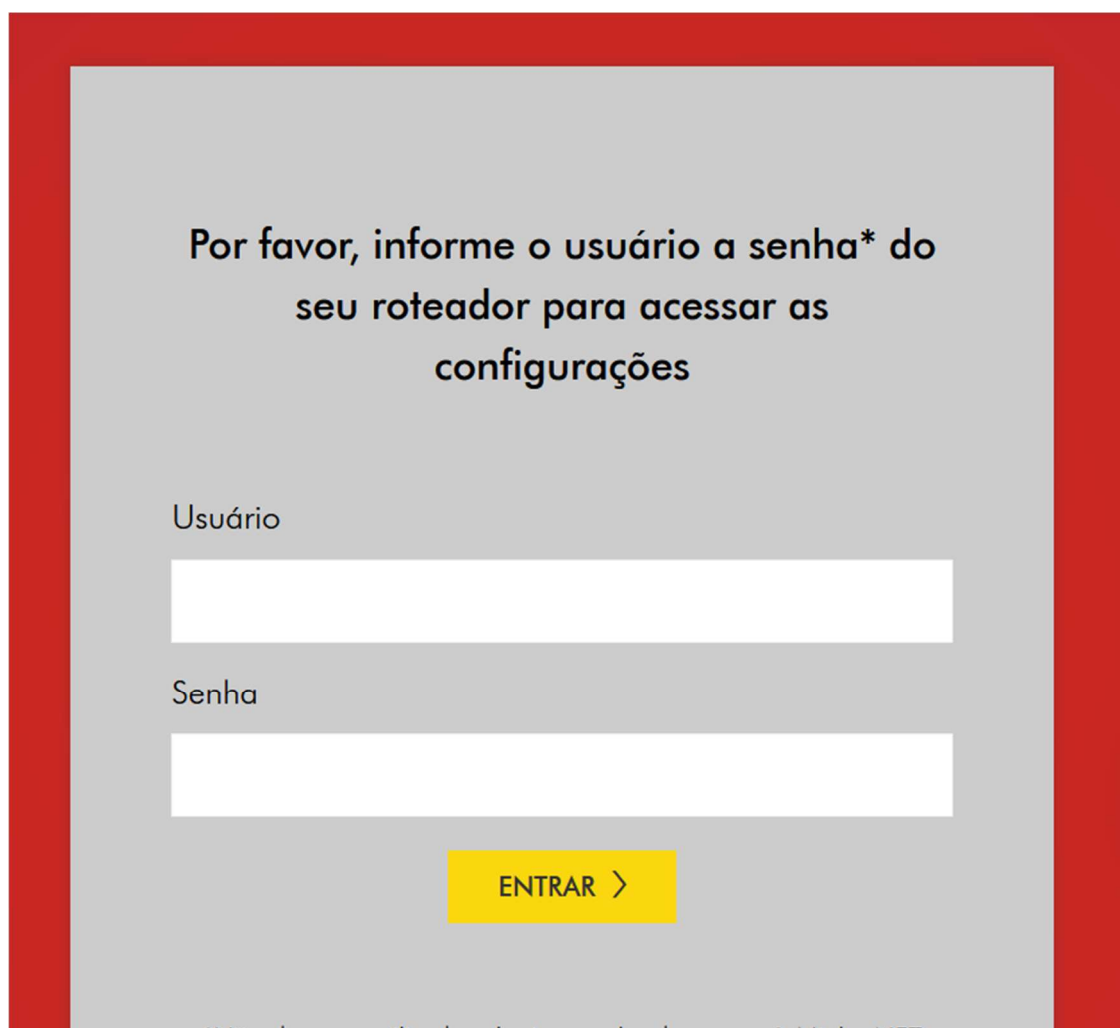


Q 192.168.0.1

Fonte: Autor (2023)

Digitando o número do gateway, entraremos em um menu que solicitará o usuário e senha de administrador do roteador. Essa informação geralmente consta no verso do roteador (caso não tenha sido alterada).

Figura 122 – Menu de usuário e senha roteador



Por favor, informe o usuário a senha\* do seu roteador para acessar as configurações

Usuário

Senha

ENTRAR >

Fonte: Autor (2023).

Após entrarmos no menu principal, temos que entrar em configurações avançadas indicadas pela seta abaixo.



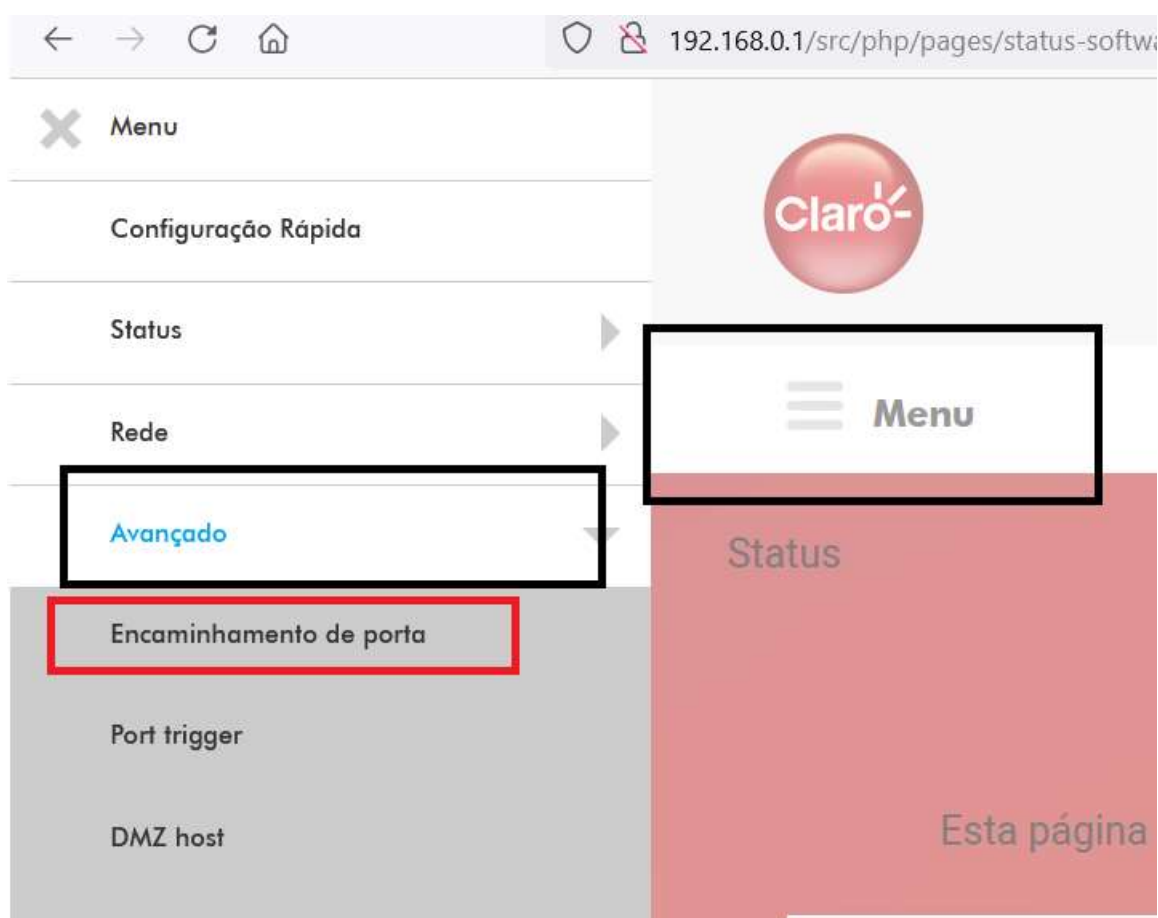
Figura 123 – Configurações avançadas



Fonte: Autor (2023).

Após selecionarmos as configurações avançadas, apertamos no menu, depois avançado e encaminhamento de porta:

Figura 124 – Menu de encaminhamento de porta



Fonte: Autor (2023).

Vamos agora criar a nossa regra de encaminhamento de porta:

Figura 125 – Menu de criação de encaminhamento de porta

Avançado

## Encaminhamento de porta

Selecione o nome do serviço e digite o endereço IP do servidor, clique em "APLICAR AJUSTES" para encaminhar pacotes IP, na porta do serviço escolhido, para o IP do servidor especificado

CRIAR >

Fonte: Autor (2023).

Por fim, escolhemos a porta do nosso servidor web que no caso é 8080, escolhemos o IP estático da placa e como serviço escreveremos ESP32.

Figura 126 – Abrindo uma Porta

Interface de Uso

wan-ip-interface/wanbridge

**Nome do Serviço**

SELECIONE UM SERVIÇO

Serviço Customizado

ESP32

Endereço IP do Servidor

192.168.0.194

Porta externa inicial	Porta externa final	Protocolo	Porta inicial interna	Porta final interna
8080	8080	Ambos ▾	8080	8080

Fonte: Autor (2023).

É necessário ressaltar que tais configurações de portas variam conforme o fabricante, o que foi utilizado aqui foi um modem TG3442 da arca Arris, cuja operadora é claro-net. Entretanto, testando em outros roteadores, as configurações se assemelham muito, mas caso tenha dúvidas, procure por encaminhamentos de portas e o modelo de seu roteador.

Ao escolhermos a porta que vai rodar nosso servidor web é necessário destacar que, é necessário adicionar a porta no final do endereço como pode ser visto a seguir:

Figura 127 – Porta final do endereço



Fonte: Autor (2023)

**Observação:** Observe que no exemplo acima depois dois endereços, o gerado pelo duckdns e o IP estático. A diferença entre um e outro é que para acessarmos de fora da rede privada, ou seja, pela internet, utilizamos o endereço pelo duckdns, caso estejamos conectados na mesma rede, utilizamos o número de IP estático.

Vamos agora para os elementos que se conecta com aba conteúdos do servidor, começando com o elemento botão

Figura 128 - Botão



Fonte: Autor (2023)

A função do botão é de controlar um GPIO, ou seja, podemos ligar ou desligar o GPIO apenas clicando no botão. Temos além de escolher o GPIO, temos o elemento de uma legenda para o nosso botão e uma variável para distinguir de outros elementos dentro do código. É necessário que cada elemento tenha um nome diferente de variável para que não aja conflito dentro do código.

O resultado dentro do código fica da seguinte maneira:

Figura 129 – Resultado Botão



Fonte: Autor (2023).

Para mudar o estado o estado do botão bata apenas clicar em cima do mesmo, caso o botão esteja ligado ele fica verde, caso esteja desligado, ele fica vermelho. O botão também conta com uma barra de status para evidenciar o seu estado.

Mas caso não desejarmos ligar um GPIO, caso desejarmos criar um comando cujo o programa irá executar uma determinada tarefa caso seja ativo e deixar de executar caso seja desativado? Para isso foi criado o botão programável.

Figura 130 – Botão programável



Fonte: Autor (2023).

O botão programável funciona de forma muito semelhante a uma estrutura do tipo se/senão, de modo que, todos os comandos que estiverem dentro do ativo serão acionados caso o botão seja ativado e todos os comandos dentro do desativo servem para uma estrutura para desativar ou não executar os comandos dentro dessa programação. Podemos criar estruturas mais complexas com esse tipo de botão. O resultado que obteremos na estrutura do servidor web pode ser visto a seguir:

Figura 131 – Resultado Botão programável

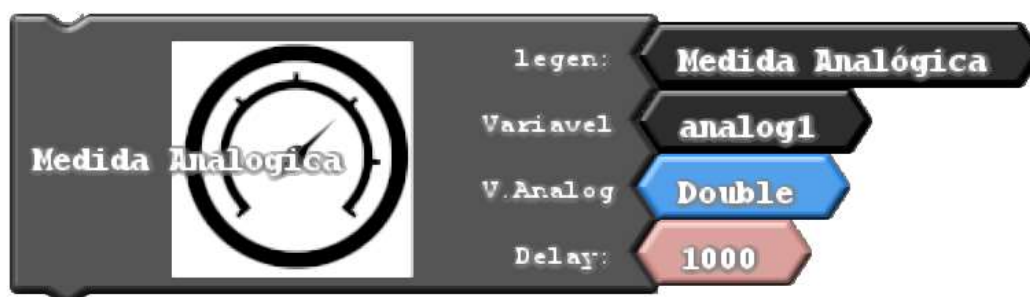


Fonte: Autor (2023).

Assim como o botão que controla o GPIO, o botão programável também necessita atribuímos uma legenda e escolhermos um nome de variável para diferenciar dos outros elementos.

O comando a seguir é para imprimir no nosso site uma medida analógica:

Figura 132 – Medida analógica servidor web



Fonte: Autor (2023)

O bloco acima apresenta como os outros elementos uma legenda, a escolha de uma variável para diferenciação dos outros elementos, que no caso do bloco acima está salvo como `analog1`, porém, o código em si não faz nenhuma medida analógica, apenas fazendo sua impressão essa é variável analógica. No caso acima está chamando uma variável do tipo decimal pelo nome da variável, entretanto, poderia ser uma variável do tipo inteira. O *delay* é o tempo que vai levar para coletar a próxima medida, podendo ser ajustado o valor. A medida analógica vai ter uma implicação

posterior quando tratarmos do gráfico, mas o resultado no servidor fica da seguinte maneira:

Figura 133 – Resultado da medida analógica servidor web

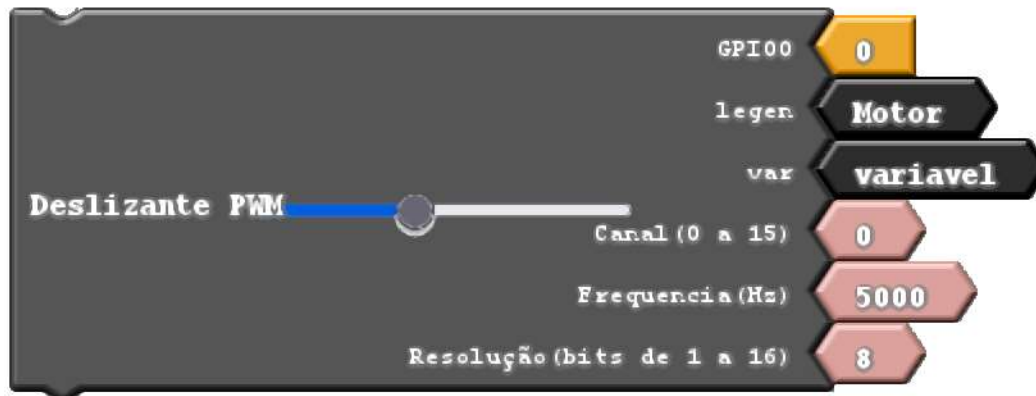
Medida Analógica 77.00

Fonte: Autor (2023)

Lembrando mais uma vez que o bloco de medida analógica não realiza a medida em si, mas apenas imprime no site.

Vamos agora para um controle deslizante de PWM, seu funcionamento é muito parecido com o que foi abordado no bloco pino PWM voltado pro pra execução dentro do ESP32, porém tem algumas diferenças, entre elas, a legenda e escolha do nome de variável para evitar conflitos dentro do ESP32.

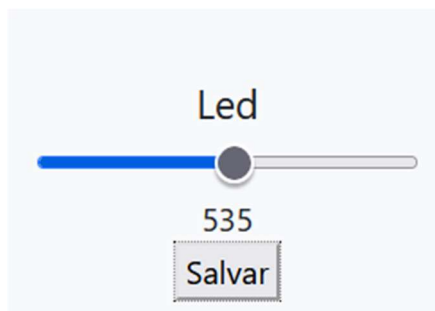
Figura 134 – Bloco de controle deslizante controlador de PWM



Fonte: Autor (2023).

Observe que o funcionamento é exatamente como o do PWM já explicado, com os canais, frequência e resolução. Vamos ver o resultado por dentro do servidor Web:

Figura 135 – Resultado PWM no servidor Web



Fonte: Autor (2023).

Perceba que temos que deslizar até o valor desejado e salvar para que possa modificar a medida do PWM.

Chegamos ao último bloco que iremos abordar aqui nesta introdução do Ardublock no ESP32. A o bloco a seguir é para traçarmos gráficos de variações de medidas analógicas pelo tempo.

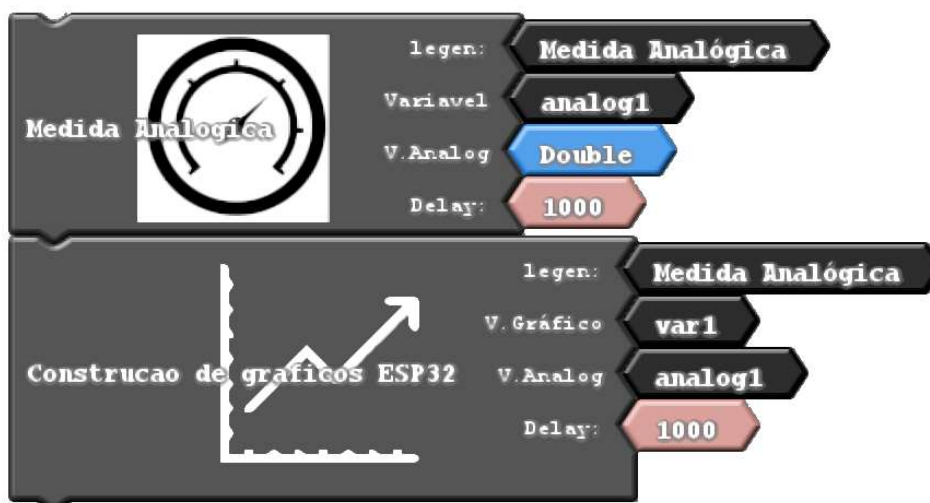
Figura 136 – Gráfico no ESP32



Fonte: Autor (2023).

O bloco acima consta com uma legenda para o gráfico, uma variável para diferenciação de elementos, porém temos a variável analógica, dependemos de outro bloco para traçarmos o bloco, que é o de impressão de medida analógica no servidor web, chamaremos o valor de medida analógica através do nome da variável atribuída. Temos também o *delay* que determina qual o tempo para coletar a próxima medida, Precisamos destes dois blocos em conjunto:

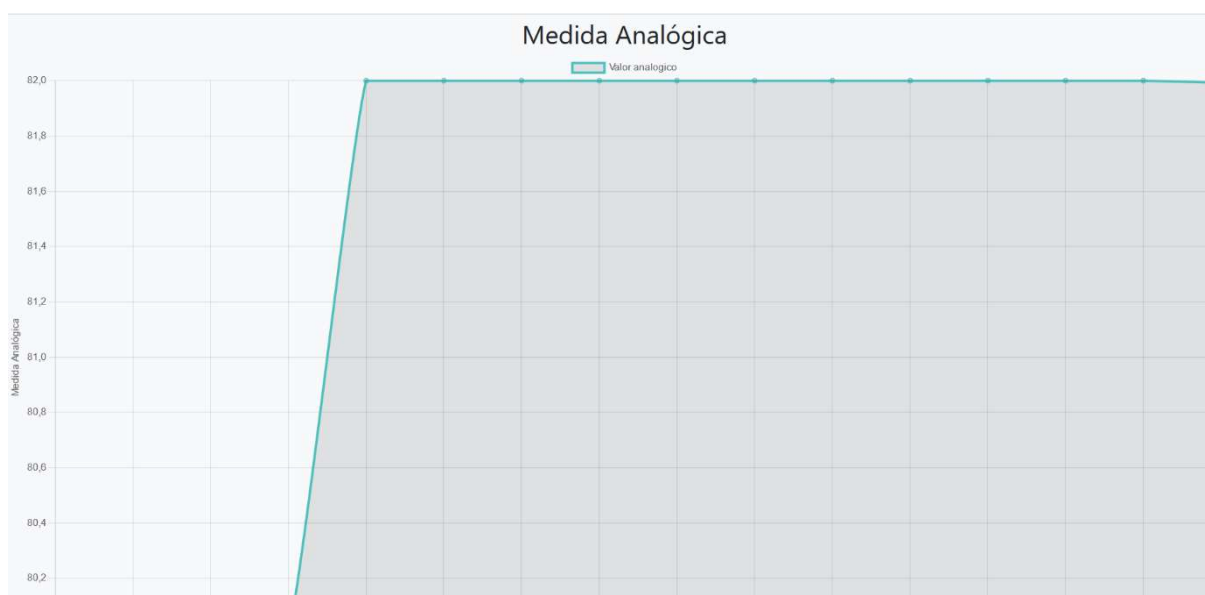
Figura 137 – Gráfico e medida analógica



Fonte: Autor (2023)

O resultado no nosso gráfico no nosso servidor web fica da seguinte maneira:

Figura 138 – Gráfico no servidor web



Fonte: Autor (2023).

## Atividade Capítulo 10

Para realizar esta atividade vamos precisar:

- Computador com Arduino IDE
- ESP32



- 1 Jumper (conector)

O objetivo desse módulo é colocar uma aplicação simples online, porém, utilizando o máximo de blocos possíveis. Esta é última atividade deste material, iniciaremos então começando com o bloco de módulo wifi do ESP32:

Figura 139 – Blocos de conexão



Fonte: Autor (2023).

Utilizamos também o bloco de conexão wifi escolhendo o nome da rede e a senha atribuída.

Em sequência utilizaremos o bloco do duckdns para podermos acessar a nossa aplicação através do domínio para isso criamos um domínio cujo o endereço foi <http://esp32testes1.duckdns.org> e utilizamos o token vinculado a nossa conta:

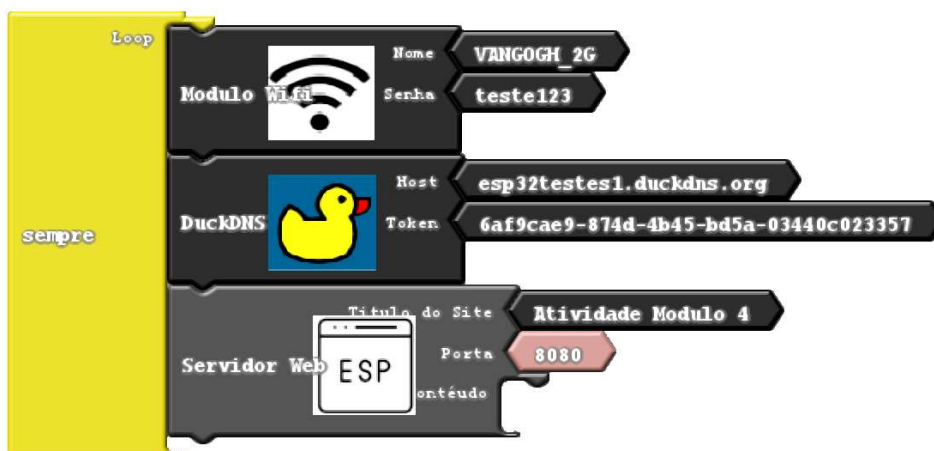
Figura 140 – inclusão do bloco duckdns



Fonte: Autor (2023).

Adicionamos então o nosso servidor web, atribuindo um título e vinculando na porta 8080 (já explicado no decorrer do módulo como vinculamos a porta com o ESP32).

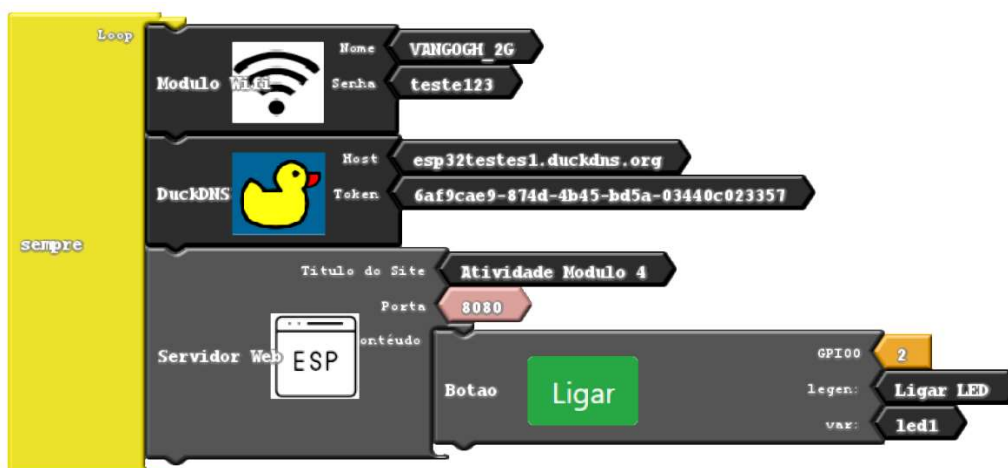
Figura 141 – Inclusão do servidor web



Fonte: Autor (2023).

Iremos adicionar agora um bloco de botão para ligarmos o GPIO 2, onde no ESP32 tem um LED de testes:

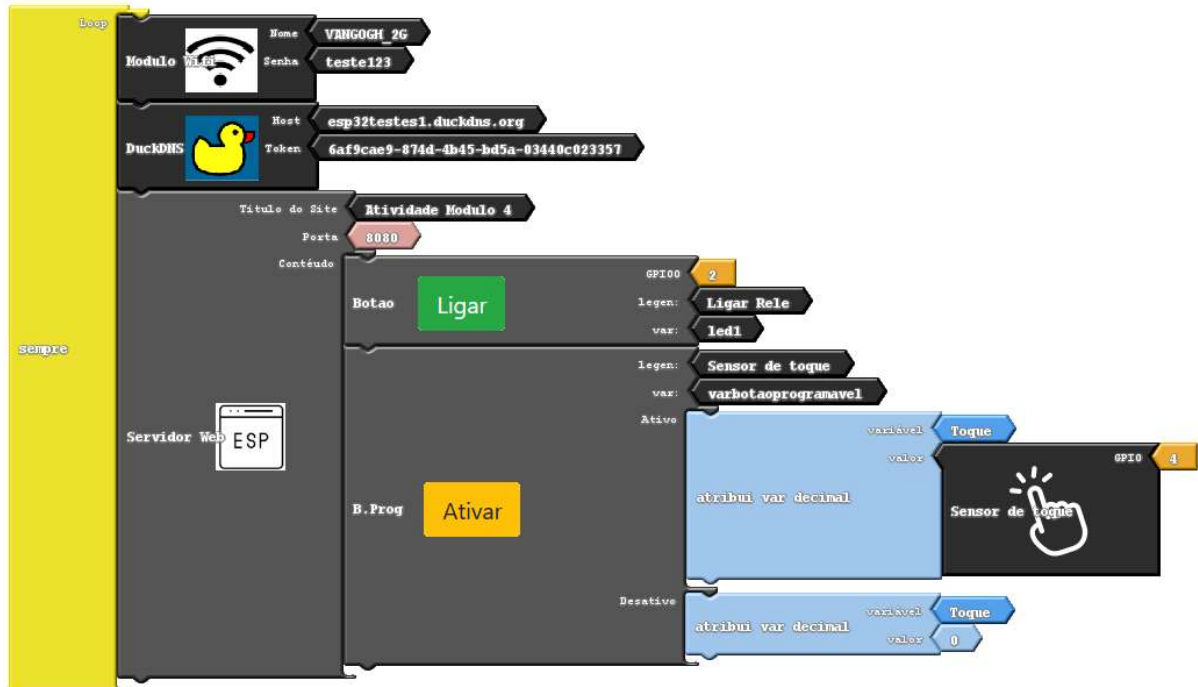
Figura 142 – Inclusão do botão



Fonte: Autor (2023)

Incluiremos agora um botão programável, que caso esteja ativo, recolhera a informação do sensor de toque do GPIO 4, caso não esteja desativado, atribuirá um valor igual a zero a variável. Observação a variável poderia estar em outro lugar dentro do código, entretanto, se optou por utilizar dentro do botão programável para podermos utiliza-lo como recurso.

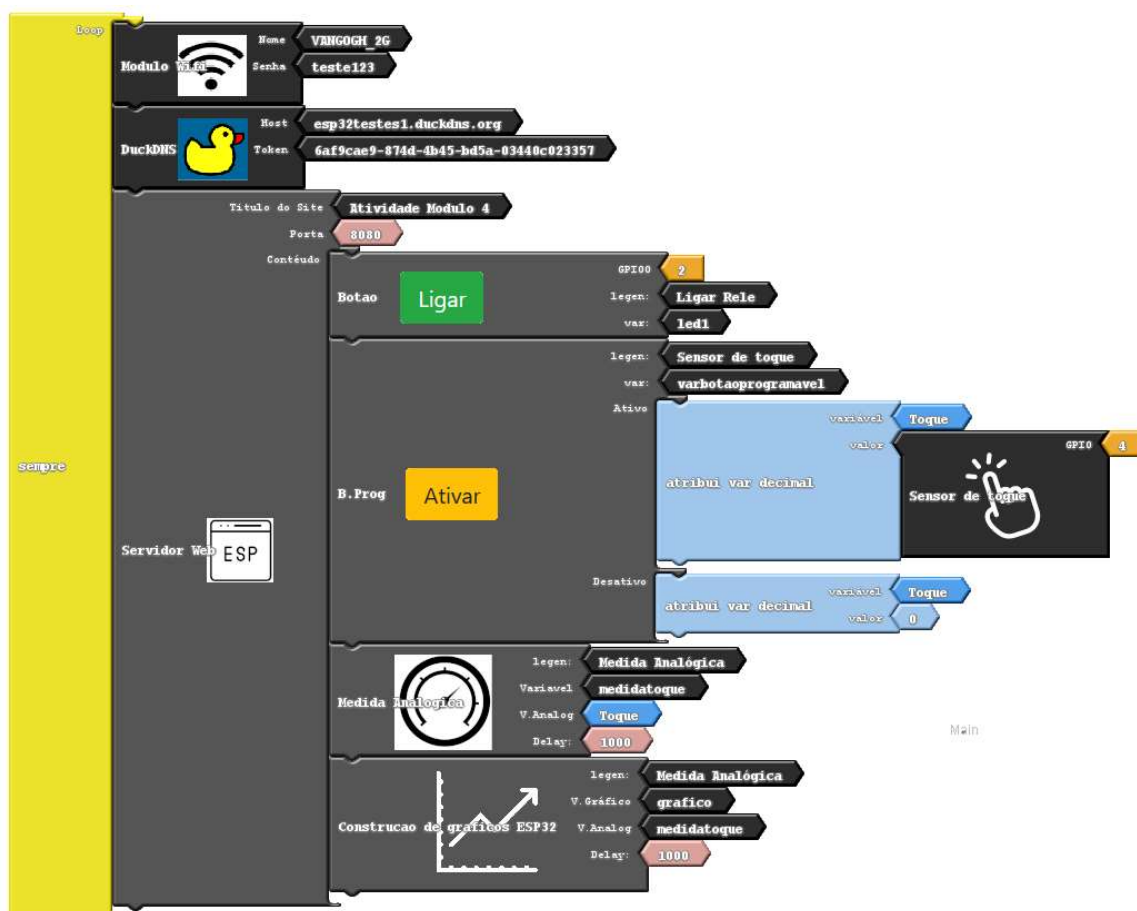
Figura 143 – Botão programável e coleta de medidas analógicas



Fonte: Autor (2023).

Após a atribuição de variável, vamos utilizar a medida analógica para ser impressa dentro do nosso código:

Figura 144 - Inclusão medida analógica

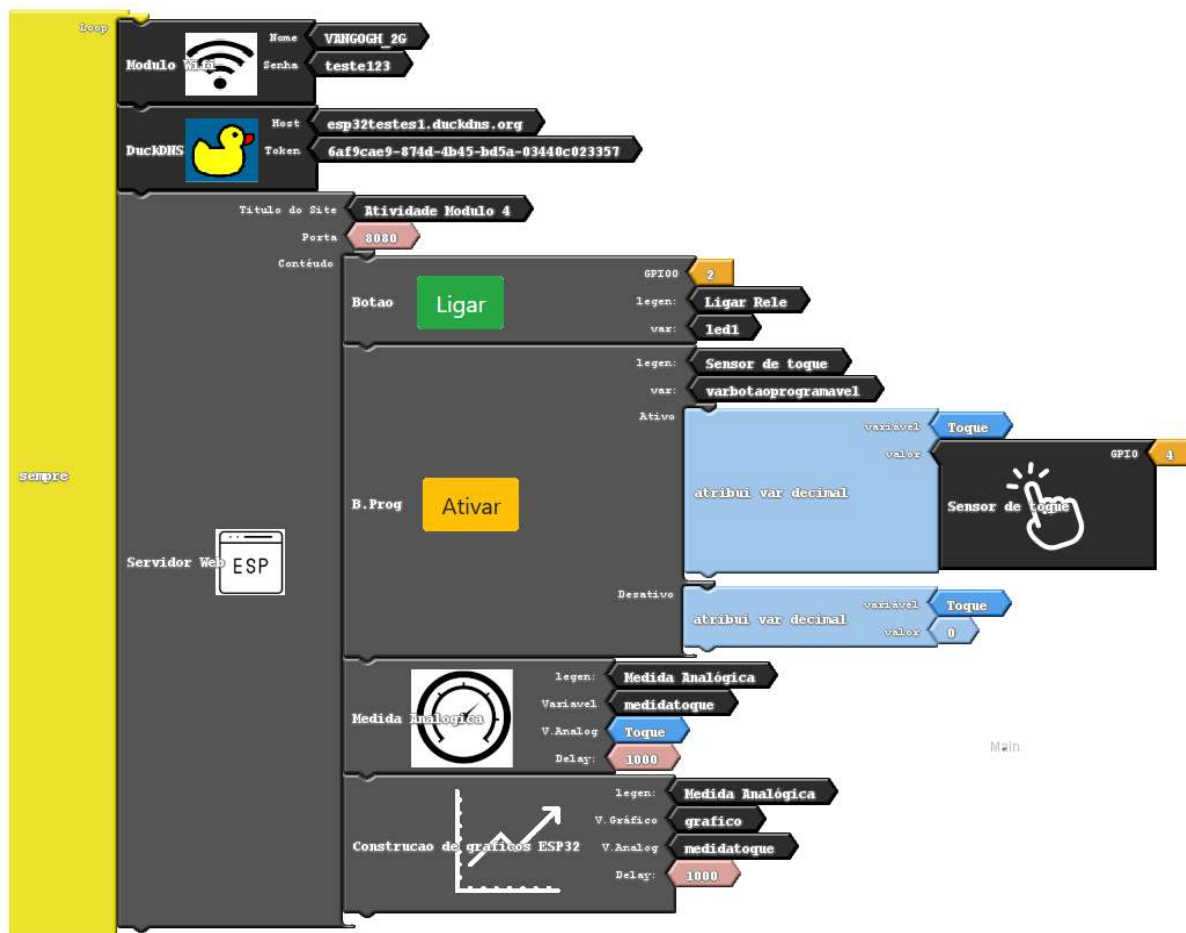


Fonte: Autor (2023)

Note para podermos imprimir nossa medida analógica, tivemos que chamar a variável exatamente pelo mesmo nome que atribuímos a coleta do valor de toque.

O último passo que temos que realizar é inclusão do bloco de gráfico, no qual temos que chamar a variável utilizada anteriormente, sendo seu nome toquemedida1 para podermos traçar o nosso gráfico.

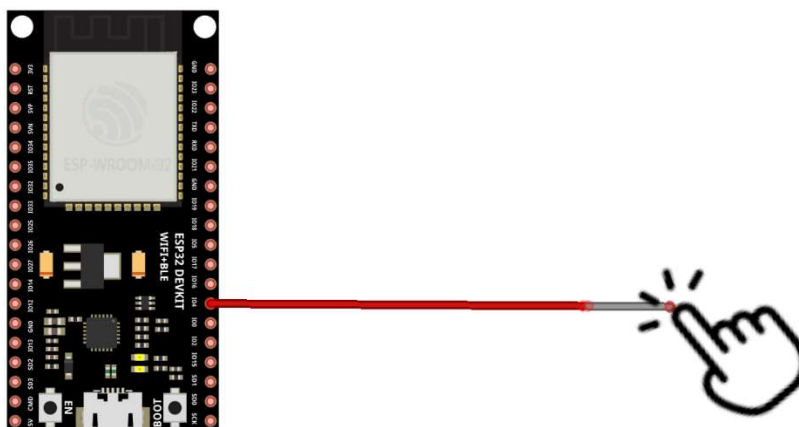
Figura 145 – Código finalizado ESP32



Fonte: Autor (2023)

As conexões no ESP32 são relativamente simples, basta ligar um *jumper* (conector) no GPIO 4 com uma ponta do tipo macho e tocar no terminal do mesmo. O LED de testes já está integrado no GPIO 2.

Figura 146 – Conexões ESP32



Fonte: Autor (2022)

Para acessarmos a nossa aplicação temos que acessar no nosso navegador o endereço que utilizamos no duckdns e a porta escolhida pela porta do servidor web separadas por dois pontos “:” : <http://esp32testes1.duckdns.org:8080/>

Caso seja uma conexão local, ou seja, se você deseja conectar na placa mas está na mesma rede de internet (conectados no mesmo roteador) é necessário digitar o número de IP da placa que no caso é 192.168.0.194 e a porta que escolhemos para o servidor web que no caso é 8080 separando por dois pontos “:” da seguinte maneira <http://192.168.0.194:8080> .

Para verificar, conectamos de uma rede diferente do roteador e verificamos que a aplicação está online e obtemos como resultado:

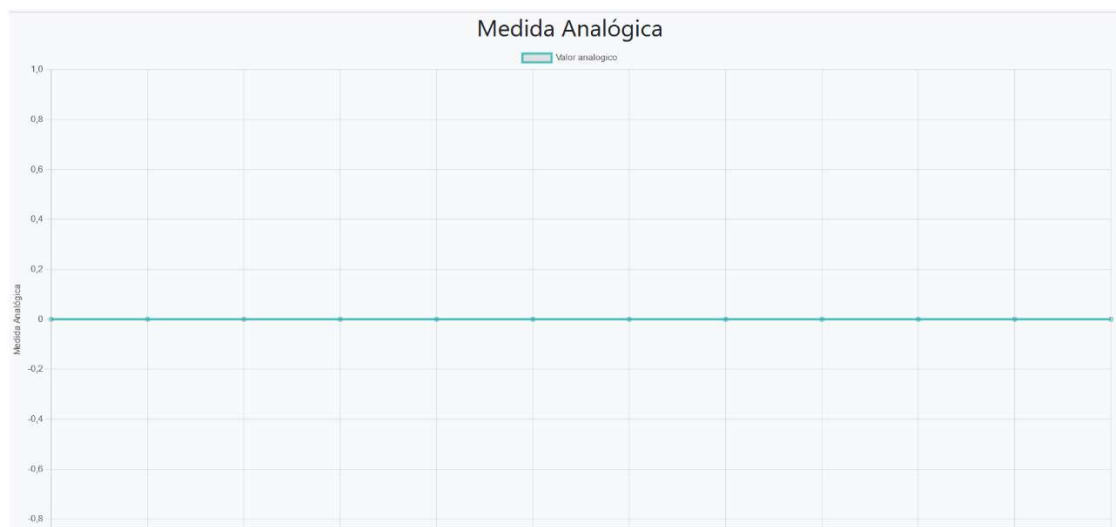
Figura 147 – Resultado aplicação do ESP32



Fonte: Autor (2023).

O gráfico fica da seguinte maneira:

Figura 148 – Resultado ESP32 (gráfico 1)



Fonte: Autor (2023).

Vamos ativar a medida analógica e ligar o LED de teste, vamos ter o seguinte resultado:

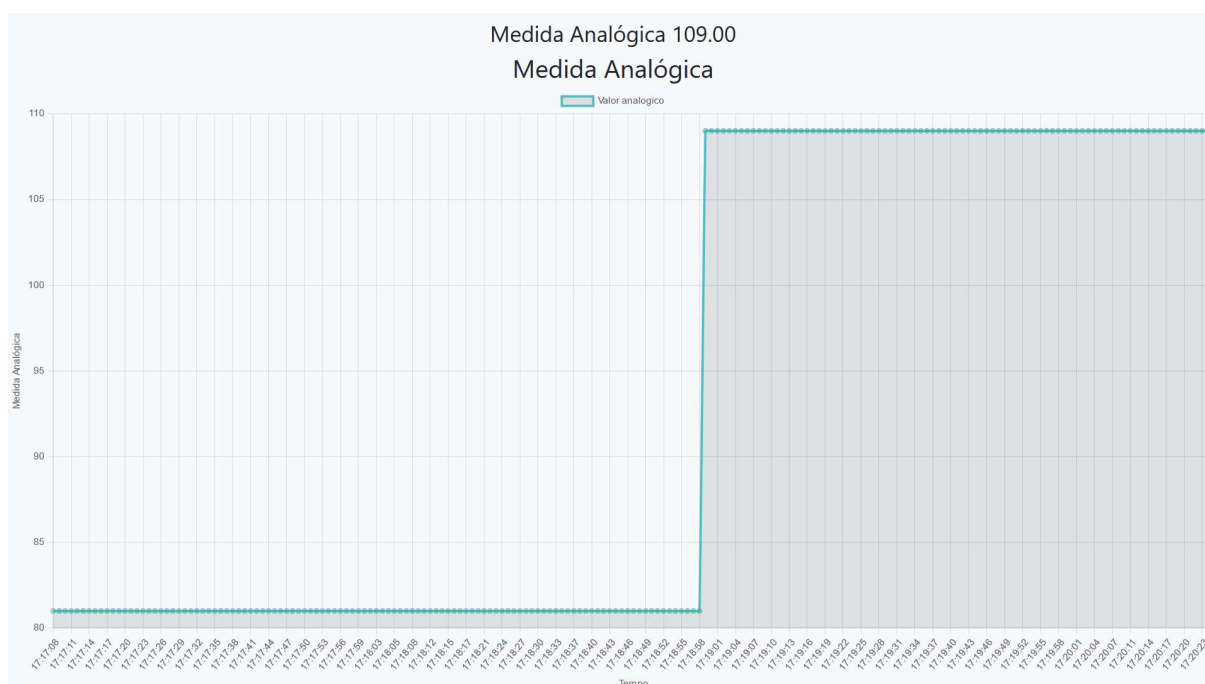
Figura 149 – Resultado aplicação ESP32 (ligando os botões)



Fonte: Autor (2023).

Por consequência temos o seguinte comportamento do gráfico:

Figura 150 – Resultado ESP32 (gráfico 2)



Fonte: Autor (2023)

Para concluirmos, a razão de ter sido feitas todas essas modificações do ESP32 se deve ao fato de que, de apesar termos aplicações que o utilizam e tem diversos elementos elas são pagas como por exemplo: Arduino Cloud e o Blink. Esses serviços oferecem os recursos de forma limitada. Existe a possibilidade gratuita, entretanto sua programação não é tão simples. As modificações do Ardublock aqui feita são para facilitar a vida de usuários iniciantes, professores e estudantes.



## REFÊRENCIAS

BRANCO, Anselmo Lázaro. Revoluções industriais - Primeira, segunda e terceira revoluções. Disponível em:

<https://educacao.uol.com.br/disciplinas/geografia/revolucoes-industriais-primeira-segunda-e-terceira-revolucoes.htm>.

LOPES, Ricardo Gomes. Iniciação à programação em blocos utilizando Ardublock e plataforma Arduino. 135 p. 2019. Trabalho de Conclusão de Curso (Graduação em Licenciatura em Física) – Universidade Federal do Pampa, Campus Bagé, Bagé, 2019.

Morsch, José Aldair. “O que são as ondas do eletrocardiograma e como interpretar?”

**Tele medicina morsch** Retirado de:

<https://telemedicinamorsch.com.br/blog/complexo-qrs>.

## Anexo A – Guia de Instalação do ARDUBLOCK

### Guia de Instalação do Ardublock para o Arduino ide 1.82 ou superior

Este guia foi retirado do site <http://teachduino.ufsc.br/2017/09/22/novo-ardublock-edicao-para-fisica/> na data de 3 de maio de 2019 e adaptado pelo autor da apostila.

#### Instalação:

- O primeiro passo é baixar o Ardublock no link abaixo:

<http://teachduino.ufsc.br/2017/09/22/novo-ardublock-edicao-para-fisica/>

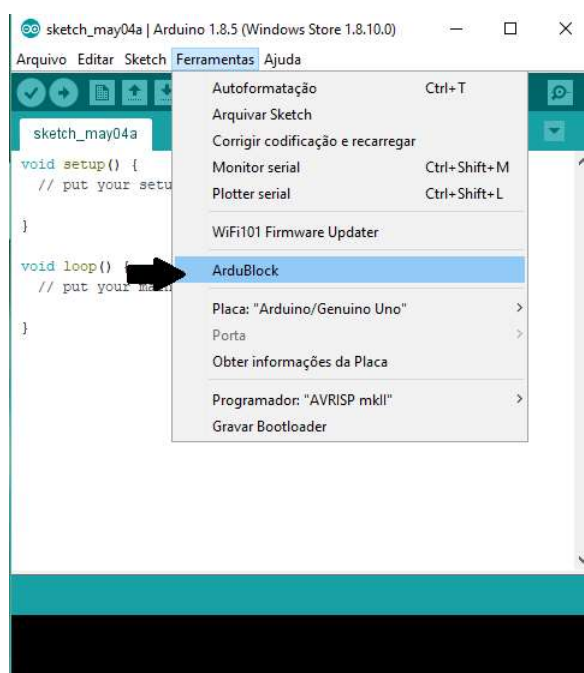
OBS: Para instalar o Ardublock o programa do Arduino deve estar fechado:

- Vá na pasta “Meus Documentos” ou em “C:/users/seu\_usuario/Documents/”
- Abra a pasta escrito “Arduino” e crie uma nova pasta chamada “tools”(a pasta deve estar com o nome todo em minúsculo)
- Posteriormente, abra a pasta “tools” e crie a pasta “ArduBlockTool”(Exatamente como está escrita). O próximo passo é abrir a pasta “ArduBlockTool” e criar uma pasta chama “tool”(tudo em minúsculo).
- O ultimo passo de instalação é copiar o arquivo baixo no link para dentro da pasta “tool”.

OBS: É necessário ter o java instalado, entretanto, se o programa Arduino IDE roda perfeitamente, significa que o programa já está instalado

**Testar o Ardublock:** Abra o Arduino IDE vá em “Ferramentas” e selecione a opção “ArduBlockTool”, conforme a imagem abai

Figura 1 – Aba Ardublock



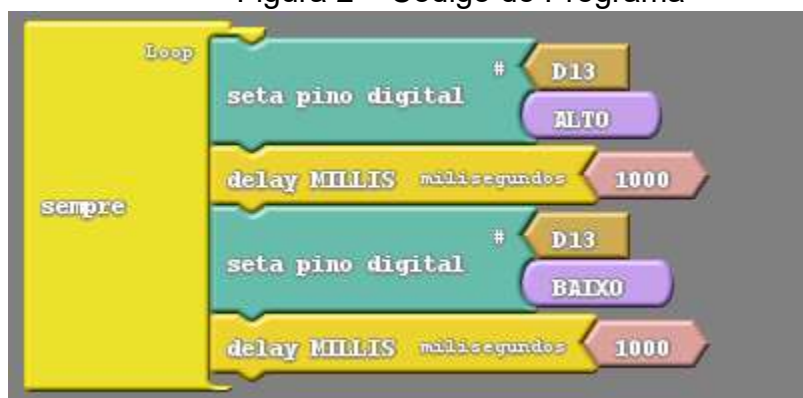
Fonte: Lopes (2019)

Fazendo programa teste (blink):

1. Vá em “controle” e selecione o código “sempre”
2. Vá em pino e selecione “seta pino digital”, selecionar a porta 13 e colocar no estado “Alto” e coloque dentro do bloco “sempre”.
3. Vá em “controle” e selecione o código “delay MILLIS(milissegundos)” estabeleça o valor 1000 e coloque dentro do bloco sempre.
4. Vá em pino e selecione “seta pino digital”, selecionar a porta 13 e colocar no estado “Baixo” e coloque dentro do bloco “sempre”.
5. Vá em “controle” e selecione o código “delay MILLIS(milissegundos)” estabeleça o valor 1000 e coloque dentro do bloco sempre.

A figura abaixo mostrará a estrutura do programa:

Figura 2 – Código do Programa



Fonte: LOPES(2019)

Salve o programa na aba “Salvar” e logo após clique em “Enviar para Arduino”, se o código foi transcrito para o Arduino IDE, o programa Ardublock funcionou sem problemas, basta compilar e executar placa Arduino.

Figura 3 – Transposição do Ardublock pro Arduino IDE



The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_may04b | Arduino 1.8.5 (Windows Store 1.8.10.0)". The menu bar includes "Arquivo", "Editar", "Sketch", "Ferramentas", and "Ajuda". The toolbar contains icons for saving, running, and uploading. The main editor area shows the following code:

```
sketch_may04b $  
  
void __ardublockDigitalWrite(int pinNumber, boolean status)  
{  
  pinMode(pinNumber, OUTPUT);  
  digitalWrite(pinNumber, status);  
}  
  
void setup()  
{  
}  
  
void loop()  
{  
  __ardublockDigitalWrite(13, HIGH);  
  delay( 1000 );  
  __ardublockDigitalWrite(13, LOW);  
  delay( 1000 );  
}
```

Below the code editor, an error message is displayed in a red box: "Problema ao carregar para a placa. Veja <http://www.arduino.cc/en/Guide/Tools> para obter mais informações." A "Copiar mensagens de erro" button is visible to the right of the message. The status bar at the bottom indicates "1" and "Arduino/Genuino Uno em COM1".

Fonte: Lopes (2019)

## **Apêndice A – Repositório Ardublock Modificado**

A versão do Ardublock está disponível no seguinte link:

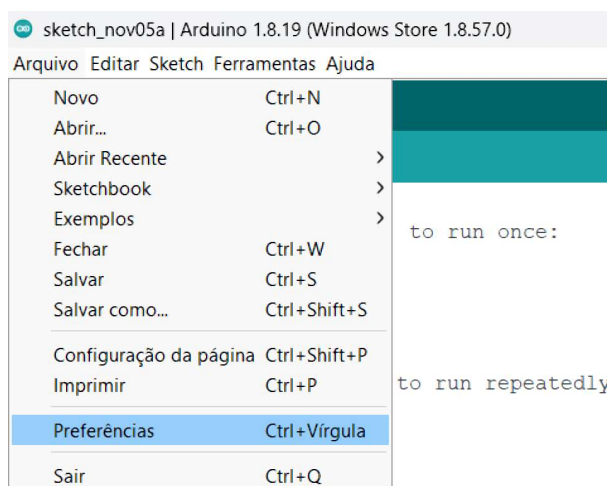
<https://github.com/Ricardio42/ArdublockModificado>

No repositório contém as bibliotecas utilizadas a uma versão criada pelo autor dos códigos trabalhados no decorrer deste repositório

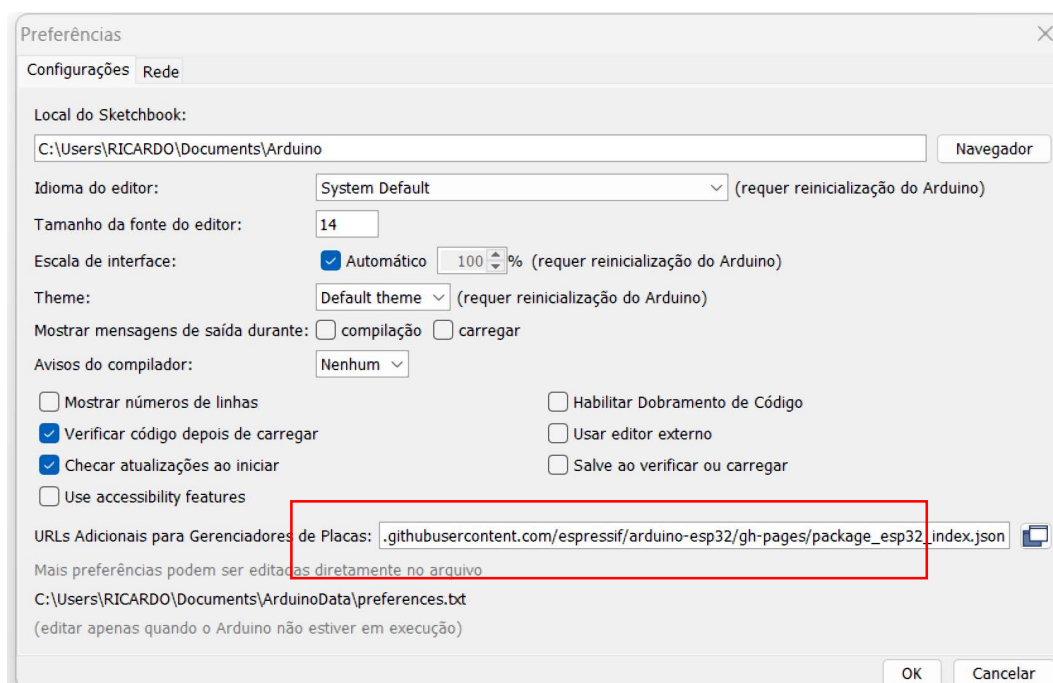
## Apêndice B – Guia de Instalação da Placa ESP32

Para instalarmos as placas do ESP32 temos que abrir a IDE do Arduino e ir até em Arquivo>Preferencias:

### Arquivo>Preferencias do Arduino IDE

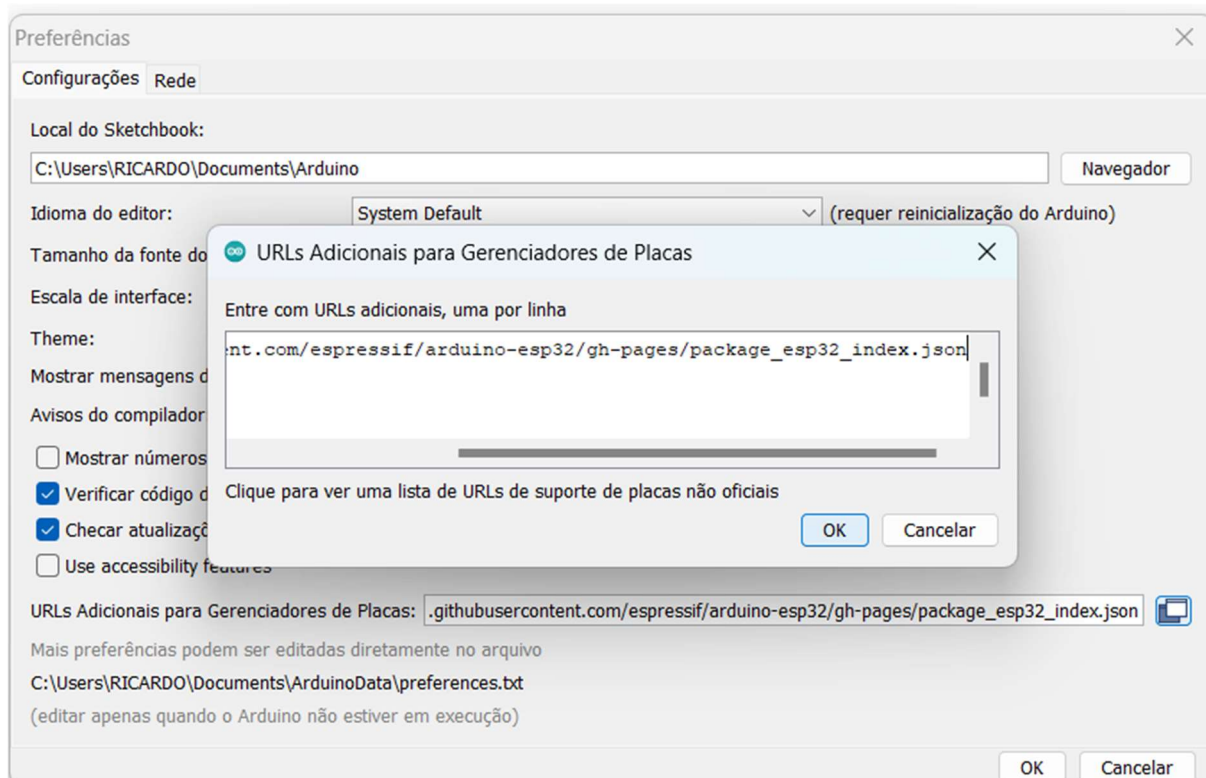


Vamos então para URL para Gerenciador de placas

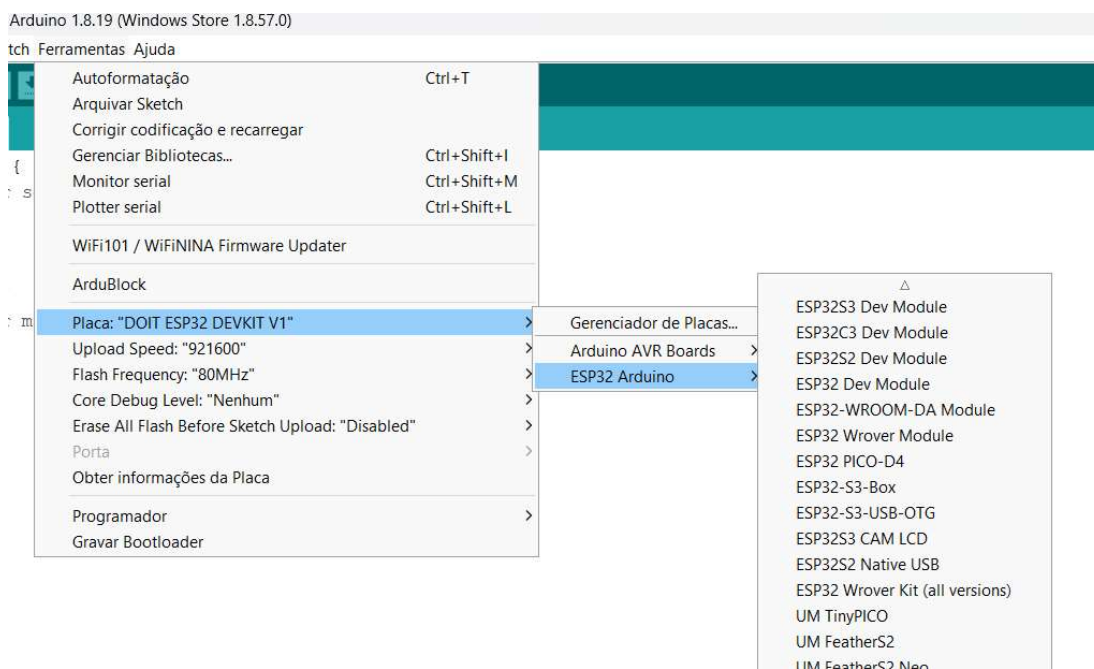


Digitamos o seguinte link: [https://raw.githubusercontent.com/espessif/arduino-esp32/gh-pages/package\\_esp32\\_index.json](https://raw.githubusercontent.com/espessif/arduino-esp32/gh-pages/package_esp32_index.json)

Apertamos no botão ao lado da barra de digitação do endereço e aparecera a seguinte tela:



Basta então apertar ok nas duas guias abertas e reiniciar a IDE do Arduino. As placas do ESP32 vão estar em ferramentas > Placas > ESP32



## Apêndice C – Licença Creative Commons



# CC BY-NC-SA 4.0 DEED

## Attribution-NonCommercial-ShareAlike 4.0 International

**BY:** O crédito deve ser dado a você, o criador.

**NC:** Somente o uso não comercial do seu trabalho é permitido. Não comercial significa que não se destina principalmente ou não é direcionado a vantagens comerciais ou compensação monetária.

**SA:** As adaptações devem ser compartilhadas nos mesmos termos.



ISBN: 978-65-00-99483-4

**BR**



9 786500 994834